aws

개발자 가이드

AWS Cloud Map

AWS Cloud Map: 개발자 가이드

Table of Contents

란 무엇입니까 AWS Cloud Map?

AWS Cloud Map 는 애플리케이션이 의존하는 백엔드 서비스 및 리소스에 논리적 이름을 매핑하는 데 사용할 수 있는 완전관리형 솔루션입니다. 또한 애플리케이션이 AWS SDKs, RESTful API 호출 또는 DNS 쿼리 중 하나를 사용하여 리소스를 검색하는 데 도움이 됩니다.는 Amazon DynamoDB(DynamoDB) 테이블, Amazon Simple Queue Service(Amazon SQS) 대기열, Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 또는 Amazon Elastic Container Service(Amazon ECS) 작업을 사용하여 빌드된 상위 수준 애플리케이션 서비스 등 정상적인 리소스만 AWS Cloud Map 제공합니다.

의 구성 요소 AWS Cloud Map

네임스페이스

시작하려면 먼저 애플리케이션에 대한 서비스를 그룹화하는 방법으로 작동하는 AWS Cloud Map 네임스페이스를 생성합니다. 네임스페이스는 리소스를 찾는 데 사용할 이름을 식별하고 AWS Cloud Map DiscoverInstances API 호출, VPC의 DNS 쿼리 또는 퍼블릭 DNS 쿼리를 사용하여 리소스를 찾는 방법을 지정합니다. 대부분의 경우 네임스페이스는 요금 청구 애플리케이션 등과 같은 단일 애플리케이션에 대한 모든 서비스를 포함합니다. 자세한 내용은 AWS Cloud Map 네임스페이스 단원을 참조하십시오.

Service

네임스페이스를 생성한 후 엔드포인트를 찾는 AWS Cloud Map 데 사용할 각 리소스 유형에 대한 AWS Cloud Map 서비스를 생성합니다. 예를 들어, 웹 서버 및 데이터베이스 서버를 위한 서비스를 생성할 수 있습니다.

서비스는 애플리케이션이 다른 웹 서버와 같은 다른 리소스를 추가할 때가 AWS Cloud Map 사용하는 템플릿입니다. 네임스페이스를 생성할 때 DNS를 사용하여 리소스를 찾도록 선택한 경우, 서비스에는 웹 서버를 찾을 때 사용하려는 레코드 유형에 대한 정보가 포함됩니다. 또한 서비스는 리소스의 상태를 확인할지 여부와 Amazon Route 53 상태 확인 또는 타사 상태 확인을 사용할지 여부를 나타냅니다. 자세한 내용은 AWS Cloud Map 서비스 단원을 참조하십시오.

서비스 인스턴스

애플리케이션이 리소스를 추가하면 코드에서 AWS Cloud Map RegisterInstance API 작업을 호출하여 서비스에서 AWS Cloud Map 서비스 인스턴스를 생성할 수 있습니다. 서비스 인스턴스에는 DNS를 사용하든 AWS Cloud Map DiscoverInstances API 작업을 사용하든 상관없이 애플리케이션이 리소스를 찾는 방법에 대한 정보가 포함되어 있습니다.

애플리케이션이 리소스에 연결해야 하는 경우 리소스와 연결된 네임스페이스 및 서비스를 지정하여 DiscoverInstances를 호출하거나 퍼블릭 또는 프라이빗 DNS 쿼리를 활용합니다.는 하나 이상의 리소스를 찾는 방법에 대한 정보를 AWS Cloud Map 반환합니다. 서비스를 생성할 때 상태 확인을 지정한 경우는 정상 인스턴스만 AWS Cloud Map 반환합니다. 자세한 내용은 AWS Cloud Map 서비스 인스턴스 단원을 참조하십시오.

액세스 AWS Cloud Map

다음과 같은 방법으로 AWS Cloud Map 에 액세스할 수 있습니다.

- AWS Management Console -이 가이드의 절차에서는를 사용하여 작업을 AWS Management Console 수행하는 방법을 설명합니다.
- AWS SDKs- SDK를 AWS 제공하는 프로그래밍 언어를 사용하는 경우 SDK를 사용하여에 액세스할 수 있습니다 AWS Cloud Map. SDK는 인증을 단순화하고, 개발 환경에 쉽게 통합되며, AWS Cloud Map 명령에 액세스할 수 있도록 합니다. 자세한 내용은 Amazon Web Services용 도구를 참조하세요.
- AWS Command Line Interface - 자세한 내용은 AWS Command Line Interface 사용 설명서의 시작하기 AWS CLI를 참조하세요.
- AWS Tools for Windows PowerShell - 자세한 내용은 AWS Tools for PowerShell 사용 설명서의 시작하기 AWS Tools for Windows PowerShell를 참조하세요.
- AWS Cloud Map API - SDK를 사용할 수 없는 프로그래밍 언어를 사용하는 경우 AWS Cloud Map API 작업 및 API 요청 방법에 대한 자세한 내용은 API 참조를 참조하세요.

> **Note**
>
> IPv6 클라이언트 지원 - 모든 새 리전에서 2023년 6월 22일부터 IPv6 클라이언트 AWS Cloud Map 에서 로 전송된 모든 명령은 새 듀얼 스택 엔드포인트()로 라우팅됩니다`servicediscovery.<region>.api.aws`. AWS Cloud Map IPv62023년 6월 22일 이전에 릴리스된 다음 리전의 레거시(`servicediscovery.<region>.amazonaws.com`) 및 듀얼 스택 엔드포인트 모두에 대해 전용 네트워크에 연결할 수 있습니다.
>
> - 미국 동부(오하이오) - us-east-2
> - 미국 동부(버지니아 북부) - us-east-1
> - 미국 서부(캘리포니아 북부) - us-west-1
> - 미국 서부(오레곤) - us-west-2
> - 아프리카(케이프타운) – af-south-1

- 아시아 태평양(홍콩) - ap-east-1
- 아시아 태평양(하이데라바드) – ap-south-2
- 아시아 태평양(자카르타) – ap-southeast-3
- 아시아 태평양(멜버른) – ap-southeast-4
- 아시아 태평양(뭄바이) - ap-south-1
- 아시아 태평양(오사카) – ap-northeast-3
- 아시아 태평양(서울) - ap-northeast-2
- 아시아 태평양(싱가포르) - ap-southeast-1
- 아시아 태평양(시드니) - ap-southeast-2
- 아시아 태평양(도쿄) - ap-northeast-1
- 캐나다(중부) - ca-central-1
- 유럽(프랑크푸르트) - eu-central-1
- 유럽(아일랜드) - eu-west-1
- 유럽(런던) - eu-west-2
- 유럽(밀라노) – eu-south-1
- 유럽(파리) - eu-west-3
- 유럽(스페인) – eu-south-2
- 유럽(스톡홀름) - eu-north-1
- 유럽(취리히) – eu-central-2
- 중동(바레인) – me-south-1
- 중동(UAE) – me-central-1
- 남아메리카(상파울루) - sa-east-1
- AWS GovCloud(미국 동부) – us-gov-east-1
- AWS GovCloud(미국 서부) - us-gov-west-1

AWS Identity and Access Management

AWS Cloud Map 는 조직에서 다음 작업을 수행하는 데 사용할 수 있는 서비스인 AWS Identity and Access Management (IAM)과 통합됩니다.

- 조직 AWS 계정에서 사용자 및 그룹 생성

- 효율적인 방식으로 AWS 계정 내 사용자 간에 계정 리소스를 공유합니다.
- 각 사용자에게 고유한 보안 자격 증명을 할당합니다.
- 서비스 및 리소스에 대한 사용자 액세스 상세 제어

예를 들어에서 IAM AWS Cloud Map 을 사용하여 AWS 계정에서 새 네임스페이스를 생성하거나 인스턴스를 등록할 수 있는 사용자를 제어할 수 있습니다.

IAM에 대한 전반적인 정보는 다음 리소스를 참조하세요.

- 에 대한 자격 증명 및 액세스 관리 AWS Cloud Map
- AWS Identity and Access Management
- IAM 사용 설명서

AWS Cloud Map 요금

AWS Cloud Map 요금은 서비스 레지스트리에 등록한 리소스와 이를 검색하기 위해 수행한 API 호출을 기반으로 합니다. AWS Cloud Map 에서는 선결제가 없으며 사용한 만큼만 지불하면 됩니다.

경우에 따라 IP 주소를 사용하여 리소스에 대한 DNS 기반 검색을 활성화할 수 있습니다. 또한 인스턴스 검색에 API 호출을 사용하는지 DNS 쿼리를 사용하는지와 관계없이 Amazon Route 53 상태 확인을 사용하여 리소스에 대한 상태 확인을 활성화할 수 있습니다. Route 53 DNS 및 상태 확인 사용과 관련해 추가 비용이 발생합니다.

자세한 내용은 AWS Cloud Map 요금을 참조하세요.

AWS Cloud Map 및 AWS 클라우드 규정 준수

다양한 보안 규정 및 감사 표준 AWS Cloud Map 준수에 대한 자세한 내용은 다음 페이지를 참조하세요.

- AWS 클라우드 규정 준수
- AWS 규정 준수 프로그램 제공 범위 내 서비스

시작하기 AWS Cloud Map

다음 가이드에서는 AWS Cloud Map 네임스페이스를 사용하여 일반적인 작업을 AWS Cloud Map 사용하고 수행하도록를 설정하는 방법을 보여줍니다.

가이드 개요	자세히 알아보기
가입 AWS 및 사용 준비 AWS Cloud Map	를 사용하도록 설정 AWS Cloud Map
DNS 쿼리 및 API 호출을 사용하여 백엔드 서비스를 검색합니다.	DNS 쿼리 및 API 호출과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다.
DNS 쿼리 및 API 호출을 사용하여를 사용하는 백엔드 서비스를 검색합니다 AWS CLI.	를 사용하여 DNS 쿼리 및 API 호출과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다. AWS CLI
샘플 애플리케이션을 생성하고 코드에서 사용자 지정 속성을 사용하여 리소스를 검색합니다.	사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다.
샘플 애플리케이션을 생성하고 코드에서 사용자 지정 속성을 사용하여를 사용하여 리소스를 검색합니다 AWS CLI.	를 사용하여 사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다. AWS CLI

를 사용하도록 설정 AWS Cloud Map

다음 섹션의 개요와 절차는를 시작하고 사용을 AWS 준비하는 데 도움이 됩니다 AWS Cloud Map.

주제

- 에 가입 AWS
- API AWS CLI, AWS Tools for Windows PowerShell또는 AWS SDKs에 액세스
- AWS Command Line Interface 또는 설정 AWS Tools for Windows PowerShell
- AWS SDK 다운로드

에 가입 AWS

에 가입 AWS 계정

이 없는 경우 다음 단계를 AWS 계정완료하여 생성합니다.

에 가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따르세요.

   등록 절차 중 전화 또는 텍스트 메시지를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 가입하면 AWS 계정AWS 계정 루트 사용자가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/으로 이동하고 내 계정을 선택하여 현재 계정 활동을 확인하고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 AWS 계정보호 AWS IAM Identity Center, AWS 계정 루트 사용자활성화 및 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자AWS Management Console로에 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 사용 설명서의 루트 사용자로 로그인을 참조하세요.
2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자(콘솔)에 대한 가상 MFA 디바이스 활성화를 참조하세요.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 자격 증명 소스 IAM Identity Center 디렉터리 로 사용하는 방법에 대한 자습서는 사용 AWS IAM Identity Center 설명서의 기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리 참조하세요.

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 사용 설명서의 AWS 액세스 포털에 로그인을 참조하세요.

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.
2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 그룹 추가를 참조하세요.

API AWS CLI, AWS Tools for Windows PowerShell또는 AWS SDKs에 액세스

API, AWS CLI AWS Tools for Windows PowerShell또는 AWS SDKs를 사용하려면 액세스 키를 생성해야 합니다. 이들 키는 액세스 키 ID 및 보안 액세스 키로 이루어져 있는데, 이를 사용하여 AWS에 보내는 프로그래밍 방식의 요청에 서명할 수 있습니다.

사용자는 AWS 외부에서와 상호 작용하려는 경우 프로그래밍 방식으로 액세스해야 합니다 AWS Management Console. 프로그래밍 방식 액세스를 부여하는 방법은에 액세스하는 사용자 유형에 따라 다릅니다 AWS.

사용자에게 프로그래밍 방식 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장) 콘솔 자격 증명을 임시 자격 증명으로 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 AWS 로컬 개발을 위한 로그인을 참조하세요. • AWS SDKs 경우 SDK 및 도구 참조 안내서의 AWS 로컬 개발을 위한 로그인을 참조하세요. AWS SDKs
작업 인력 ID (IAM Identity Center에서 관리되는 사용자)	임시 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 를 사용하도록 구성을 AWS IAM Identity Center 참조하세요. • AWS SDKs, 도구 및 AWS APIs 경우 SDK 및 도구 참조 안내서의 IAM Identity Center 인증을 참조하세요. AWS SDKs
IAM	임시 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	IAM 사용 설명서의 AWS 리소스에서 임시 자격 증명 사용의 지침을 따릅니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 IAM 사용자 자격 증명을 사용하여 인증을 참조하세요. • AWS SDKs 및 도구의 경우 SDK 및 도구 참조 안내서의 장기 자격 증명을 사용하여 인증을 참조하세요. AWS SDKs • AWS APIs 경우 IAM 사용 설명서의 IAM 사용자의 액세스 키 관리를 참조하세요.

AWS Command Line Interface 또는 설정 AWS Tools for Windows PowerShell

AWS Command Line Interface (AWS CLI)는 AWS 서비스 관리를 위한 통합 도구입니다. 설치 및 구성 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 의 최신 버전 설치 또는 업데이트를 AWS CLI AWS CLI참조하세요.

Windows PowerShell을 사용한 경험이 있다면 AWS Tools for Windows PowerShell을 사용하는 것이 좋습니다. 자세한 내용은 AWS Tools for PowerShell 사용 설명서에서 AWS Tools for Windows PowerShell설정을 참조하세요.

AWS SDK 다운로드

SDK를 AWS 제공하는 프로그래밍 언어를 사용하는 경우 AWS Cloud Map API 대신 SDK를 사용하는 것이 좋습니다. SDK를 사용하면 여러 가지 장점이 있습니다. SDK는 인증을 단순화하고, 개발 환경에 쉽게 통합되며, AWS Cloud Map 명령에 액세스할 수 있도록 합니다. 자세한 내용은 Amazon Web Services용 도구를 참조하세요.

DNS 쿼리 및 API 호출과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다.

다음 자습서에서는 두 개의 백엔드 서비스가 있는 마이크로서비스 아키텍처를 시뮬레이션합니다. 첫 번째 서비스는 DNS 쿼리를 사용하여 검색할 수 있습니다. 두 번째 서비스는 AWS Cloud Map API만 사용하여 검색할 수 있습니다.

> **Note**
>
> 도메인 이름 및 IP 주소와 같은 리소스 세부 정보는 시뮬레이션 목적으로만 사용됩니다. 인터넷을 통해 해결할 수 없습니다.

이 자습서의 end-to-end AWS CLI 버전은 섹션을 참조하세요를 사용하여 DNS 쿼리 및 API 호출과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다. AWS CLI.

사전 조건

자습서를 성공적으로 완료하려면 다음 사전 조건을 충족해야 합니다.

- 시작하기 전에 를 사용하도록 설정 AWS Cloud Map의 단계를 완료해야 합니다.
- 아직를 설치하지 않은 경우 의 최신 버전 설치 또는 업데이트 AWS CLI 단계에 AWS Command Line Interface따라 설치합니다.

  이 자습서에서는 명령을 실행할 셸 또는 명령줄 터미널이 필요합니다. Linux 및 macOS에서는 선호하는 셸과 패키지 관리자를 사용합니다.

  > **Note**
  >
  > Windows에서는 Lambda와 함께 일반적으로 사용하는 일부 Bash CLI 명령(예:`zip`)은 운영 체제의 기본 제공 터미널에서 지원되지 않습니다. Ubuntu와 Bash의 Windows 통합 버전을 가져오려면 Linux용 Windows Subsystem을 설치합니다.

- 자습서에는 `dig` DNS 조회 유틸리티 명령이 있는 로컬 환경이 필요합니다.

1단계: AWS Cloud Map 네임스페이스 생성

이 단계에서는 퍼블릭 AWS Cloud Map 네임스페이스를 생성합니다.는 사용자를 대신하여 동일한 이름으로 Route 53 호스팅 영역을 AWS Cloud Map 생성합니다. 이렇게 하면 퍼블릭 DNS 레코드를 사용하거나 AWS Cloud Map API 호출을 사용하여이 네임스페이스에서 생성된 서비스 인스턴스를 검색할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. Create namespace(네임스페이스 생성)를 선택합니다.
3. 네임스페이스 이름에를 지정합니다`cloudmap-tutorial.com`.

   > **Note**
   >
   > 프로덕션 환경에서 이를 사용하려는 경우 소유하거나 액세스할 수 있는 도메인의 이름을 지정했는지 확인해야 합니다. 그러나이 자습서의 목적상 사용 중인 실제 도메인일 필요는 없습니다.

4. (선택 사항) 네임스페이스 설명에서 네임스페이스를 사용할 대상에 대한 설명을 지정합니다.
5. 인스턴스 검색에서 API 호출 및 퍼블릭 DNS 쿼리를 선택합니다.
6. 나머지 기본값을 그대로 두고 네임스페이스 생성을 선택합니다.

2단계: AWS Cloud Map 서비스 생성

이 단계에서는 두 개의 서비스를 생성합니다. 첫 번째 서비스는 퍼블릭 DNS 및 API 호출을 사용하여 검색할 수 있습니다. 두 번째 서비스는 API 호출만 사용하여 검색할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 네임스페이스를 선택하여 생성한 네임스페이스를 나열합니다.
3. 네임스페이스 목록에서 네임`cloudmap-tutorial.com`스페이스를 선택하고 세부 정보 보기를 선택합니다.
4. 서비스 섹션에서 서비스 생성을 선택하고 다음을 수행하여 첫 번째 서비스를 생성합니다.

a. 서비스 이름에 `public-service`를 입력합니다. 서비스 이름은가 AWS Cloud Map 생성하는 DNS 레코드에 적용됩니다. 사용되는 형식은 입니다*<service-name>*.*<namespace-name>*.

b. 서비스 검색 구성에서 API 및 DNS를 선택합니다.

c. DNS 구성 섹션의 라우팅 정책에서 다중 값 응답 라우팅을 선택합니다.

> **Note**
>
> 콘솔을 선택하면이 값이 MULTIVALUE로 변환됩니다. 사용 가능한 라우팅 옵션에 대한 자세한 내용은 Route 53 개발자 안내서의 라우팅 정책 선택을 참조하세요.

d. 나머지 기본값을 그대로 두고 서비스 생성을 선택하면 네임스페이스 세부 정보 페이지로 돌아갑니다.

5. 서비스 섹션에서 서비스 생성을 선택하고 다음을 수행하여 두 번째 서비스를 생성합니다.

a. 서비스 이름에 `backend-service`를 입력합니다.

b. 서비스 검색 구성에서 API만 선택합니다.

c. 나머지 기본값을 그대로 두고 서비스 생성을 선택합니다.

3단계: AWS Cloud Map 서비스 인스턴스 등록

이 단계에서는 네임스페이스의 각 서비스에 대해 하나씩 두 개의 서비스 인스턴스를 생성합니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 네임스페이스 목록에서 1단계에서 생성한 네임스페이스를 선택하고 세부 정보 보기를 선택합니다.
3. 네임스페이스 세부 정보 페이지의 서비스 목록에서 `public-service` 서비스를 선택하고 세부 정보 보기를 선택합니다.
4. 서비스 인스턴스 섹션에서 서비스 인스턴스 등록을 선택하고 다음을 수행하여 첫 번째 서비스 인스턴스를 생성합니다.

a. 서비스 인스턴스 ID에를 지정합니다`first`.

b. IPv4 주소에를 지정합니다`192.168.2.1`.

c. 나머지 기본값을 그대로 두고 서비스 인스턴스 등록을 선택합니다.

5. 페이지 상단의 브레드크럼을 사용하여 cloudmap-tutorial.com 선택하여 네임스페이스 세부 정보 페이지로 돌아갑니다.
6. 네임스페이스 세부 정보 페이지의 서비스 목록에서 백엔드 서비스 서비스를 선택하고 세부 정보 보기를 선택합니다.
7. 서비스 인스턴스 섹션에서 서비스 인스턴스 등록을 선택하고 다음을 수행하여 두 번째 서비스 인스턴스를 생성합니다.

   a. 서비스 인스턴스 ID에서이 인스턴스가 두 번째 서비스 인스턴스임을 나타내`second`도록를 지정합니다.

   b. 인스턴스 유형에서 다른 리소스에 대한 식별 정보를 선택합니다.

   c. 사용자 지정 속성의 경우를 키`service-name`로,를 값으로 사용하여 키-값 페어`backend`를 추가합니다.

   d. 서비스 인스턴스 등록을 선택합니다.

4단계: AWS Cloud Map 서비스 인스턴스 검색

이제 AWS Cloud Map 네임스페이스, 서비스 및 서비스 인스턴스가 생성되었으므로 인스턴스를 검색하여 모든 것이 작동하는지 확인할 수 있습니다. `dig` 명령을 사용하여 퍼블릭 DNS 설정을 확인하고 AWS Cloud Map API를 사용하여 백엔드 서비스를 확인합니다. `dig` 명령에 대한 자세한 내용은 dig - DNS 조회 유틸리티를 참조하세요.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/route53/ Route 53 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Hosted Zones(호스팅 영역)를 선택합니다.
3. cloudmap-tutorial.com 호스팅 영역을 선택합니다. 그러면 호스팅 영역 세부 정보가 별도의 창에 표시됩니다. 다음 단계에서 사용할 호스팅 영역과 연결된 이름 서버를 기록해 둡니다.
4. dig 명령과 호스팅 영역의 Route 53 이름 서버 중 하나를 사용하여 서비스 인스턴스의 DNS 레코드를 쿼리합니다.

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

   출력의 `ANSWER SECTION`에는 `public-service` 서비스와 연결한 IPv4 주소가 표시되어야 합니다.

   ```
   ;; ANSWER SECTION:
   ```

```
public-service.cloudmap-tutorial.com. 300 IN A 192.168.2.1
```

5. 를 사용하여 두 번째 서비스 인스턴스의 속성을 AWS CLI쿼리합니다.

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

출력에는 서비스와 연결한 속성이 키-값 페어로 표시됩니다.

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

5단계: 리소스 정리

자습서를 완료하면 resources. AWS Cloud Map requires를 삭제할 수 있습니다. 먼저 서비스 인스턴스, 서비스, 네임스페이스 순서로 정리해야 합니다.이 단계를 거치면 AWS Cloud Map 가 사용자를 대신하여 Route 53 리소스를 정리합니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 네임스페이스 목록에서 네임`cloudmap-tutorial.com`스페이스를 선택하고 세부 정보 보기를 선택합니다.
3. 네임스페이스 세부 정보 페이지의 서비스 목록에서 `public-service` 서비스를 선택하고 세부 정보 보기를 선택합니다.
4. 서비스 인스턴스 섹션에서 `first` 인스턴스를 선택하고 등록 취소를 선택합니다.

5. 페이지 상단의 브레드크럼을 사용하여 cloudmap-tutorial.com 선택하여 네임스페이스 세부 정보 페이지로 돌아갑니다.
6. 네임스페이스 세부 정보 페이지의 서비스 목록에서 퍼블릭 서비스 및 삭제를 선택합니다.
7. 에 대해 3~6단계를 반복합니다`backend-service`.
8. 왼쪽 탐색 창에서 네임스페이스를 선택합니다.
9. `cloudmap-tutorial.com` 네임스페이스를 선택하고 삭제를 선택합니다.

> **Note**
>
> 는 사용자를 대신하여 Route 53 리소스를 AWS Cloud Map 정리하지만 Route 53 콘솔로 이동하여 `cloudmap-tutorial.com` 호스팅 영역이 삭제되었는지 확인할 수 있습니다.

를 사용하여 DNS 쿼리 및 API 호출과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다. AWS CLI

이 자습서에서는 AWS Command Line Interface (CLI)를 사용하여 AWS Cloud Map 서비스 검색을 사용하는 방법을 보여줍니다. 두 개의 백엔드 서비스로 마이크로서비스 아키텍처를 생성합니다. 하나는 DNS 쿼리를 사용하여 검색할 수 있고 다른 하나는 AWS Cloud Map API만 사용하여 검색할 수 있습니다.

AWS Cloud Map 콘솔 단계가 포함된 자습서는 단원을 참조하십시오DNS 쿼리 및 API 호출과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다..

사전 조건

자습서를 성공적으로 완료하려면 다음 사전 조건을 충족해야 합니다.

- 시작하기 전에 를 사용하도록 설정 AWS Cloud Map의 단계를 완료해야 합니다.
- 아직를 설치하지 않은 경우 의 최신 버전 설치 또는 업데이트 AWS CLI 단계에 AWS Command Line Interface따라 설치합니다.

  이 자습서에서는 명령을 실행할 셸 또는 명령줄 터미널이 필요합니다. Linux 및 macOS에서는 선호하는 셸과 패키지 관리자를 사용합니다.

> **Note**
>
> Windows에서는 Lambda와 함께 일반적으로 사용하는 일부 Bash CLI 명령(예:`zip`)은 운영 체제의 기본 제공 터미널에서 지원되지 않습니다. Ubuntu와 Bash의 Windows 통합 버전을 가져오려면 Linux용 Windows Subsystem을 설치합니다.

- 자습서에는 `dig` DNS 조회 유틸리티 명령이 있는 로컬 환경이 필요합니다.

AWS Cloud Map 네임스페이스 생성

먼저 퍼블릭 AWS Cloud Map 네임스페이스를 생성합니다. AWS Cloud Map 는 동일한 이름으로 Route 53 호스팅 영역을 생성하여 DNS 레코드와 API 호출을 통해 서비스 검색을 활성화합니다.

1. 퍼블릭 DNS 네임스페이스를 생성합니다.

```
aws servicediscovery create-public-dns-namespace \
    --name cloudmap-tutorial.com \
    --creator-request-id cloudmap-tutorial-request-1 \
    --region us-east-2
```

명령은 네임스페이스 생성 상태를 확인하는 데 사용할 수 있는 작업 ID를 반환합니다.

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9xmplyzd"
}
```

2. 작업 상태를 확인하여 네임스페이스가 성공적으로 생성되었는지 확인합니다.

```
aws servicediscovery get-operation \
    --operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9xmplyzd \
    --region us-east-2
```

3. 작업이 성공하면 네임스페이스 ID를 가져옵니다.

```
aws servicediscovery list-namespaces \
    --region us-east-2 \
    --query "Namespaces[?Name=='cloudmap-tutorial.com'].Id" \
    --output text
```

이 명령은 후속 단계에 필요한 네임스페이스 ID를 반환합니다.

```
ns-abcd1234xmplefgh
```

AWS Cloud Map 서비스 생성

이제 네임스페이스 내에 두 개의 서비스를 생성합니다. 첫 번째 서비스는 DNS 및 API 호출을 모두 사용하여 검색할 수 있고, 두 번째 서비스는 API 호출만 사용하여 검색할 수 있습니다.

1. DNS 검색이 활성화된 첫 번째 서비스를 생성합니다.

```
aws servicediscovery create-service \
    --name public-service \
    --namespace-id ns-abcd1234xmplefgh \
    --dns-config "RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=300}]" \
    --region us-east-2
```

명령은 생성된 서비스에 대한 세부 정보를 반환합니다.

```
{
    "Service": {
        "Id": "srv-abcd1234xmplefgh",
        "Arn": "arn:aws:servicediscovery:us-east-2:123456789012:service/srv-
abcd1234xmplefgh",
        "Name": "public-service",
        "NamespaceId": "ns-abcd1234xmplefgh",
        "DnsConfig": {
            "NamespaceId": "ns-abcd1234xmplefgh",
            "RoutingPolicy": "MULTIVALUE",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 300
                }
            ]
        },
        "CreateDate": 1673613600.000,
        "CreatorRequestId": "public-service-request"
    }
```

```
}
```

2. API 전용 검색을 사용하여 두 번째 서비스를 생성합니다.

```
aws servicediscovery create-service \
    --name backend-service \
    --namespace-id ns-abcd1234xmplefgh \
    --type HTTP \
    --region us-east-2
```

명령은 생성된 서비스에 대한 세부 정보를 반환합니다.

```
{
    "Service": {
        "Id": "srv-ijkl5678xmplmnop",
        "Arn": "arn:aws:servicediscovery:us-east-2:123456789012:service/srv-
ijkl5678xmplmnop",
        "Name": "backend-service",
        "NamespaceId": "ns-abcd1234xmplefgh",
        "Type": "HTTP",
        "CreateDate": 1673613600.000,
        "CreatorRequestId": "backend-service-request"
    }
}
```

AWS Cloud Map 서비스 인스턴스 등록

그런 다음 각 서비스에 대한 서비스 인스턴스를 등록합니다. 이러한 인스턴스는 검색될 실제 리소스를 나타냅니다.

1. DNS 검색을 위해 첫 번째 인스턴스를 IPv4 주소로 등록합니다.

```
aws servicediscovery register-instance \
    --service-id srv-abcd1234xmplefgh \
    --instance-id first \
    --attributes AWS_INSTANCE_IPV4=192.168.2.1 \
    --region us-east-2
```

명령은 작업 ID를 반환합니다.

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k9xmplyzd"
}
```

2. 작업 상태를 확인하여 인스턴스가 성공적으로 등록되었는지 확인합니다.

```
aws servicediscovery get-operation \
    --operation-id 4yejorelbukcjzpnr6tlmrghsjwpngf4-k9xmplyzd \
    --region us-east-2
```

3. API 검색을 위한 사용자 지정 속성으로 두 번째 인스턴스를 등록합니다.

```
aws servicediscovery register-instance \
    --service-id srv-ijkl5678xmplmnop \
    --instance-id second \
    --attributes service-name=backend \
    --region us-east-2
```

명령은 작업 ID를 반환합니다.

```
{
    "OperationId": "7zxcvbnmasdfghjklqwertyuiop1234-k9xmplyzd"
}
```

4. 작업 상태를 확인하여 인스턴스가 성공적으로 등록되었는지 확인합니다.

```
aws servicediscovery get-operation \
    --operation-id 7zxcvbnmasdfghjklqwertyuiop1234-k9xmplyzd \
    --region us-east-2
```

AWS Cloud Map 서비스 인스턴스 검색

이제 서비스 인스턴스를 생성하고 등록했으므로 DNS 쿼리와 AWS Cloud Map API를 모두 사용하여 인스턴스를 검색하여 모든 것이 작동하는지 확인할 수 있습니다.

1. 먼저 Route 53 호스팅 영역 ID를 가져옵니다.

```
aws route53 list-hosted-zones-by-name \
    --dns-name cloudmap-tutorial.com \
```

```
    --query "HostedZones[0].Id" \
    --output text
```

이렇게 하면 호스팅 영역 ID가 반환됩니다.

```
/hostedzone/Z1234ABCDXMPLEFGH
```

2. 호스팅 영역의 이름 서버를 가져옵니다.

```
aws route53 get-hosted-zone \
    --id Z1234ABCDXMPLEFGH \
    --query "DelegationSet.NameServers[0]" \
    --output text
```

그러면 이름 서버 중 하나가 반환됩니다.

```
ns-1234.awsdns-12.org
```

3. dig 명령을 사용하여 퍼블릭 서비스의 DNS 레코드를 쿼리합니다.

```
dig @ns-1234.awsdns-12.org public-service.cloudmap-tutorial.com
```

출력에는 서비스와 연결한 IPv4 주소가 표시되어야 합니다.

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

4. AWS CLI 를 사용하여 백엔드 서비스 인스턴스를 검색합니다.

```
aws servicediscovery discover-instances \
    --namespace-name cloudmap-tutorial.com \
    --service-name backend-service \
    --region us-east-2
```

출력에는 서비스와 연결한 속성이 표시됩니다.

```
{
    "Instances": [
        {
            "InstanceId": "second",
```

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

리소스 정리

자습서를 완료한 후에는 요금이 발생하지 않도록 리소스를 정리합니다. AWS Cloud Map 에서는 먼저 서비스 인스턴스, 서비스, 마지막으로 네임스페이스 순서로 리소스를 정리해야 합니다.

1. 첫 번째 서비스 인스턴스 등록을 취소합니다.

```
aws servicediscovery deregister-instance \
    --service-id srv-abcd1234xmplefgh \
    --instance-id first \
    --region us-east-2
```

2. 두 번째 서비스 인스턴스 등록을 취소합니다.

```
aws servicediscovery deregister-instance \
    --service-id srv-ijkl5678xmplmnop \
    --instance-id second \
    --region us-east-2
```

3. 퍼블릭 서비스를 삭제합니다.

```
aws servicediscovery delete-service \
    --id srv-abcd1234xmplefgh \
    --region us-east-2
```

4. 백엔드 서비스를 삭제합니다.

```
aws servicediscovery delete-service \
    --id srv-ijkl5678xmplmnop \
    --region us-east-2
```

5. 네임스페이스를 삭제합니다.

```
aws servicediscovery delete-namespace \
    --id ns-abcd1234xmplefgh \
    --region us-east-2
```

6. Route 53 호스팅 영역이 삭제되었는지 확인합니다.

```
aws route53 list-hosted-zones-by-name \
    --dns-name cloudmap-tutorial.com
```

사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다.

다음 자습서에서는 AWS Cloud Map API를 사용하여 검색할 수 있는 사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 보여줍니다. 이 자습서에서는를 사용하여 클라이언트 애플리케이션을 생성하고 실행하는 방법을 안내합니다 AWS CloudShell. 애플리케이션은 두 Lambda 함수를 사용하여 DynamoDB 테이블에 데이터를 쓴 다음 테이블에서 읽습니다. Lambda 함수 및 DynamoDB 테이블은에 서비스 인스턴스 AWS Cloud Map 로 등록됩니다. 클라이언트 애플리케이션 및 Lambda 함수의 코드는 사용자 지정 속성을 사용하여 AWS Cloud Map 작업을 수행하는 데 필요한 리소스를 검색합니다.

이 자습서의 AWS CLI기반 버전은 섹션을 참조하세요를 사용하여 사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다. AWS CLI.

> **Important**
>
> 워크숍 중에 계정에 AWS 비용이 발생하는 AWS 리소스를 생성합니다. 워크숍을 마치는 즉시 리소스를 정리하여 비용을 최소화하는 것이 좋습니다.

사전 조건

시작하기 전에 를 사용하도록 설정 AWS Cloud Map의 단계를 완료해야 합니다.

1단계: AWS Cloud Map 네임스페이스 생성

이 단계에서는 AWS Cloud Map 네임스페이스를 생성합니다. 네임스페이스는 애플리케이션의 서비스를 그룹화하는 데 사용되는 구문입니다. 네임스페이스를 생성할 때 리소스를 검색할 수 있는 방법을 지정합니다. 이 단계에서 생성된 네임스페이스에 생성된 리소스는 사용자 지정 속성을 사용하여 AWS Cloud Map API 직접 호출을 통해 검색할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. Create namespace(네임스페이스 생성)를 선택합니다.
3. 네임스페이스 이름에를 지정합니다`cloudmap-tutorial`.
4. (선택 사항) 네임스페이스 설명에서 네임스페이스를 사용할 대상에 대한 설명을 지정합니다.
5. 인스턴스 검색에서 API 호출을 선택합니다.
6. 나머지 기본값을 그대로 두고 네임스페이스 생성을 선택합니다.

2단계: DynamoDB 테이블 생성

이 단계에서는 DynamoDB 테이블을 생성합니다. 이 테이블은 다음 단계에서 생성할 샘플 애플리케이션의 데이터를 저장하고 검색하는 데 사용됩니다.

DynamoDB를 생성하는 방법에 대한 자세한 내용은 DynamoDB 개발자 안내서의 1단계: DynamoDB에서 테이블 생성을 참조하고 다음 표를 사용하여 지정할 옵션을 결정합니다. DynamoDB

옵션	값
테이블 이름	클라우드맵
파티션 키	id

나머지 설정의 기본값을 유지하고 테이블을 생성합니다.

3단계: AWS Cloud Map 데이터 서비스 생성 및 DynamoDB 테이블을 인스턴스로 등록

이 단계에서는 AWS Cloud Map 서비스를 생성한 다음 마지막 단계에서 생성한 DynamoDB 테이블을 서비스 인스턴스로 등록합니다.

1. https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 네임스페이스 목록에서 네임`cloudmap-tutorial`스페이스를 선택하고 세부 정보 보기를 선택합니다.
3. 서비스 섹션에서 서비스 생성을 선택하고 다음을 수행합니다.
   a. 서비스 이름에 `data-service`를 입력합니다.
   b. 나머지 기본값을 그대로 두고 서비스 생성을 선택합니다.
4. 서비스 섹션에서 `data-service` 서비스를 선택하고 세부 정보 보기를 선택합니다.
5. 서비스 인스턴스 섹션에서 서비스 인스턴스 등록을 선택합니다.
6. 서비스 인스턴스 등록 페이지에서 다음을 수행합니다.
   a. 인스턴스 유형에서 다른 리소스에 대한 식별 정보를 선택합니다.
   b. 서비스 인스턴스 ID에를 지정합니다`data-instance`.
   c. 사용자 지정 속성 섹션에서 키-값 페어를 지정합니다. 키 = `tablename`, 값 = `cloudmap`.

4단계: AWS Lambda 실행 역할 생성

이 단계에서는 다음 단계에서 AWS Lambda 함수가 사용하는 IAM 역할을 생성합니다. 역할이이 자습서에만 사용되고 나중에 삭제할 수 있으므로 IAM 역할의 이름을 `cloudmap-tutorial-role` 지정하고 권한 경계를 생략할 수 있습니다.

Lambda에 대한 서비스 역할을 생성하려면(IAM 콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.
3. 신뢰할 수 있는 엔터티 유형에서 AWS 서비스를 선택합니다.
4. 서비스 또는 사용 사례에서 Lambda를 선택한 다음 Lambda 사용 사례를 선택합니다.
5. 다음을 선택합니다.
6. `PowerUserAccess` 정책을 검색하고 옆에 있는 상자를 선택한 후 다음을 선택합니다.
7. 다음을 선택합니다.
8. 역할 이름에를 지정합니다`cloudmap-tutorial-role`.
9. 역할을 검토한 다음 역할 생성을 선택합니다.

5단계: Lambda 함수를 생성하여 데이터 쓰기

이 단계에서는 AWS Cloud Map API를 사용하여 생성한 AWS Cloud Map 서비스를 쿼리하여 DynamoDB 테이블에 데이터를 쓰는 Lambda 함수를 처음부터 생성합니다.

Lambda 함수 생성에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 콘솔을 사용하여 Lambda 함수 생성을 참조하고 다음 표를 사용하여 지정하거나 선택할 옵션을 결정합니다.

옵션	값
함수 이름	writefunction
런타임	Python 3.12
아키텍처	x86_64
권한	기존 역할 사용
기존 역할	cloudmap-tutorial-role

함수를 생성한 후 다음 Python 코드를 반영하도록 예제 코드를 업데이트한 다음 함수를 배포합니다. DynamoDB 테이블에 대해 생성한 AWS Cloud Map 서비스 인스턴스와 연결한 `datatable` 사용자 지정 속성을 지정합니다. 함수는 1에서 100 사이의 임의의 숫자인 키를 생성하고 호출 시 함수에 전달되는 값과 연결합니다.

```
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')

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

    dynamodbclient = boto3.resource('dynamodb')
```

```
    table = dynamodbclient.Table(tablename)

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

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

함수를 배포한 후 제한 시간 오류를 방지하려면 함수 제한 시간을 5초로 업데이트합니다. 자세한 내용은 AWS Lambda 개발자 가이드의 Lambda 함수 제한 시간 구성을 참조하세요.

6단계: AWS Cloud Map 앱 서비스 생성 및 Lambda 쓰기 함수를 인스턴스로 등록

이 단계에서는 AWS Cloud Map 서비스를 생성한 다음 Lambda 쓰기 함수를 서비스 인스턴스로 등록합니다.

1. https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 네임스페이스를 선택합니다.
3. 네임스페이스 목록에서 네임`cloudmap-tutorial`스페이스를 선택하고 세부 정보 보기를 선택합니다.
4. 서비스 섹션에서 서비스 생성을 선택하고 다음을 수행합니다.
    a. 서비스 이름에 `app-service`를 입력합니다.
    b. 나머지 기본값을 그대로 두고 서비스 생성을 선택합니다.
5. 서비스 섹션에서 `app-service` 서비스를 선택하고 세부 정보 보기를 선택합니다.
6. 서비스 인스턴스 섹션에서 서비스 인스턴스 등록을 선택합니다.
7. 서비스 인스턴스 등록 페이지에서 다음을 수행합니다.
    a. 인스턴스 유형에서 다른 리소스에 대한 식별 정보를 선택합니다.
    b. 서비스 인스턴스 ID에를 지정합니다`write-instance`.
    c. 사용자 지정 속성 섹션에서 다음 키-값 페어를 지정합니다.
        - 키 = `action`, 값 = `write`

- 키 = `functionname`, 값 = `writefunction`

7단계: Lambda 함수를 생성하여 데이터 읽기

이 단계에서는 생성한 DynamoDB 테이블에 데이터를 쓰는 Lambda 함수를 처음부터 생성합니다.

Lambda 함수 생성에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 콘솔을 사용하여 Lambda 함수 생성을 참조하고 다음 표를 사용하여 지정하거나 선택할 옵션을 결정합니다.

옵션	값
함수 이름	읽기 함수
런타임	Python 3.12
아키텍처	x86_64
권한	기존 역할 사용
기존 역할	cloudmap-tutorial-role

함수를 생성한 후 다음 Python 코드를 반영하도록 예제 코드를 업데이트한 다음 함수를 배포합니다. 함수는 테이블 amd를 스캔하여 모든 항목을 반환합니다.

```
import json
import boto3

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

    response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
 ServiceName='data-service')

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

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

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

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

함수를 배포한 후 제한 시간 오류를 방지하려면 함수 제한 시간을 5초로 업데이트합니다. 자세한 내용은 AWS Lambda 개발자 가이드의 Lambda 함수 제한 시간 구성을 참조하세요.

8단계: Lambda 읽기 함수를 AWS Cloud Map 서비스 인스턴스로 등록

이 단계에서는 Lambda 읽기 함수를 이전에 생성한 서비스의 `app-service` 서비스 인스턴스로 등록합니다.

1. https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 네임스페이스를 선택합니다.
3. 네임스페이스 목록에서 네임`cloudmap-tutorial`스페이스를 선택하고 세부 정보 보기를 선택합니다.
4. 서비스 섹션에서 `app-service` 서비스를 선택하고 세부 정보 보기를 선택합니다.
5. 서비스 인스턴스 섹션에서 서비스 인스턴스 등록을 선택합니다.
6. 서비스 인스턴스 등록 페이지에서 다음을 수행합니다.
    a. 인스턴스 유형에서 다른 리소스에 대한 식별 정보를 선택합니다.
    b. 서비스 인스턴스 ID에를 지정합니다`read-instance`.
    c. 사용자 지정 속성 섹션에서 다음 키-값 페어를 지정합니다.
        - 키 = `action`, 값 = `read`
        - 키 = `functionname`, 값 = `readfunction`

9단계:에서 읽기 및 쓰기 클라이언트 생성 및 실행 AWS CloudShell

코드를 AWS CloudShell 사용하여에서 클라이언트 애플리케이션을 생성하고 실행하여에서 구성한 서비스를 검색 AWS Cloud Map 하고 이러한 서비스를 호출할 수 있습니다.

1. https://console.aws.amazon.com/cloudshell/ AWS CloudShell 콘솔을 엽니다.
2. 다음 명령을 사용하여 라는 파일을 생성합니다`writefunction.py`.

```
vim writeclient.py
```

3. writeclient.py 파일에서 i 버튼을 눌러 삽입 모드로 들어갑니다. 그런 다음 다음 코드를 복사하여 붙여 넣습니다. 이 코드는 app-service 서비스name=writeservice에서 사용자 지정 속성을 검색하여 데이터를 쓰는 Lambda 함수를 검색합니다. DynamoDB 테이블에 데이터를 쓰는 Lambda 함수의 이름이 반환됩니다. 그런 다음 Lambda 함수가 호출되어 테이블에 기록된 샘플 페이로드를 값으로 전달합니다.

```
import boto3

serviceclient = boto3.client('servicediscovery')

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

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

lambdaclient = boto3.client('lambda')

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

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

4. 이스케이프 키를 누르고를 입력한 다음 Enter 키를 :wq눌러 파일을 저장하고 종료합니다.
5. 다음 명령을 사용하여 Python 코드를 실행합니다.

```
python3 writeclient.py
```

출력은 다음과 유사한 200 응답이어야 합니다.

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

6. 이전 단계에서 쓰기에 성공했는지 확인하려면 읽기 클라이언트를 생성합니다.

a. 다음 명령을 사용하여 라는 파일을 생성합니다`readfunction.py`.

```
vim readclient.py
```

b. `readclient.py` 파일에서 `i` 버튼을 눌러 삽입 모드로 들어갑니다. 그런 다음 다음 코드를 복사하여 붙여 넣습니다. 이 코드는 테이블을 스캔하고 이전 단계에서 작성한 값을 테이블에 반환합니다.

```
import boto3

serviceclient = boto3.client('servicediscovery')

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

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

lambdaclient = boto3.client('lambda')

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

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

c. 이스케이프 키를 누르고를 입력한 다음 Enter 키를 `:wq`눌러 파일을 저장하고 종료합니다.

d. 다음 명령을 사용하여 Python 코드를 실행합니다.

```
python3 readclient.py
```

출력은 다음과 비슷해야 하며,를 실행하여 테이블에 기록된 값과 Lambda 쓰기 함수에서 생성된 `writefunction.py` 임의 키를 나열합니다.

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

```
amzn-requestid\\": \\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\",
 \\"x-amz-crc32\\": \\"1163081893\\"}, \\"RetryAttempts\\": 0}}"}'
```

10단계: 리소스 정리

자습서를 완료한 후 추가 요금이 발생하지 않도록 리소스를 삭제합니다. AWS Cloud Map 에서는 먼저 서비스 인스턴스, 서비스, 마지막으로 네임스페이스를 역순으로 정리해야 합니다. 다음 단계에서는 자습서에 사용된 AWS Cloud Map 리소스를 정리하는 방법을 안내합니다.

AWS Cloud Map 리소스를 삭제하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 네임스페이스 목록에서 네임`cloudmap-tutorial`스페이스를 선택하고 세부 정보 보기를 선택합니다.
3. 네임스페이스 세부 정보 페이지의 서비스 목록에서 `data-service` 서비스를 선택하고 세부 정보 보기를 선택합니다.
4. 서비스 인스턴스 섹션에서 `data-instance` 인스턴스를 선택하고 등록 취소를 선택합니다.
5. 페이지 상단의 브레드크럼을 사용하여 cloudmap-tutorial.com 선택하여 네임스페이스 세부 정보 페이지로 돌아갑니다.
6. 네임스페이스 세부 정보 페이지의 서비스 목록에서 데이터 서비스 서비스를 선택하고 삭제를 선택합니다.
7. 서비스 및 `write-instance` 및 `app-service` 서비스 `read-instance` 인스턴스에 대해 3~6단계를 반복합니다.
8. 왼쪽 탐색 창에서 네임스페이스를 선택합니다.
9. `cloudmap-tutorial` 네임스페이스를 선택하고 삭제를 선택합니다.

다음 표에는 자습서에서 사용되는 다른 리소스를 삭제하기 위해 따를 수 있는 절차가 나열되어 있습니다.

Resource	단계(Steps)	
DynamoDB 테이블	6단계: (선택 사항) Amazon DynamoDB 개발자 안내서의	

Resource	단계(Steps)	
	DynamoDB 테이블을 삭제하여 리소스 정리 DynamoDB	
Lambda 함수 및 관련 IAM 실행 역할	AWS Lambda 개발자 안내서의 정리	

를 사용하여 사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다. AWS CLI

이 자습서에서는 사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 보여줍니다. 를 사용하여 사용자 지정 속성을 사용하여 리소스를 동적으로 AWS Cloud Map 검색하는 마이크로서비스 애플리케이션을 생성합니다. 애플리케이션은 모든 리소스가 등록된 상태에서 DynamoDB 테이블에 데이터를 쓰고 읽는 두 Lambda 함수로 구성됩니다 AWS Cloud Map.

자습서 AWS Management Console 버전은 섹션을 참조하세요사용자 지정 속성과 함께 AWS Cloud Map 서비스 검색을 사용하는 방법을 알아봅니다..

사전 조건

이 자습서를 시작하기 전에의 단계를 완료합니다를 사용하도록 설정 AWS Cloud Map.

AWS Cloud Map 네임스페이스 생성

네임스페이스는 애플리케이션의 서비스를 그룹화하는 데 사용되는 구문입니다. 이 단계에서는 AWS Cloud Map API 호출을 통해 리소스를 검색할 수 있는 네임스페이스를 생성합니다.

1. 다음 명령을 실행하여 HTTP 네임스페이스를 생성합니다.

```
aws servicediscovery create-http-namespace \
  --name cloudmap-tutorial \
  --creator-request-id cloudmap-tutorial-request
```

명령은 작업 ID를 반환합니다. 다음 명령을 사용하여 작업 상태를 확인할 수 있습니다.

```
aws servicediscovery get-operation \
  --operation-id operation-id
```

2. 네임스페이스가 생성되면 후속 명령에 사용할 네임스페이스의 ID를 검색할 수 있습니다.

```
aws servicediscovery list-namespaces \
  --query "Namespaces[?Name=='cloudmap-tutorial'].Id" \
  --output text
```

3. 나중에 사용할 수 있도록 네임스페이스 ID를 변수에 저장합니다.

```
NAMESPACE_ID=$(aws servicediscovery list-namespaces \
  --query "Namespaces[?Name=='cloudmap-tutorial'].Id" \
  --output text)
```

DynamoDB 테이블 생성

그런 다음 애플리케이션에 대한 데이터를 저장할 DynamoDB 테이블을 생성합니다.

1. 다음 명령을 실행하여 테이블을 생성합니다.

```
aws dynamodb create-table \
  --table-name cloudmap \
  --attribute-definitions AttributeName=id,AttributeType=S \
  --key-schema AttributeName=id,KeyType=HASH \
  --billing-mode PAY_PER_REQUEST
```

2. 계속하기 전에 테이블이 활성화될 때까지 기다립니다.

```
aws dynamodb wait table-exists --table-name cloudmap
```

   이 명령은 테이블이 완전히 생성되고 사용할 준비가 될 때까지 기다립니다.

AWS Cloud Map 데이터 서비스 생성 및 DynamoDB 테이블 등록

이제 네임스페이스에서 데이터 스토리지 리소스를 나타내는 서비스를 생성합니다.

1. 다음 명령을 실행하여 데이터 스토리지 리소스에 대한 AWS Cloud Map 서비스를 생성합니다.

```
aws servicediscovery create-service \
  --name data-service \
  --namespace-id $NAMESPACE_ID \
```

```
  --creator-request-id data-service-request
```

2. 데이터 서비스의 서비스 ID를 가져옵니다.

```
DATA_SERVICE_ID=$(aws servicediscovery list-services \
  --query "Services[?Name=='data-service'].Id" \
  --output text)
```

3. 테이블 이름을 지정하는 사용자 지정 속성을 사용하여 DynamoDB 테이블을 서비스 인스턴스로 등록합니다.

```
aws servicediscovery register-instance \
  --service-id $DATA_SERVICE_ID \
  --instance-id data-instance \
  --attributes tablename=cloudmap
```

사용자 지정 속성을 `tablename=cloudmap` 사용하면 다른 서비스에서 DynamoDB 테이블 이름을 동적으로 검색할 수 있습니다.

Lambda 함수에 대한 IAM 역할 생성

Lambda 함수가 리소스에 액세스하는 AWS 데 사용할 IAM 역할을 생성합니다.

1. 다음 JSON을 사용하여 IAM 역할에 대한 신뢰 정책 문서를 생성합니다.

JSON

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

2. 다음 명령을 실행하여 신뢰 정책을 사용하여 IAM 역할을 생성합니다.

```
aws iam create-role \
  --role-name cloudmap-tutorial-role \
  --assume-role-policy-document file://lambda-trust-policy.json
```

3. 다음 JSON을 사용하여 최소 권한 권한이 있는 사용자 지정 IAM 정책에 대한 파일을 생성합니다.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "servicediscovery:DiscoverInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/cloudmap"
    }
  ]
}
```

4. 정책을 생성하고 IAM 역할에 연결합니다.

```
aws iam create-policy \
  --policy-name CloudMapTutorialPolicy \
  --policy-document file://cloudmap-policy.json

POLICY_ARN=$(aws iam list-policies \
  --query "Policies[?PolicyName=='CloudMapTutorialPolicy'].Arn" \
  --output text)

aws iam attach-role-policy \
  --role-name cloudmap-tutorial-role \
  --policy-arn $POLICY_ARN

aws iam attach-role-policy \
  --role-name cloudmap-tutorial-role \
  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
```

Lambda 함수를 생성하여 데이터 쓰기

DynamoDB 테이블에 데이터를 쓰는 Lambda 함수를 생성하려면 다음 단계를 따르세요.

1. 쓰기 함수에 대한 Python 파일을 생성합니다.

```
cat > writefunction.py << EOF
import json
import boto3
import random

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

        response = serviceclient.discover_instances(
            NamespaceName='cloudmap-tutorial',
            ServiceName='data-service')

        if not response.get("Instances"):
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "No instances found"})
            }
```

```
        tablename = response["Instances"][0]["Attributes"].get("tablename")
        if not tablename:
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "Table name attribute not found"})
            }

        dynamodbclient = boto3.resource('dynamodb')

        table = dynamodbclient.Table(tablename)

        # Validate input
        if not isinstance(event, str):
            return {
                'statusCode': 400,
                'body': json.dumps({"error": "Input must be a string"})
            }

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

        return {
            'statusCode': 200,
            'body': json.dumps(response)
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps({"error": str(e)})
        }
EOF
```

이 함수는 AWS Cloud Map 를 사용하여 사용자 지정 속성에서 DynamoDB 테이블 이름을 검색한 다음 테이블에 데이터를 씁니다.

2. Lambda 함수를 패키징하고 배포합니다.

```
zip writefunction.zip writefunction.py

ROLE_ARN=$(aws iam get-role --role-name cloudmap-tutorial-role \
  --query 'Role.Arn' --output text)

aws lambda create-function \
```

```
  --function-name writefunction \
  --runtime python3.12 \
  --role $ROLE_ARN \
  --handler writefunction.lambda_handler \
  --zip-file fileb://writefunction.zip \
  --architectures x86_64
```

3. 제한 시간 오류를 방지하려면 함수 제한 시간을 업데이트합니다.

```
aws lambda update-function-configuration \
  --function-name writefunction \
  --timeout 5
```

AWS Cloud Map 앱 서비스 생성 및 Lambda 쓰기 함수 등록

네임스페이스에 애플리케이션 함수를 나타내는 다른 서비스를 생성하려면 다음 단계를 따르세요.

1. 애플리케이션 함수에 대한 서비스를 생성합니다.

```
aws servicediscovery create-service \
  --name app-service \
  --namespace-id $NAMESPACE_ID \
  --creator-request-id app-service-request
```

2. 앱 서비스의 서비스 ID를 가져옵니다.

```
APP_SERVICE_ID=$(aws servicediscovery list-services \
  --query "Services[?Name=='app-service'].Id" \
  --output text)
```

3. Lambda 쓰기 함수를 사용자 지정 속성이 있는 서비스 인스턴스로 등록합니다.

```
aws servicediscovery register-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id write-instance \
  --attributes action=write,functionname=writefunction
```

사용자 지정 속성 `action=write` 및는 클라이언트가 목적에 따라이 함수를 검색할 수 `functionname=writefunction` 있도록 허용합니다.

Lambda 함수를 생성하여 데이터 읽기

DynamoDB 테이블에서 데이터를 읽는 Lambda 함수를 생성하려면 다음 단계를 따릅니다.

1. 읽기 함수에 대한 Python 파일을 생성합니다.

```
cat > readfunction.py << EOF
import json
import boto3

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

        response = serviceclient.discover_instances(
            NamespaceName='cloudmap-tutorial',
            ServiceName='data-service')

        if not response.get("Instances"):
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "No instances found"})
            }

        tablename = response["Instances"][0]["Attributes"].get("tablename")
        if not tablename:
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "Table name attribute not found"})
            }

        dynamodbclient = boto3.resource('dynamodb')

        table = dynamodbclient.Table(tablename)

        # Use pagination for larger tables
        response = table.scan(
            Select='ALL_ATTRIBUTES',
            Limit=50  # Limit results for demonstration purposes
        )

        # For production, you would implement pagination like this:
        # items = []
```

```
        # while 'LastEvaluatedKey' in response:
        #     items.extend(response['Items'])
        #     response = table.scan(
        #         Select='ALL_ATTRIBUTES',
        #         ExclusiveStartKey=response['LastEvaluatedKey']
        #     )
        # items.extend(response['Items'])

        return {
            'statusCode': 200,
            'body': json.dumps(response)
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps({"error": str(e)})
        }
EOF
```

또한이 함수는 AWS Cloud Map 를 사용하여 DynamoDB 테이블 이름을 검색한 다음 테이블에서 데이터를 읽습니다. 여기에는 오류 처리 및 페이지 매김 주석이 포함됩니다.

2. Lambda 함수를 패키징하고 배포합니다.

```
zip readfunction.zip readfunction.py

aws lambda create-function \
  --function-name readfunction \
  --runtime python3.12 \
  --role $ROLE_ARN \
  --handler readfunction.lambda_handler \
  --zip-file fileb://readfunction.zip \
  --architectures x86_64
```

3. 함수 제한 시간을 업데이트합니다.

```
aws lambda update-function-configuration \
  --function-name readfunction \
  --timeout 5
```

Lambda 읽기 함수를 서비스 인스턴스로 등록

Lambda 읽기 함수를 앱 서비스의 다른 서비스 인스턴스로 등록하려면 다음 단계를 따릅니다.

```
aws servicediscovery register-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id read-instance \
  --attributes action=read,functionname=readfunction
```

사용자 지정 속성 `action=read` 및를 `functionname=readfunction` 사용하면 클라이언트가 목적에 따라이 함수를 검색할 수 있습니다.

클라이언트 애플리케이션 생성 및 실행

가 쓰기 함수를 검색하고 호출하는 AWS Cloud Map 데 사용하는 Python 클라이언트 애플리케이션을 생성하려면 다음 단계를 따르세요.

1. 쓰기 클라이언트 애플리케이션을 위한 Python 파일을 생성합니다.

```
cat > writeclient.py << EOF
import boto3
import json

try:
    serviceclient = boto3.client('servicediscovery')

    print("Discovering write function...")
    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='app-service',
        QueryParameters={ 'action': 'write' }
    )

    if not response.get("Instances"):
        print("Error: No instances found")
        exit(1)

    functionname = response["Instances"][0]["Attributes"].get("functionname")
    if not functionname:
        print("Error: Function name attribute not found")
        exit(1)
```

```
    print(f"Found function: {functionname}")

    lambdaclient = boto3.client('lambda')

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

    payload = resp["Payload"].read()
    print(f"Response: {payload.decode('utf-8')}")

except Exception as e:
    print(f"Error: {str(e)}")
EOF
```

이 클라이언트는 QueryParameters 옵션을 사용하여 action=write 속성이 있는 서비스 인스턴스를 찾습니다.

2. 읽기 클라이언트 애플리케이션을 위한 Python 파일을 생성합니다.

```
cat > readclient.py << EOF
import boto3
import json

try:
    serviceclient = boto3.client('servicediscovery')

    print("Discovering read function...")
    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='app-service',
        QueryParameters={ 'action': 'read' }
    )

    if not response.get("Instances"):
        print("Error: No instances found")
        exit(1)

    functionname = response["Instances"][0]["Attributes"].get("functionname")
    if not functionname:
        print("Error: Function name attribute not found")
```

```
        exit(1)

    print(f"Found function: {functionname}")

    lambdaclient = boto3.client('lambda')

    print("Invoking Lambda function...")
    resp = lambdaclient.invoke(
        FunctionName=functionname,
        InvocationType='RequestResponse'
    )

    payload = resp["Payload"].read()
    print(f"Response: {payload.decode('utf-8')}")

except Exception as e:
    print(f"Error: {str(e)}")
EOF
```

3. 쓰기 클라이언트를 실행하여 DynamoDB 테이블에 데이터를 추가합니다.

```
python3 writeclient.py
```

출력에는 HTTP 상태 코드 200으로 성공적인 응답이 표시되어야 합니다.

4. 읽기 클라이언트를 실행하여 DynamoDB 테이블에서 데이터를 검색합니다.

```
python3 readclient.py
```

출력에는 무작위로 생성된 ID와 "테스트 데이터입니다" 값을 포함하여 테이블에 기록된 데이터가 표시되어야 합니다.

리소스 정리

자습서를 마치면 추가 요금이 발생하지 않도록 리소스를 정리합니다.

1. 먼저 다음 명령을 실행하여 서비스 인스턴스의 등록을 취소합니다.

```
aws servicediscovery deregister-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id read-instance
```

```
aws servicediscovery deregister-instance \
  --service-id $APP_SERVICE_ID \
  --instance-id write-instance

aws servicediscovery deregister-instance \
  --service-id $DATA_SERVICE_ID \
  --instance-id data-instance
```

2. 다음 명령을 실행하여 서비스를 삭제합니다.

```
aws servicediscovery delete-service \
  --id $APP_SERVICE_ID

aws servicediscovery delete-service \
  --id $DATA_SERVICE_ID
```

3. 다음 명령을 실행하여 네임스페이스를 삭제합니다.

```
aws servicediscovery delete-namespace \
  --id $NAMESPACE_ID
```

4. 다음 명령을 실행하여 Lambda 함수를 삭제합니다.

```
aws lambda delete-function --function-name writefunction
aws lambda delete-function --function-name readfunction
```

5. 다음 명령을 실행하여 IAM 역할 및 정책을 삭제합니다.

```
aws iam detach-role-policy \
  --role-name cloudmap-tutorial-role \
  --policy-arn $POLICY_ARN

aws iam detach-role-policy \
  --role-name cloudmap-tutorial-role \
  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole

aws iam delete-policy \
  --policy-arn $POLICY_ARN

aws iam delete-role --role-name cloudmap-tutorial-role
```

6. 다음 명령을 실행하여 DynamoDB 테이블을 삭제합니다.

```
aws dynamodb delete-table --table-name cloudmap
```

7. 다음 명령을 실행하여 임시 파일을 정리합니다.

```
rm -f lambda-trust-policy.json cloudmap-policy.json writefunction.py
 readfunction.py writefunction.zip readfunction.zip writeclient.py readclient.py
```

AWS Cloud Map 네임스페이스

네임스페이스는 애플리케이션의 서비스를 일반적인 이름과 검색 가능성 수준으로 그룹화 AWS Cloud Map 하는 데 사용되는의 논리적 엔터티입니다. 네임스페이스를 생성할 때 다음을 지정합니다.

- 애플리케이션이 인스턴스를 검색하는 데 사용할 이름입니다.
- 등록하는 서비스 인스턴스를 검색할 AWS Cloud Map 수 있는 방법입니다. 리소스를 인터넷을 통해 공개적으로 검색해야 하는지, 특정 Virtual Private Cloud(VPC)에서 비공개로 검색해야 하는지 또는 API 직접 호출로만 검색해야 하는지 결정할 수 있습니다.

다음은 네임스페이스에 대한 일반적인 개념입니다.

- 네임스페이스는 생성된에 따라 다릅니다 AWS 리전 . 여러 리전 AWS Cloud Map 에서를 사용하려면 각 리전에 네임스페이스를 생성해야 합니다.
- VPC AWS Cloud Map 에서 DNS 쿼리에 의한 인스턴스 검색을 허용하는 네임스페이스를 생성하면 가 프라이빗 Route 53 호스팅 영역을 자동으로 생성합니다. 이 호스팅 영역은 여러 VPCs. 자세한 내용은 Amazon Route 53 API 참조의 AssociateVPCWithHostedZone을 참조하세요.

주제

애플리케이션 서비스를 그룹화하기 위한 AWS Cloud Map 네임스페이스 생성

네임스페이스를 생성하여 API 호출 또는 DNS 쿼리를 통해 애플리케이션 리소스를 검색할 수 있는 표시 이름으로 애플리케이션에 대한 서비스를 그룹화할 수 있습니다.

인스턴스 검색 옵션

다음 표에는의 다양한 인스턴스 검색 옵션 AWS Cloud Map 과 애플리케이션의 서비스 및 설정에 따라 생성할 수 있는 해당 네임스페이스 유형이 요약되어 있습니다.

네임스페이스 유형	인스턴스 검색 방법	작동 방식	추가 정보
HTTP	API 호출	애플리케이션의 리소스는 `DiscoverInstances` API만 호출하여 다른 리소스를 검색할 수 있습니다.	• DiscoverInstances • CreateHttpNamespace
프라이빗 DNS	VPC의 API 호출 및 DNS 쿼리	프라이빗 DNS 네임스페이스를 생성할 때는 해당 Amazon Route 53 프라이빗 호스팅 영역을 AWS Cloud Map 생성합니다. 애플리케이션의 리소스는 `DiscoverInstances` API를 호출하고가 AWS Cloud Map 자동으로 생성하는 프라이빗 Route 53 호스팅 영역의 네임서버를 쿼리하여 다른 리소스를 검색할 수 있습니다. 에서 생성한 호스팅 영역은 네임스페이스와 이름이 AWS Cloud Map 동일하며 *`service-name`* .*`namespace-name`* 형식의 이름을 가진 DNS 레코드를 포함합니다.	• DiscoverInstances • CreatePrivateDnsNamespace

네임스페이스 유형	인스턴스 검색 방법	작동 방식	추가 정보
		Note Route 53 해석기는 프라이빗 호스팅 영역의 레코드를 사용하여 VPC에서 시작되는 DNS 쿼리를 해석합니다. 프라이빗 호스팅 영역에 DNS 쿼리의 도메인 이름과 일치하는 레코드가 없는 경우, Route 53에서는 `NXDOMAIN`(존재하지 않는 도메인)를 사용하여 쿼리에 응답합니다.	

네임스페이스 유형	인스턴스 검색 방법	작동 방식	추가 정보
퍼블릭 DNS	API 호출 및 퍼블릭 DNS 쿼리	퍼블릭 DNS 네임스페이스를 생성할 때는 해당 Amazon Route 53 퍼블릭 호스팅 영역을 AWS Cloud Map 생성합니다. 애플리케이션의 리소스는 `DiscoverInstances` API를 호출하고가 AWS Cloud Map 자동으로 생성하는 퍼블릭 Route 53 호스팅 영역의 네임서버를 쿼리하여 다른 리소스를 검색할 수 있습니다. 퍼블릭 호스팅 영역은 네임스페이스와 이름이 동일하며 *service-name*.*namespace-name* 형식의 이름을 가진 DNS 레코드를 포함합니다. **Note** 이 경우 네임스페이스 이름은 등록한 도메인 이름이어야 합니다.	• DiscoverInstances • CreatePublicDnsNamespace

절차

다음 단계에 따라 AWS CLI AWS Management Console또는 SDK for Python을 사용하여 네임스페이스를 생성할 수 있습니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. Create namespace(네임스페이스 생성)를 선택합니다.
3. 네임스페이스 이름에 인스턴스를 검색하는 데 사용할 이름을 입력합니다.

   > **Note**
   >
   > - 퍼블릭 DNS 쿼리에 대해 구성된 네임스페이스는 최상위 도메인으로 끝나야 합니다. 예를 들어 `.com`입니다.
   > - 이름을 먼저 퓨니코드로 변환하는 경우에는 다국어 도메인 이름(IDN)을 지정할 수 있습니다. 온라인 변환기에 대한 자세한 내용은 인터넷에서 "punycode converter"를 검색하세요.
   >
   >   또한 네임스페이스를 프로그래밍 방식으로 생성하는 경우 다국어 도메인 이름을 퓨니코드로 변환할 수 있습니다. 예를 들어, Java를 사용하는 경우 java.net.IDN 라이브러리의 `toASCII` 메서드를 사용하여 유니코드 값을 퓨니코드로 변환할 수 있습니다.

4. (선택 사항) 네임스페이스 설명에 네임스페이스 페이지와 네임스페이스 정보에 표시될 네임스페이스에 대한 정보를 입력합니다. 이 정보를 사용하여 네임스페이스를 쉽게 식별할 수 있습니다.
5. 인스턴스 검색의 경우 API 호출, VPCs의 API 호출 및 DNS 쿼리, API 호출 및 퍼블릭 DNS 쿼리 중에서 선택하여 각각 HTTP, 프라이빗 DNS 또는 퍼블릭 DNS 네임스페이스를 생성할 수 있습니다. 자세한 내용은 인스턴스 검색 옵션 단원을 참조하십시오.

   선택에 따라 다음 단계를 따릅니다.

   - VPCs에서 API 호출 및 DNS 쿼리를 선택하는 경우 VPC에서 네임스페이스를 연결할 Virtual Private Cloud(VPC)를 선택합니다.
   - VPCs를 선택하거나 API 호출 및 퍼블릭 DNS 쿼리를 선택하는 경우 TTL에 대해 초 단위의 숫자 값을 지정합니다. TTL(Time To Live) 값은 DNS 해석기가 네임스페이스로 생성된

Route 53 호스팅 영역의 SOA(권한 시작) DNS 레코드에 대한 정보를 캐시하는 기간을 결정합니다. TTL에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 TTL(초)을 참조하세요.

6. (선택 사항) 태그에서 태그 추가를 선택한 다음 네임스페이스에 태그를 지정할 키와 값을 지정합니다. 네임스페이스에 추가할 태그를 하나 이상 지정할 수 있습니다. 태그를 사용하면 AWS 리소스를 보다 쉽게 관리할 수 있도록 리소스를 분류할 수 있습니다. 자세한 내용은 AWS Cloud Map 리소스에 태그 지정 단원을 참조하십시오.
7. Create namespace(네임스페이스 생성)를 선택합니다. ListOperations. 자세한 내용은 API 참조의 ListOperations를 참조하세요. AWS Cloud Map

AWS CLI

- 원하는 인스턴스 검색 유형의 명령을 사용하여 네임스페이스를 생성(### 값을 사용자 고유 값으로 대체)합니다.

  - create-http-namespace를 사용하여 HTTP 네임스페이스를 생성합니다. HTTP 네임스페이스를 사용하여 등록하는 서비스 인스턴스는 DiscoverInstances 요청을 사용하여 검색할 수 있지만 DNS를 사용하여 검색할 수 없습니다.

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

  - create-private-dns-namespace를 사용하여 지정된 Amazon VPC 내에서만 볼 수 있는 DNS에 기반한 프라이빗 네임스페이스를 만듭니다. DiscoverInstances 요청을 사용하거나 DNS를 사용하여 프라이빗 DNS 네임스페이스에 등록된 인스턴스를 검색할 수 있습니다.

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

  - create-public-dns-namespace를 사용하여 인터넷에서 볼 수 있는 DNS 기반 퍼블릭 네임스페이스를 생성합니다. DiscoverInstances 요청을 사용하거나 DNS를 사용하여 퍼블릭 DNS 네임스페이스에 등록된 인스턴스를 검색할 수 있습니다.

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

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

3. 원하는 인스턴스 검색 유형의 명령을 사용하여 네임스페이스를 생성(### 값을 사용자 고유 값으로 대체)합니다.

   - create_http_namespace()를 사용하여 HTTP 네임스페이스를 생성합니다. HTTP 네임스페이스를 사용하여 등록하는 서비스 인스턴스는 discover_instances()를 사용하여 검색할 수 있지만 DNS를 사용하여 검색할 수 없습니다.

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

   - create_private_dns_namespace()를 사용하여 지정된 Amazon VPC 내에서만 볼 수 있는 DNS에 기반한 프라이빗 네임스페이스를 만듭니다. discover_instances()를 사용하거나 DNS를 사용하여 퍼블릭 DNS 네임스페이스에 등록된 인스턴스를 검색할 수 있습니다.

     ```
     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()를 사용하여 인터넷에서 볼 수 있는 DNS 기반 퍼블릭 네임스페이스를 생성합니다. discover_instances() 또는 DNS를 사용하여 퍼블릭 DNS 네임스페이스에 등록된 인스턴스를 검색할 수 있습니다.

     ```
     response = client.create_public_dns_namespace(
         Name='name-of-namespace',
     ```

```
)
# If you want to see the response
print(response)
```

- 예시 응답 출력

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

다음 단계

네임스페이스를 생성한 후 네임스페이스에 서비스를 생성하여 애플리케이션에서 특정 목적을 집합적으로 제공하는 애플리케이션 리소스를 그룹화할 수 있습니다. 서비스는 애플리케이션 리소스를 인스턴스로 등록하기 위한 템플릿 역할을 합니다. AWS Cloud Map 서비스 생성에 대한 자세한 내용은 섹션을 참조하세요애플리케이션 구성 요소에 대한 AWS Cloud Map 서비스 생성.

AWS Cloud Map 네임스페이스 나열

네임스페이스를 생성한 후 다음 단계에 따라 생성한 네임스페이스 목록을 볼 수 있습니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택하여 네임스페이스 목록을 봅니다. 이름, 설명, 인스턴스 검색 모드, 소유자 또는 네임스페이스 ID를 기준으로 네임스페이스를 정렬할 수 있습니다. 검색 필드에 네임스페이스 이름 또는 ID를 입력하여 특정 네임스페이스를 찾고 볼 수도 있습니다.

AWS CLI

- `list-namespaces` 명령을 사용하여 네임스페이스를 나열합니다.

```
aws servicediscovery list-namespaces
```

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

3. list_namespaces()을 사용하여 네임스페이스를 나열합니다.

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

예시 응답 출력

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

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

AWS Cloud Map 네임스페이스 삭제

네임스페이스 사용을 완료한 후 삭제할 수 있습니다. 네임스페이스를 삭제하면 서비스 인스턴스를 등록 또는 검색하는 데 해당 네임스페이스를 더 이상 사용할 수 없습니다.

> **Note**
>
> DNS 네임스페이스를 삭제하면는 네임스페이스 생성 중에 생성된 해당 Amazon Route 53 호스팅 영역을 AWS Cloud Map 삭제합니다.

네임스페이스를 삭제하기 전에 모든 서비스 인스턴스의 등록을 취소한 다음 네임스페이스에서 생성된 모든 서비스를 삭제해야 합니다. 자세한 내용은 AWS Cloud Map 서비스 인스턴스 등록 취소 및 AWS Cloud Map 서비스 삭제 섹션을 참조하세요.

네임스페이스에서 생성된 인스턴스 및 삭제된 서비스를 등록 취소한 후 다음 단계에 따라 네임스페이스를 삭제합니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 삭제할 네임스페이스를 선택한 다음 삭제를 선택합니다.
4. 삭제를 다시 선택하여 서비스를 삭제할지 확인합니다.

AWS CLI

- `delete-namespace` 명령으로 네임스페이스를 삭제(*###* 값을 사용자 고유 값으로 대체)합니다. 네임스페이스에 여전히 하나 이상의 서비스가 포함되어 있으면 요청이 실패합니다.

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

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

3. delete_namespace()로 네임스페이스를 삭제(*###* 값을 사용자 고유 값으로 대체)합니다. 네임스페이스에 여전히 하나 이상의 서비스가 포함되어 있으면 요청이 실패합니다.

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

```
print(response)
```

예시 응답 출력

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

공유 AWS Cloud Map 네임스페이스

AWS Cloud Map 를 사용하면 네임스페이스 소유자가의 다른 AWS 계정 또는 조직 내에서 네임스페이스를 공유 AWS Organizations 하여 계정 간 서비스 검색 및 서비스 레지스트리를 간소화할 수 있습니다. 이를 통해 AWS 조직 내 다른 AWS 계정 또는 팀에서 관리하는 네임스페이스를 더 쉽게 사용할 수 있습니다.

AWS Cloud Map 는 AWS Resource Access Manager (AWS RAM)와 통합되어 리소스 공유를 활성화합니다. AWS RAM 는 일부 AWS Cloud Map 리소스를 다른 AWS 계정 또는를 통해 공유할 수 있는 서비스입니다 AWS Organizations. AWS RAM를 사용하면 리소스 공유를 생성하여 소유한 리소스를 공유할 수 있습니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다. 소비자에는 다음이 포함될 수 있습니다.

- 의 조직 AWS 계정 내에서 특정 AWS Organizations
- 의 조직 내 조직 단위 AWS Organizations
- 의 전체 조직 AWS Organizations

에 대한 자세한 내용은 AWS RAM 사용 설명서를 AWS RAM참조하세요.

이 항목에서는 소유한 리소스를 공유하는 방법과 공유 리소스를 사용하는 방법을 설명합니다.

내용

- 공유 AWS Cloud Map 네임스페이스 식별
- 네임스페이스를 공유할 수 있는 권한 부여
- 공유 네임스페이스에 대한 책임 및 권한
- 결제 및 측정
- 할당량

네임스페이스 공유 시 고려 사항

- 네임스페이스를 공유하려면에서 해당 네임스페이스를 소유해야 합니다 AWS 계정. 즉, 계정에서 리소스를 할당하거나 프로비저닝해야 합니다. 공유된 네임스페이스는 공유할 수 없습니다.
- 에서 조직 또는 조직 단위와 네임스페이스를 공유하려면 와의 공유를 활성화 AWS Organizations해야 합니다 AWS Organizations. 자세한 내용은 AWS RAM 사용 설명서의 AWS Organizations과(와) 공유 활성화를 참조하세요.
- 공유 프라이빗 DNS 네임스페이스에서 DNS 쿼리를 사용하는 서비스 검색의 경우 네임스페이스 소유자는 네임스페이스 및 소비자의 VPC`create-vpc-association-authorization`와 연결된 프라이빗 호스팅 영역의 ID로를 호출해야 합니다.

  ```
  aws route53 create-vpc-association-authorization --hosted-zone-id Z1234567890ABC --
  vpc VPCRegion=us-east-1,VPCId=vpc-12345678
  ```

  네임스페이스 소비자는 프라이빗 호스팅 영역의 ID를 사용하여 `associate-vpc-with-hosted-zone`을 직접 호출해야 합니다.

  ```
  aws route53 associate-vpc-with-hosted-zone --hosted-zone-id Z1234567890ABC --vpc
   VPCRegion=us-east-1,VPCId=vpc-12345678
  ```

  자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon VPC와 다른 로 생성한 프라이빗 호스팅 영역 연결을 참조하세요 AWS 계정.
- 공유 DNS 네임스페이스와 연결된 서비스의 up-to-date 네트워크 위치를 검색한 후 다른 VPC에 있는 경우 서비스와 통신하도록 VPCs. VPC 피어링 연결을 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 Amazon Virtual Private Cloud VPC 피어링 가이드의 VPC 피어링 연결 생성을 참조하세요.
- `ListOperations`를 사용하여 다른 계정에서 수행하는 공유 네임스페이스에 대한 작업을 나열할 수 없습니다.
- 공유 네임스페이스에는 태그 지정이 지원되지 않습니다.

AWS Cloud Map 네임스페이스 공유

소유한 AWS Cloud Map 네임스페이스를 다른 AWS 계정 (소비자)와 공유하면 이러한 계정이 임시 자격 증명 없이 네임스페이스에서 서비스의 up-to-date 네트워크 위치를 검색할 수 있습니다.

네임스페이스를 공유하려면 리소스 공유에 추가해야 합니다. 리소스 공유는 AWS 계정전반에서 리소스를 공유할 수 있게 해주는 AWS RAM 리소스입니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다. 새 리소스 공유에 네임스페이스를 추가하려면 먼저 AWS RAM 콘솔을 사용하여 리소스 공유를 생성해야 합니다.

의 조직에 속 AWS Organizations 해 있고 조직 내 공유가 활성화된 경우 조직의 소비자에게 공유 네임스페이스에 대한 액세스 권한이 자동으로 부여됩니다. 그렇지 않으면 소비자는 리소스 공유에 가입하라는 초대를 받고 초대를 수락한 후 공유 네임스페이스에 대한 액세스 권한이 부여됩니다.

AWS RAM 콘솔 또는를 사용하여 소유한 네임스페이스를 공유할 수 있습니다 AWS CLI.

AWS RAM console

AWS RAM 콘솔을 사용하여 소유한 네임스페이스를 공유하려면

AWS RAM 사용 설명서의 Creating a resource share in AWS RAM을 참조하세요.

AWS CLI

를 사용하여 소유한 네임스페이스를 공유하려면 AWS CLI

AWS RAM create-resource-share 명령을 사용합니다.

AWS Cloud Map 네임스페이스 공유 중지

네임스페이스가 더 이상 공유되지 않으면 소비자가 더 이상 네임스페이스와 해당 네임스페이스와 연결된 서비스 및 인스턴스에 액세스할 수 없습니다 AWS 계정. 여기에는 소비자가 네임스페이스에 액세스할 때 네임스페이스에서 생성된 리소스가 포함됩니다.

소유한 네임스페이스의 공유를 중지하려면 리소스 공유에서 해당 네임스페이스를 제거해야 합니다. AWS RAM 콘솔 또는를 사용하여이 작업을 수행할 수 있습니다 AWS CLI.

AWS RAM console

AWS RAM 콘솔을 사용하여 소유한 네임스페이스 공유를 중지하려면

AWS RAM 사용 설명서에서 리소스 공유 업데이트를 참조하세요.

AWS CLI

를 사용하여 소유한 네임스페이스 공유를 중지하려면 AWS CLI

disassociate-resource-share 명령을 사용합니다.

공유 AWS Cloud Map 네임스페이스 식별

소유자와 소비자는 AWS Cloud Map 콘솔 및를 사용하여 공유 네임스페이스를 식별할 수 있습니다 AWS CLI. 네임스페이스 소유자는 `ResourceOwner` 속성을 사용하여 식별할 수 있습니다. `CreatedByAccount` 속성을 사용하여 서비스를 생성하거나 공유 네임스페이스에 인스턴스를 등록 AWS 계정 하는를 식별할 수 있습니다.

AWS Cloud Map console

AWS Cloud Map 콘솔을 사용하여 공유 네임스페이스를 식별하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 네임스페이스 페이지의 리소스 소유자에서 네임스페이스를 소유 AWS 계정 한의 ID를 찾을 수 있습니다.
3. 식별하려는 네임스페이스의 도메인 이름을 선택합니다.
4. 네임스페이스: ***namespace-name*** 페이지의 네임스페이스 정보 섹션의 리소스 소유자에서 네임스페이스를 소유 AWS 계정 한의 ID를 찾을 수 있습니다.

AWS CLI

를 사용하여 공유 네임스페이스를 식별하려면 list-namespaces 명령을 AWS CLI사용합니다. 이 명령은 사용자가 소유한 네임스페이스와 사용자와 공유된 네임스페이스를 반환합니다. `ResourceOwner` 필드에는 네임스페이스 소유자의 AWS 계정 ID가 표시됩니다.

계정에서 다음 `list-namespaces` 호출을 수행합니다111122223333.

```
aws servicediscovery list-namespaces
```

출력:

```
{
```

```
    "Namespaces": [
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:111122223333:namespace/ns-
abcdef01234567890",
            "CreateDate": 1585354387.357,
            "Id": "ns-abcdef01234567890",
            "Name": "local",
            "Properties": {
                "DnsProperties": {
                    "HostedZoneId": "Z06752353VBUDTC32S84S"
                },
                "HttpProperties": {
                    "HttpName": "local"
                 }
            },
            "Type": "DNS_PRIVATE",
            "ServiceCount": 2,
           "ResourceOwner": "111122223333"
        },
        {
            "Arn": "arn:aws:servicediscovery:us-west-2:444455556666:namespace/
ns-021345abcdef6789",
            "CreateDate": 1586468974.698,
            "Description": "Shared second namespace",
            "Id": "ns-021345abcdef6789",
            "Name": "My-second-namespace",
            "Properties": {
                "DnsProperties": {},
                "HttpProperties": {
                    "HttpName": "Shared-second-namespace"
                }
            },
            "Type": "HTTP",
            "ServiceCount": 0,
            "ResourceOwner": "444455556666"
        }
    ]
}
```

이 시나리오에서는 네임스페이스ns-abcdef01234567890가에서 생성 및 소유111122223333되고 네임스페이스ns-021345abcdef6789가에서 생성 및 소유됩니다444455556666. 네임스페이스ns-021345abcdef6789는 계정 111122223333에서 계정와 공유됩니다444455556666.

네임스페이스를 공유할 수 있는 권한 부여

IAM 보안 주체가 네임스페이스를 공유하려면 최소 권한 집합이 필요합니다. `AWSCloudMapFullAccess` 및 `AWSResourceAccessManagerFullAccess` 관리형 정책을 사용하여 IAM 보안 주체가 공유 네임스페이스를 공유하고 사용하는 데 필요한 권한을 갖도록 하는 것이 좋습니다.

사용자 지정 IAM 정책을 사용하는 경우 네임스페이스를 공유하려면 `servicediscovery:PutResourcePolicyservicediscovery:GetResourcePolicy`, 및 `servicediscovery:DeleteResourcePolicy` 작업이 필요합니다. 이는 권한 전용 IAM 작업입니다. IAM 보안 주체에게 이러한 권한이 부여되지 않은 경우를 사용하여 네임스페이스를 공유하려고 할 때 오류가 발생합니다 AWS RAM.

에서 IAM AWS RAM 을 사용하는 방법에 대한 자세한 내용은 AWS RAM 사용 설명서의 IAM을 AWS RAM 사용하는 방법을 참조하세요.

공유 네임스페이스에 대한 책임 및 권한

네임스페이스 소유자와 소비자는 공유 네임스페이스에서 서로 다른 작업을 수행할 수 있습니다.

소유자에 대한 권한

네임스페이스 소유자는 공유 네임스페이스에서 다음 작업을 수행할 수 있습니다.

- 소비자 계정에서 생성한 서비스 및 이러한 서비스에 등록된 인스턴스를 포함하여 네임스페이스와 연결된 서비스에 액세스합니다.
- 소비자 계정 및 이러한 서비스에 등록된 인스턴스가 생성한 서비스에 대한 액세스를 포함하여 네임스페이스에 대한 액세스를 취소합니다.
- 소비자 또는 네임스페이스 소유자가 공유 네임스페이스에서 생성한 서비스에서 인스턴스를 등록 및 등록 취소하도록 다른 계정에 대한 권한을 구성합니다.
- 소비자 계정에서 생성한 서비스 및 등록된 인스턴스를 포함하여 서비스를 삭제하고 인스턴스 등록을 취소합니다.
- 공유 네임스페이스를 업데이트하거나 삭제합니다.

소비자에 대한 권한

네임스페이스 소비자는 공유 네임스페이스에서 다음 작업을 수행할 수 있습니다.

- 네임스페이스에서 서비스를 생성하고 삭제합니다.
- 네임스페이스에서 생성된 서비스에서 인스턴스를 등록 및 등록 취소합니다.
- 네임스페이스에서 생성된 서비스에 등록된 인스턴스를 검색합니다.

소비자는 공유 네임스페이스를 업데이트하거나 삭제할 수 없습니다. 공유 네임스페이스에 대한 액세스 권한을 잃으면 소비자 계정도 네임스페이스에서 생성한 서비스에 대한 액세스 권한을 잃게 됩니다.

결제 및 측정

공유 네임스페이스에 등록하는 모든 인스턴스와 이러한 인스턴스를 등록할 때 생성되는 모든 Route 53 상태 확인에 대해 소유자에게 요금이 청구됩니다. 소비자는 네임스페이스에 등록한 모든 인스턴스와 이러한 인스턴스를 등록할 때 생성된 모든 Route 53 상태 확인에 대해 요금이 청구됩니다. 공유 네임스페이스가 DNS 네임스페이스인 경우 네임스페이스에서 서비스가 생성될 때 생성된 Route 53 DNS 레코드에 대해 네임스페이스 소유자에게 요금이 청구됩니다. 소유자는 모든 `DiscoverInstances` 및 `DiscoverInstancesRevision` 호출에 대해 요금이 청구됩니다. 소비자는 모든 `DiscoverInstances` 및 `DiscoverInstancesRevision` 호출에 대해 요금이 청구됩니다.

할당량

공유 네임스페이스는 리전 할당량당 네임스페이스 소유자의 네임스페이스에만 포함됩니다. 공유 네임스페이스에 소비자가 등록한 인스턴스는 네임스페이스 할당량당 소유자의 인스턴스에 포함됩니다. 소비자가 공유 네임스페이스에서 서비스를 생성하는 경우 서비스에 등록된 모든 인스턴스는 서비스 할당량당 소비자의 인스턴스에 포함됩니다. 소유자가 공유 네임스페이스에서 서비스를 생성하는 경우 서비스에 등록된 모든 인스턴스는 서비스 할당량당 소유자의 인스턴스에 포함됩니다.

AWS Cloud Map 서비스

AWS Cloud Map 서비스는 서비스에 대한 서비스 이름 및 DNS 구성으로 구성된 서비스 인스턴스를 등록하기 위한 템플릿입니다. 상태 확인을 설정하여 서비스에서 인스턴스의 상태를 확인하고 비정상 리소스를 필터링할 수도 있습니다. 서비스는 애플리케이션의 구성 요소를 나타낼 수 있습니다. 예를 들어 애플리케이션에서 결제를 처리하는 리소스와 사용자를 관리하는 리소스에 대한 서비스를 생성할 수 있습니다.

서비스를 사용하면 리소스에 연결하는 데 사용할 수 있는 하나 이상의 엔드포인트를 다시 가져와서 애플리케이션의 리소스를 찾을 수 있습니다. 리소스의 위치는 네임스페이스를 AWS Cloud Map `DiscoverInstances` 구성한 방법에 따라 DNS 쿼리 또는 API 작업을 사용하여 수행됩니다. AWS Cloud Map 콘솔을 사용하여 서비스 수준에서 인스턴스 검색의 범위를 지정할 수 있습니다.

`UpdateServiceAttributes` API를 사용하여 서비스 수준에서 사용자 지정 메타데이터를 속성으로 지정할 수도 있습니다. 인스턴스 간에 속성이 중복되지 않도록 서비스 속성을 설정하고 인스턴스 속성을 변경할 필요 없이 이러한 속성을 수정할 수 있습니다. 서비스 수준에서 속성으로 지정할 수 있는 정보에는 다음이 포함되지만 이에 국한되지는 않습니다.

- 점진적 배포 중 트래픽 이동을 위한 엔드포인트 가중치입니다.
- API 제한 시간 및 권장 재시도 정책과 같은 서비스 기본 설정입니다.

자세한 내용은 AWS Cloud Map API 참조의 UpdateServiceAttributes를 참조하세요.

다음 주제에서는 서비스의 상태 확인 및 DNS 구성에 대해 설명하고 서비스 생성, 나열, 업데이트 및 삭제 지침을 포함합니다.

주제

- AWS Cloud Map 서비스 상태 확인 구성
- AWS Cloud Map 서비스 DNS 구성
- 애플리케이션 구성 요소에 대한 AWS Cloud Map 서비스 생성
- AWS Cloud Map 서비스 업데이트
- 네임스페이스에 AWS Cloud Map 서비스 나열
- AWS Cloud Map 서비스 삭제

AWS Cloud Map 서비스 상태 확인 구성

상태 확인은 서비스 인스턴스가 정상인지 여부를 확인하는 데 도움이 됩니다. 서비스 생성 중에 상태 확인을 구성하지 않으면 인스턴스의 상태에 관계없이 트래픽이 서비스 인스턴스로 라우팅됩니다. 상태 확인을 구성하면가 기본적으로 정상 리소스를 AWS Cloud Map 반환합니다. `DiscoverInstances` API의 `HealthStatus` 파라미터를 사용하여 상태별로 리소스를 필터링하고 비정상 리소스 목록을 가져올 수 있습니다. `GetInstancesHealthStatus` API를 사용하여 특정 서비스 인스턴스의 상태를 검색할 수도 있습니다.

AWS Cloud Map 서비스를 생성할 때 Route 53 상태 확인 또는 사용자 지정 타사 상태 확인을 구성할 수 있습니다.

Route 53 상태 확인

Amazon Route 53 상태 확인에 대한 설정을 지정하는 경우는 인스턴스를 등록할 때마다 Route 53 상태 확인을 AWS Cloud Map 생성하고 인스턴스 등록을 취소할 때 상태 확인을 삭제합니다.

퍼블릭 DNS 네임스페이스의 경우 인스턴스를 등록할 때가 AWS Cloud Map 생성하는 Route 53 레코드와 상태 확인을 AWS Cloud Map 연결합니다. 서비스의 DNS 구성에서 `A` 및 `AAAA` 레코드 유형을 모두 지정하는 경우는 IPv4 주소를 사용하여 리소스의 상태를 확인하는 상태 확인을 AWS Cloud Map 생성합니다. IPv4 주소로 지정된 엔드포인트가 비정상인 경우 Route 53는 `A` 및 `AAAA` 레코드를 모두 비정상으로 간주합니다. 서비스의 DNS 구성에서 `CNAME` 레코드 유형을 지정하는 경우 Route 53 상태 확인을 구성할 수 없습니다.

API 호출을 사용하여 인스턴스를 검색하는 네임스페이스에 대해 AWS Cloud Map 에서는 Route 53 상태 확인을 생성합니다. 그러나 상태 확인을 연결할 AWS Cloud Map 에 대한 DNS 레코드는 없습니다. 상태 확인이 정상인지 확인하기 위해 Route 53 콘솔 또는 Amazon CloudWatch를 사용하여 모니터링을 구성할 수 있습니다. Route 53 콘솔 사용에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 상태 확인 실패 시 알림 메시지를 받음을 참조하세요. CloudWatch 사용에 대한 자세한 내용은 Amazon CloudWatch API 참조의 PutMetricAlarm을 참조하세요.

> **Note**
>
> - 프라이빗 DNS 네임스페이스에서 생성된 서비스에 대해서는 Amazon Route 53 상태 확인을 구성할 수 없습니다.
> - 각 상태 확인의 Route 53 상태 확인 프로그램은 30초마다 엔드포인트에 상태 확인 요청을 AWS 리전 보냅니다. 평균적으로 엔드포인트에서는 약 2초 간격으로 상태 확인 요청을 수신

합니다. 그러나 상태 검사기는 서로 조정하지 않습니다. 따라서 1초에 여러 건의 요청이 있고 이후 몇 초간 상태 확인이 아예 없는 경우가 종종 있습니다. 상태 확인 리전 목록은 리전을 참조하세요.

Route 53 상태 확인 비용에 대한 자세한 내용은 Route 53 요금을 참조하세요.

사용자 지정 상태 확인

인스턴스를 등록할 때 사용자 지정 상태 확인을 AWS Cloud Map 사용하도록를 구성하는 경우 타사 상태 확인 프로그램을 사용하여 리소스의 상태를 평가해야 합니다. 사용자 지정 상태 확인은 다음과 같은 경우에 유용합니다.

- 인터넷을 통해 리소스를 사용할 수 없어 Route 53 상태 확인을 사용할 수 없는 경우. 예를 들어, Amazon VPC에 있는 인스턴스가 있다고 가정해 보겠습니다. 이 인스턴스에 대해 사용자 지정 상태 확인을 사용할 수 있습니다. 하지만 상태 확인이 작동하려면 상태 확인 검사기가 인스턴스와 동일한 VPC에 있어야 합니다.
- 리소스 위치와 상관없이 타사 상태 확인 프로그램을 사용하려는 경우

사용자 지정 상태 확인을 사용하는 경우 AWS Cloud Map 는 지정된 리소스의 상태를 직접 확인하지 않습니다. 대신 타사 상태 확인 프로그램은 리소스의 상태를 확인하고 애플리케이션에 상태를 반환합니다. 그러면 애플리케이션에서이 상태를 전달하는 `UpdateInstanceCustomHealthStatus` 요청을 제출해야 합니다 AWS Cloud Map. 초기 릴레이 상태가 이고 UNHEALTHY30초 `UpdateInstanceCustomHealthStatus` 이내에 상태를 릴레이하는 다른이 없는 경우 HEALTHY리소스가 비정상으로 확인됩니다.는 해당 리소스로 트래픽 라우팅을 AWS Cloud Map 중지합니다.

AWS Cloud Map 서비스 DNS 구성

DNS 쿼리에 의한 인스턴스 검색을 지원하는 네임스페이스에서 서비스를 생성하면가 Route 53 DNS 레코드를 AWS Cloud Map 생성합니다. 가 AWS Cloud Map 생성하는 모든 Route 53 DNS 레코드에 적용되는 Route 53 라우팅 정책 및 DNS 레코드 유형을 지정해야 합니다.

라우팅 정책

라우팅 정책은 Route 53가 서비스 인스턴스 검색에 사용되는 DNS 쿼리에 응답하는 방법을 결정합니다. 지원되는 라우팅 정책 및 관련 방법은 다음과 AWS Cloud Map 같습니다.

가중치 기반 라우팅

Route 53은 동일한 서비스를 사용하여 등록한 인스턴스 중에서 무작위로 선택한 AWS Cloud Map 하나의 AWS Cloud Map 서비스 인스턴스에서 해당 값을 반환합니다. 모든 레코드가 동일한 가중치를 갖기 때문에 인스턴스로 라우팅되는 트래픽을 늘리거나 줄일 수 없습니다.

예를 들어, 서비스에 A 레코드 하나와 상태 확인에 대한 구성이 포함되어 있는데, 이 서비스를 사용하여 인스턴스 10개를 등록한다고 가정해 보겠습니다. Route 53은 정상 인스턴스 중 무작위로 선택한 하나의 인스턴스에 대한 IP 주소를 사용하여 DNS 쿼리에 응답합니다. 정상 인스턴스가 없는 경우 Route 53은 마치 모든 인스턴스가 정상인 것처럼 DNS 쿼리에 응답합니다.

이 서비스에 대해 상태 확인을 정의하지 않은 경우 Route 53에서는 모든 인스턴스가 정상이라고 가정하고 임의로 선택한 인스턴스 하나에 대해 해당 값을 반환합니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 가중치 기반 라우팅을 참조하세요.

다중값 응답 라우팅

이 서비스에 대해 상태 확인을 정의했고 상태 확인 결과가 정상인 경우 Route 53에서는 최대 8개 인스턴스에 대해 해당 값을 반환합니다.

예를 들어 서비스에 하나의 A 레코드와 상태 확인에 대한 구성이 포함되어 있다고 가정합니다. 서비스를 사용하여 10개의 인스턴스를 등록합니다. Route 53에서는 최대 8개의 정상 인스턴스에 대해서만 IP 주소를 사용하여 DNS 쿼리에 응답합니다. 정상 인스턴스가 8개 미만인 경우 Route 53에서는 전체 정상 인스턴스의 IP 주소로 모든 DNS 쿼리에 응답합니다.

이 서비스에 대해 상태 확인을 정의하지 않은 경우 Route 53에서는 모든 인스턴스가 정상이라고 가정하고 최대 8개 인스턴스에 대한 값을 반환합니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 다중 응답 라우팅을 참조하세요.

레코드 유형

Route 53 DNS 레코드 유형은 서비스 인스턴스 검색에 사용되는 DNS 쿼리에 대한 응답으로 Route 53가 반환하는 값의 유형을 결정합니다. 지정할 수 있는 다양한 DNS 레코드 유형과 쿼리에 대한 응답으로 Route 53에서 반환되는 관련 값은 다음과 같습니다.

A

이 유형을 지정하면 Route 53는 리소스의 IP 주소를 192.0.2.44와 같은 IPv4 형식으로 반환합니다.

AAAA

이 유형을 지정하면 Route 53는 리소스의 IP 주소를 2001:0db8:85a3:0000:0000:abcd:0001:2345와 같은 IPv6 형식으로 반환합니다.

CNAME

이 유형을 지정하면 Route 53는 리소스의 도메인 이름(예: www.example.com)을 반환합니다.

> **Note**
>
> - CNAME DNS 레코드를 구성하려면 가중치 기반 라우팅 라우팅 정책을 지정해야 합니다.
> - CNAME DNS 레코드를 구성할 때는 Route 53 상태 확인을 구성할 수 없습니다.

SRV

이 유형을 지정하면 Route 53가 SRV 레코드 값을 반환합니다. SRV 레코드의 값은 다음 값을 사용합니다.

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

다음을 고려하세요.

- `priority` 및 `weight` 값은 둘 다 1로 설정되어 있고 변경할 수 없습니다.
- 의 경우 인스턴스를 등록할 때 포트(AWS_INSTANCE_PORT)에 지정한 값을 `port` AWS Cloud Map 사용합니다.
- `service-hostname`의 값은 다음 값의 연결입니다.
  - 인스턴스를 등록할 때 서비스 인스턴스 ID(InstanceID)에 지정하는 값입니다.
  - 서비스의 이름
  - 네임스페이스의 이름

  예를 들어 인스턴스를 등록할 때 테스트를 인스턴스 ID로 지정한다고 가정해 보겠습니다. 서비스 이름은 백엔드이고 네임스페이스의 이름은 example.com입니다. AWS Cloud Map 에서는 SRV 레코드의 `service-hostname` 속성에 다음 값을 할당합니다.

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

> **Note**
>
> 인스턴스를 등록할 때 IPv4 주소, IPv6 주소 또는 둘 다 값을 지정하면는 SRV 레코드`service-hostname`의 값과 이름이 동일한 A 및/또는 AAAA 레코드를 AWS Cloud Map 자동으로 생성합니다.

레코드 유형은 다음 조합으로 지정할 수 있습니다.

- A
- AAAA
- A 및 AAAA
- CNAME
- SRV

A 및 AAAA 레코드 유형을 지정한 경우, 인스턴스를 등록할 때 IPv4 IP 주소, IPv6 IP 주소 또는 둘 다를 지정할 수 있습니다.

애플리케이션 구성 요소에 대한 AWS Cloud Map 서비스 생성

네임스페이스를 생성한 후 특정 목적에 맞는 애플리케이션의 다양한 구성 요소를 나타내는 서비스를 생성할 수 있습니다. 예를 들어 결제를 처리하는 애플리케이션의 리소스에 대한 서비스를 생성할 수 있습니다.

> **Note**
>
> 대/소문자만 다른 이름(예: EXAMPLE 및 예제)으로 DNS 쿼리에서 액세스할 수 있는 여러 서비스를 생성할 수 없습니다. 이렇게 하면 이러한 서비스의 DNS 이름이 동일합니다. API 호출로만 액세스할 수 있는 네임스페이스를 사용하는 경우, 철자는 같지만 대소문자는 다른 이름을 가진 서비스를 생성할 수 있습니다.

다음 단계에 따라 AWS Management Console AWS CLI및 SDK for Python을 사용하여 서비스를 생성합니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 네임스페이스 페이지에서 서비스를 추가하려는 네임스페이스를 선택합니다.
4. 네임스페이스: ***namespace-name*** 페이지에서 서비스 생성을 선택합니다.
5. 서비스 이름에이 서비스를 사용할 때 등록하는 인스턴스를 설명하는 이름을 입력합니다. 값은 API 호출 또는 DNS 쿼리에서 AWS Cloud Map 서비스 인스턴스를 검색하는 데 사용됩니다.

   > **Note**
   >
   > 인스턴스를 등록할 때 SRV 레코드를 AWS Cloud Map 생성하고 특정 SRV 형식(예: HAProxy)이 필요한 시스템을 사용하는 경우 서비스 이름에 다음을 지정합니다.
   >
   > - 이름은 밑줄(_)로 시작합니다(예: _exampleservice).
   > - 이름을 *._protocol*로 끝냅니다(예: ._tcp).
   >
   > 인스턴스를 등록할 때 AWS Cloud Map 는 SRV 레코드를 생성하고 서비스 이름과 네임스페이스 이름을 연결하여 이름을 할당합니다. 예를 들면 다음과 같습니다. _exampleservice._tcp.example.com

6. (선택 사항) 서비스 설명에 서비스에 대한 설명을 입력합니다. 여기에 입력하는 설명은 서비스 페이지와 각 서비스의 세부 정보 페이지에 표시됩니다.
7. 네임스페이스가 DNS 쿼리를 지원하는 경우 서비스 검색 구성에서 서비스 수준에서 검색 가능성을 구성할 수 있습니다. API 호출과 DNS 쿼리를 모두 허용하거나이 서비스의 인스턴스 검색을 위한 API 호출만 허용하도록 선택합니다.

   > **Note**
   >
   > API 호출을 선택하면 AWS Cloud Map 는 인스턴스를 등록할 때 SRV 레코드를 생성하지 않습니다.

   API 및 DNS를 선택하는 경우 다음 단계에 따라 DNS 레코드를 구성합니다. DNS 레코드를 추가하거나 제거할 수 있습니다.

1. 라우팅 정책에서 인스턴스를 등록할 때가 생성하는 AWS Cloud Map DNS 레코드에 대한 Amazon Route 53 라우팅 정책을 선택합니다. 가중치 기반 라우팅과 다중값 응답 라우팅 중에서 선택할 수 있습니다. 자세한 내용은 라우팅 정책 단원을 참조하십시오.

   > **Note**
   >
   > 인스턴스를 등록할 때 콘솔을 사용하여 Route 53 별칭 레코드를 생성 AWS Cloud Map 하도록를 구성할 수 없습니다. 프로그래밍 방식으로 인스턴스 AWS Cloud Map 를 등록할 때 Elastic Load Balancing 로드 밸런서에 대한 별칭 레코드를 생성하려면 라우팅 정책에 대해 가중치 기반 라우팅을 선택합니다.

2. 레코드 유형에서 DNS 쿼리에 대한 응답으로 Route 53가 반환하는 대상을 결정하는 DNS 레코드 유형을 선택합니다 AWS Cloud Map. 자세한 내용은 레코드 유형 단원을 참조하십시오.

3. TTL의 경우 숫자 값을 지정하여 서비스 수준에서 TTL(Time to Live) 값을 초 단위로 정의합니다. TTL 값은 업데이트된 설정을 얻기 위해 DNS 해석기가 다른 DNS 쿼리를 Amazon Route 53에 전달하기 전에 이 레코드에 대한 정보를 캐싱하는 기간을 결정합니다.

8. 상태 확인 구성의 상태 확인 옵션에서 서비스 인스턴스에 적용할 수 있는 상태 확인 유형을 선택합니다. 상태 확인을 구성하지 않도록 선택하거나 인스턴스에 대한 Route 53 상태 확인 또는 외부 상태 확인 중에서 선택할 수 있습니다. 자세한 내용은 AWS Cloud Map 서비스 상태 확인 구성 단원을 참조하십시오.

   > **Note**
   >
   > Route 53 상태 확인은 퍼블릭 DNS 네임스페이스의 서비스에 대해서만 구성할 수 있습니다.

   Route 53 상태 확인을 선택하는 경우 다음 정보를 제공합니다.

   1. 실패 임계값에 서비스 인스턴스가 상태를 변경하기 위해 통과하거나 실패해야 하는 연속 Route 53 상태 확인 수를 정의하는 1~10 사이의 숫자를 입력합니다.
   2. 상태 확인 프로토콜에서 Route 53가 서비스 인스턴스의 상태를 확인하는 데 사용할 방법을 선택합니다.
   3. HTTP 또는 HTTPS 상태 확인 프로토콜을 선택하는 경우 상태 확인 경로에 상태 확인을 수행할 때 Amazon Route 53에서 요청할 경로를 제공합니다. 경로는 `/docs/route53-`

`health-check.html` 파일과 같은 모든 값이 될 수 있습니다. 리소스가 정상일 때 반환되는 값은 2xx 또는 3xx 형식의 HTTP 상태 코드입니다. 쿼리 문자열 파라미터를 포함해도 됩니다(예: `/welcome.html?language=jp&login=y`). AWS Cloud Map 콘솔에서는 앞에 슬래시(/) 문자를 자동으로 덧붙입니다.

Route 53 상태 확인에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53가 상태 확인이 정상인지 확인하는 방법을 참조하세요.

9. (선택 사항) 태그에서 태그 추가를 선택한 다음 네임스페이스에 태그를 지정할 키와 값을 지정합니다. 네임스페이스에 추가할 태그를 하나 이상 지정할 수 있습니다. 태그를 사용하면 AWS 리소스를 보다 쉽게 관리할 수 있도록 리소스를 분류할 수 있습니다. 자세한 내용은 AWS Cloud Map 리소스에 태그 지정 단원을 참조하십시오.
10. 서비스 생성을 선택합니다.

AWS CLI

- `create-service` 명령을 사용하여 서비스를 생성합니다. `###` 값을 사용자 값으로 바꿉니다.

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

출력:

```
{
        "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)

아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.

1. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

2. 를 사용하여 서비스를 생성합니다create_service(). ### 값을 사용자 값으로 바꿉니다. 자세한 내용은 create_service를 참조하세요.

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

예시 응답 출력

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

다음 단계

서비스를 생성한 후 애플리케이션이 리소스를 찾는 방법에 대한 정보가 포함된 애플리케이션 리소스를 서비스 인스턴스로 등록할 수 있습니다. AWS Cloud Map 서비스 인스턴스 등록에 대한 자세한 내용은 섹션을 참조하세요리소스를 AWS Cloud Map 서비스 인스턴스로 등록.

서비스를 생성한 후 엔드포인트 가중치, API 제한 시간 및 재시도 정책과 같은 사용자 지정 메타데이터를 서비스 속성으로 지정할 수도 있습니다. 자세한 내용은 API 참조`UpdateServiceAttributes`의 ServiceAttributes 및 섹션을 참조하세요. AWS Cloud Map

AWS Cloud Map 서비스 업데이트

서비스의 구성에 따라 DNS 해석기의 태그, Route 53 상태 확인 실패 임계값 및 TTL(Time To Live)을 업데이트할 수 있습니다. 서비스를 업데이트하려면 다음 절차를 수행합니다.

> **Note**
>
> HTTP 네임스페이스와 연결된 서비스의 설정은 업데이트할 수 없습니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 네임스페이스 페이지에서 서비스가 생성되는 네임스페이스를 선택합니다.
4. 네임스페이스: ***namespace-name*** 페이지에서 편집하려는 서비스를 선택하고 세부 정보 보기를 선택합니다.
5. 서비스: ***service-name*** 페이지에서 편집을 선택합니다.

   > **Note**
   >
   > 편집 버튼 워크플로를 사용하여 인스턴스 검색에 대한 API 호출만 허용하는 서비스의 값을 편집할 수 없습니다. 그러나 서비스: ***### ##*** 페이지에서 태그를 추가하거나 제거할 수 있습니다.

6. 서비스 편집 페이지의 서비스 설명에서 서비스에 대해 이전에 설정한 설명을 업데이트하거나 새 설명을 추가할 수 있습니다. DNS 해석기의 태그를 추가하고 TTL을 업데이트할 수도 있습니다.
7. DNS 구성에서 TTL의 경우 업데이트된 설정을 가져오기 위해 DNS 해석기가 다른 DNS 쿼리를 Amazon Route 53에 전달하기 전에이 레코드에 대한 정보를 캐시하는 기간을 결정하는 업데이트된 기간을 초 단위로 지정할 수 있습니다.
8. Route 53 상태 확인을 설정한 경우 실패 임계값에 대해 서비스 인스턴스가 상태를 변경하기 위해 통과하거나 실패해야 하는 연속 Route 53 상태 확인 수를 정의하는 1~10 사이의 새 숫자를 지정할 수 있습니다.
9. 서비스 업데이트를 선택합니다.

AWS CLI

- update-service 명령을 사용하여 서비스를 업데이트(### 값을 사용자 고유 값으로 대체)합니다.

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

출력:

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

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

3. update_service()로 서비스를 업데이트(### 값을 사용자 고유 값으로 대체)합니다.

```
response = client.update_service(
    Id='srv-xxxxxxxxxxx',
    Service={
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 300,
                    'Type': 'A',
                },
            ],
        },
        'Description': "new description",
    }
)
```

예시 응답 출력

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

네임스페이스에 AWS Cloud Map 서비스 나열

네임스페이스에서 생성한 서비스 목록을 보려면 다음 절차를 수행합니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 나열하려는 서비스가 포함된 네임스페이스의 도메인 이름을 선택합니다. 서비스에서 모든 서비스 목록을 보고 검색 필드에 서비스 이름 또는 ID를 입력하여 특정 서비스를 찾을 수 있습니다. 생성자 필드를 사용하여 서비스를 AWS 계정 생성한를 식별하고 리소스 소유자 필드를 사용하여 서비스를 소유한 계정을 식별할 수 있습니다.

> **Note**
>
> 네임스페이스가 공유 네임스페이스인 경우 리소스 소유자 아래의 AWS 계정 ID는 네임스페이스를 생성하고 공유한 계정입니다. 네임스페이스 소비자가 서비스를 생성한 경우 에서 생성한 계정 ID는 리소스 소유자의 ID와 다를 수 있습니다. 계정 IDs는 계정 ID와 동일하지 않을 수 있습니다. 공유 네임스페이스에 대한 자세한 내용은 섹션을 참조하세요공유 AWS Cloud Map 네임스페이스.

AWS CLI

- `list-services` 명령을 사용하여 서비스를 나열합니다. 다음 명령은 네임스페이스 ID를 필터로 사용하여 네임스페이스의 모든 서비스를 나열합니다. *###* 값을 자신의 값으로 바꿉니다.

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

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 `servicediscovery`를 사용하세요.

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

3. `list_services()`를 사용하여 서비스를 나열하세요.

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

예시 응답 출력

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

AWS Cloud Map 서비스 삭제

서비스를 삭제하기 전에 해당 서비스를 사용해 등록한 모든 서비스 인스턴스를 등록 취소해야 합니다. 자세한 내용은 AWS Cloud Map 서비스 인스턴스 등록 취소 단원을 참조하십시오.

서비스를 사용하여 등록된 모든 인스턴스의 등록을 취소한 후 다음 절차를 수행하여 서비스를 삭제합니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 삭제하려는 서비스가 포함된 네임스페이스에 대해 해당 옵션을 선택합니다.
4. 네임스페이스: ***namespace-name*** 페이지에서 삭제하려는 서비스에 대해 해당 옵션을 선택합니다.
5. Delete(삭제)를 선택합니다.
6. 서비스 삭제를 확인합니다.

AWS CLI

- `delete-service` 명령을 사용하여 서비스를 삭제(*###* 값을 사용자 고유 값으로 대체)합니다.

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

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

3. delete_service()로 서비스를 삭제(*###* 값을 사용자 고유 값으로 대체)합니다.

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

예시 응답 출력

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

AWS Cloud Map 서비스 인스턴스

서비스 인스턴스에는 애플리케이션의 리소스(예: 웹 서버)를 찾는 방법에 대한 정보가 포함되어 있습니다. 인스턴스를 등록한 후 DNS 쿼리 또는 AWS Cloud Map DiscoverInstances API 작업을 사용하여 인스턴스를 찾습니다. 등록할 수 있는 리소스에는 다음이 포함되지만 이에 국한되지는 않습니다.

- Amazon EC2 인스턴스
- Amazon DynamoDB 테이블
- Amazon S3 버킷
- Amazon Simple Queue Service(Amazon SQS) 대기열
- Amazon APIs 배포된 API Amazon API Gateway

서비스 인스턴스의 속성 값을 지정할 수 있으며 클라이언트는 이러한 속성을 사용하여가 AWS Cloud Map 반환하는 리소스를 필터링할 수 있습니다. 예를 들어 애플리케이션은 특정 배포 단계의 리소스를 요청할 수 있습니다(예: BETA 또는 PROD). 버전 관리에 속성을 사용할 수도 있습니다.

다음 절차에서는 애플리케이션에 리소스를 서비스 인스턴스로 등록하고, 서비스에 등록된 인스턴스 목록을 보고, 특정 인스턴스 파라미터를 편집하고, 인스턴스 등록을 취소하는 방법을 설명합니다.

주제

리소스를 AWS Cloud Map 서비스 인스턴스로 등록

애플리케이션의 리소스를 AWS Cloud Map 서비스의 인스턴스로 등록할 수 있습니다. 예를 들어 사용자 데이터를 관리하는 모든 애플리케이션 리소스`users`에 대해 라는 서비스를 생성했다고 가정합니다. 그런 다음 사용자 데이터를이 서비스의 인스턴스로 저장하는 데 사용되는 DynamoDB 테이블을 등록할 수 있습니다.

> **Note**
>
> AWS Cloud Map 콘솔에서는 다음 기능을 사용할 수 없습니다.

- 콘솔을 사용하여 서비스 인스턴스를 등록하는 경우, 트래픽을 Elastic Load Balancing(ELB) 로드 밸런서로 라우팅하는 별칭 레코드를 생성할 수 없습니다. 인스턴스를 등록할 때 `AWS_ALIAS_DNS_NAME` 속성을 포함시켜야 합니다. 자세한 내용은 AWS Cloud Map API 참조의 인스턴스 등록를 참조하세요.
- 사용자 지정 상태 확인이 포함된 서비스를 사용하여 인스턴스를 등록하는 경우 사용자 지정 상태 확인에 대한 초기 상태를 지정할 수 없습니다. 기본적으로 사용자 지정 상태 확인의 초기 상태는 정상입니다. 초기 상태를 이상 있음으로 설정하려면 인스턴스를 프로그래밍 방식으로 등록하고 `AWS_INIT_HEALTH_STATUS` 속성을 포함시킵니다. 자세한 내용은 API 참조AWS Cloud Map 의 인스턴스 등록을 참조하세요.

서비스에 인스턴스를 등록하려면 다음 단계를 따릅니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 네임스페이스 페이지에서 서비스 인스턴스 등록을 위한 템플릿으로 사용하려는 서비스가 포함된 네임스페이스를 선택합니다.
4. 네임스페이스: ***namespace-name*** 페이지에서 사용하려는 서비스를 선택합니다.
5. 서비스: ***service-name*** 페이지에서 서비스 인스턴스 등록을 선택합니다.
6. 서비스 인스턴스 등록 페이지에서 인스턴스 유형을 선택합니다. 네임스페이스 인스턴스 검색 구성에 따라 IP 주소, Amazon EC2 인스턴스 ID 또는 IP 주소가 없는 리소스에 대한 기타 식별 정보를 지정하도록 선택할 수 있습니다.

   > **Note**
   >
   > HTTP 네임스페이스에서만 EC2 인스턴스를 선택할 수 있습니다.

7. 서비스 인스턴스 ID에 서비스 인스턴스와 연결된 식별자를 제공합니다.

> **Note**
>
> 기존 인스턴스를 업데이트하려면 업데이트하려는 인스턴스와 연결된 식별자를 제공합니다. 그런 다음 다음 단계에 따라 값을 업데이트하고 인스턴스를 다시 등록합니다.

8. 선택한 인스턴스 유형에 따라 다음 단계를 수행합니다.

> **Important**
>
> 사용자 지정 속성을 지정할 때는 키에 `AWS_` 접두사(대소문자를 구분하지 않음)를 사용할 수 없습니다.

인스턴스 유형	단계(Steps)
IP 주소	a. 표준 속성의 IPv4 주소에서 애플리케이션이 서비스 인스턴스와 연결된 리소스에 액세스할 수 있는 IPv4 주소를 입력합니다. b. IPv6 주소의 경우 애플리케이션이 서비스 인스턴스와 연결된 리소스에 액세스할 수 있는 IPv6 IP 주소를 제공합니다. c. 포트에서이 서비스 인스턴스와 연결된 리소스에 액세스하기 위해 애플리케이션에서 포함해야 하는 포트를 지정합니다. 서비스에 SRV 레코드 또는 Amazon Route 53 상태 확인이 포함된 경우 포트가 필요합니다.

인스턴스 유형	단계(Steps)	
	d. (선택 사항) 사용자 지정 속성에서 리소스와 연결할 키-값 페어를 지정합니다.	
EC2 인스턴스	a. EC2 인스턴스 ID에서 AWS Cloud Map 서비스 인스턴스로 등록할 Amazon EC2 인스턴스의 ID를 선택합니다. b. (선택 사항) 사용자 지정 속성에서 리소스와 연결할 키-값 페어를 지정합니다.	

인스턴스 유형	단계(Steps)	
다른 리소스에 대한 정보 식별	a. 표준 속성에서 서비스 구성에 CNAME DNS 레코드가 포함된 경우 CNAME 필드가 표시됩니다. CNAME에서 Route 53가 DNS 쿼리에 대한 응답으로 반환할 도메인 이름을 지정합니다(예: `example.com`). b. 사용자 지정 속성에서 IP 주소 또는 Amazon EC2 인스턴스 ID가 아닌 리소스의 식별 정보를 키-값 페어로 지정합니다. 예를 들어 라는 키를 지정`function`하고 Lambda 함수의 이름을 값으로 제공하여 Lambda 함수를 등록할 수 있습니다. 라는 키를 지정`name`하고 프로그래밍 방식 인스턴스 검색에 사용할 수 있는 이름을 제공할 수도 있습니다.	

9. 서비스 인스턴스 등록을 선택합니다.

AWS CLI

- `RegisterInstance` 요청을 제출하는 경우:
  - `ServiceId`에 지정된 서비스에서 정의한 각 DNS 레코드에 대해 해당 네임스페이스와 연결된 호스팅 영역에서 레코드가 생성되거나 업데이트됩니다.
  - 서비스에 `HealthCheckConfig`가 포함된 경우, 상태 확인 구성의 설정을 기반으로 상태 확인이 생성됩니다.

- 모든 상태 확인은 새 레코드 또는 업데이트된 각 레코드와 연결됩니다.

register-instance 명령을 사용하여 서비스 인스턴스를 등록(### 값을 사용자 고유 값으로 대체)합니다.

```
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. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

3. RegisterInstance 요청을 제출하는 경우:

   - ServiceId에 지정된 서비스에서 정의한 각 DNS 레코드에 대해 해당 네임스페이스와 연결된 호스팅 영역에서 레코드가 생성되거나 업데이트됩니다.
   - 서비스에 HealthCheckConfig가 포함된 경우, 상태 확인 구성의 설정을 기반으로 상태 확인이 생성됩니다.
   - 모든 상태 확인은 새 레코드 또는 업데이트된 각 레코드와 연결됩니다.

   register_instance()로 서비스 인스턴스를 등록(### 값을 사용자 고유 값으로 대체)합니다.

```
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)
```

예시 응답 출력

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

AWS Cloud Map 서비스 인스턴스 나열

서비스를 사용하여 등록한 서비스 인스턴스 목록을 보려면 다음 절차를 수행합니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 서비스 인스턴스를 나열하려는 서비스가 포함된 네임스페이스의 이름을 선택합니다.
4. 서비스 인스턴스를 생성하는 데 사용한 서비스의 이름을 선택합니다. 서비스 인스턴스 아래에 인스턴스 목록이 표시됩니다. 검색 필드에 인스턴스 ID를 입력하여 특정 인스턴스를 나열할 수 있습니다. 생성자 필드에는 인스턴스를 AWS 계정 등록한의 ID가 표시됩니다.

> **Note**
>
> 인스턴스가 등록된 네임스페이스가 공유 네임스페이스인 경우 에서 생성한의 AWS 계정 ID가 계정 ID와 동일하지 않을 수 있습니다. 공유 네임스페이스에 대한 자세한 내용은 섹션을 참조하세요공유 AWS Cloud Map 네임스페이스.

AWS CLI

- `list-instances` 명령을 사용하여 서비스 인스턴스를 나열합니다(### 값을 자신의 것으로 대체).

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

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우 Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 servicediscovery를 사용하세요.

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

3. list_instances()로 서비스 인스턴스를 나열합니다(### 값을 자체 값으로 대체).

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

예시 응답 출력

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

AWS Cloud Map 서비스 인스턴스 업데이트

업데이트하려는 값에 따라 다음 두 가지 방법으로 서비스 인스턴스를 업데이트할 수 있습니다.

- 값 업데이트: 사용자 지정 속성을 포함하여 서비스 인스턴스를 등록할 때 지정한 값을 업데이트하려면 서비스 인스턴스를 다시 등록하고 모든 값을 다시 지정해야 합니다. 의 단계에 따라 서비스 인스턴스 ID에 대한 기존 서비스 인스턴스의 인스턴스 ID를 리소스를 AWS Cloud Map 서비스 인스턴스로 등록지정합니다.

  또는 `RegisterInstance` API를 사용할 수 있습니다. `InstanceId` 및 `ServiceId` 파라미터를 사용하여 기존 인스턴스 및 서비스의 ID를 지정하고 다른 값을 다시 지정할 수 있습니다.
- 사용자 지정 속성만 업데이트: 서비스 인스턴스의 사용자 지정 속성만 업데이트하려는 경우 인스턴스를 재등록할 필요가 없습니다. 해당 값만 업데이트하면 됩니다. 서비스 인스턴스의 사용자 지정 속성 업데이트을(를) 참조하세요.

서비스 인스턴스의 사용자 지정 속성 업데이트

서비스 인스턴스에 대한 사용자 지정 속성만 업데이트하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 네임스페이스 페이지에서 서비스 인스턴스를 등록하는 데 원래 사용한 서비스가 포함된 네임스페이스를 선택합니다.
4. 네임스페이스: ***namespace-name*** 페이지에서 서비스 인스턴스를 등록하는 데 사용한 서비스를 선택합니다.
5. 서비스: ***service-name*** 페이지에서 업데이트하려는 서비스 인스턴스의 이름을 선택합니다.
6. 사용자 지정 속성 섹션에서 편집을 선택합니다.
7. 서비스 인스턴스 편집: ***instance-name*** 페이지에서 사용자 지정 속성을 추가, 제거 또는 업데이트합니다. 기존 속성의 키와 값을 모두 업데이트할 수 있습니다.
8. 서비스 인스턴스 업데이트를 선택합니다.

AWS Cloud Map 서비스 인스턴스 등록 취소

서비스를 삭제하기 전에 해당 서비스를 사용해 등록한 모든 서비스 인스턴스를 등록 취소해야 합니다.

서비스 인스턴스를 등록 취소하려면 다음 절차를 수행합니다.

AWS Management Console

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스를 선택합니다.
3. 등록 취소하려는 서비스 인스턴스가 포함된 네임스페이스에 대해 해당 옵션을 선택합니다.
4. 네임스페이스: ***namespace-name*** 페이지에서 서비스 인스턴스를 등록하는 데 사용한 서비스를 선택합니다.
5. 서비스: ***### ##*** 페이지에서 등록 취소하려는 서비스 인스턴스를 선택합니다.
6. 등록 취소(Deregister)를 선택합니다.
7. 서비스 인스턴스 등록 취소를 확인합니다.

AWS CLI

- `deregister-instance` 명령을 사용하여 서비스 인스턴스를 등록 취소(*###* 값을 사용자 고유 값으로 대체)합니다. 이 명령은 Amazon Route 53 DNS 레코드와 지정된 인스턴스에 대해 AWS Cloud Map 생성된 모든 상태 확인을 삭제합니다.

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

AWS SDK for Python (Boto3)

1. 아직 Boto3이 설치되지 않은 경우, Boto3을 사용하여 여기에서 설치, 구성, 사용에 대한 지침을 찾을 수 있습니다.
2. Boto3을 가져와서 서비스로 `servicediscovery`를 사용하세요.

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

3. `deregister-instance()`로 서비스 인스턴스를 등록 취소(*###* 값을 사용자 고유 값으로 대체)합니다. 이 명령은 Amazon Route 53 DNS 레코드와 지정된 인스턴스에 대해 AWS Cloud Map 생성된 모든 상태 확인을 삭제합니다.

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

예시 응답 출력

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

의 보안 AWS Cloud Map

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이를 클라우드의 보안과 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 AWS 규정 준수 프로그램 제공 범위 내 서비스를 AWS Cloud Map참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

다음 설명서는를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다 AWS Cloud Map. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 AWS Cloud Map 를 구성하는 방법을 보여줍니다. 또한 AWS Cloud Map 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- 에 대한 자격 증명 및 액세스 관리 AWS Cloud Map
- 에 대한 규정 준수 검증 AWS Cloud Map
- 의 복원력 AWS Cloud Map
- 의 인프라 보안 AWS Cloud Map

에 대한 자격 증명 및 액세스 관리 AWS Cloud Map

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 누가 AWS Cloud Map 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청(참조AWS Cloud Map 자격 증명 및 액세스 문제 해결)
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출(AWS Cloud Map 에서 IAM을 사용하는 방법 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성(에 대한 자격 증명 기반 정책 예제 AWS Cloud Map 참조)

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증되어야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 연동을 사용하여 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS Cloud Map 에서 IAM을 사용하는 방법

IAM을 사용하여 액세스를 관리하기 전에 사용할 수 있는 IAM 기능에 대해 AWS Cloud Map알아봅니다 AWS Cloud Map.

IAM 특성	AWS Cloud Map 지원
자격 증명 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예

IAM 특성	AWS Cloud Map 지원
전달 액세스 세션(FAS)	예
서비스 역할	아니요
서비스 연결 역할	예

AWS Cloud Map 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

에 대한 자격 증명 기반 정책 AWS Cloud Map

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

에 대한 자격 증명 기반 정책 예제 AWS Cloud Map

자격 AWS Cloud Map 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS Cloud Map.

내의 리소스 기반 정책 AWS Cloud Map

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는가 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM에서 교차 계정 리소스 액세스를 참조하세요.

> **Note**
>
> AWS Resource Access Manager (AWS RAM)를 사용하여 AWS Cloud Map 네임스페이스를 안전하게 공유할 수 있습니다. 리소스 기반 정책은 AWS RAM 서비스에 의해 네임스페이스에 적용됩니다. 자세한 내용은 공유 AWS Cloud Map 네임스페이스 단원을 참조하십시오.

에 대한 정책 작업 AWS Cloud Map

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

AWS Cloud Map 작업 목록을 보려면 서비스 승인 참조의에서 정의한 작업을 AWS Cloud Map 참조하세요.

의 정책 작업은 작업 앞에 다음 접두사를 AWS Cloud Map 사용합니다.

```
servicediscovery
```

단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

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

자격 AWS Cloud Map 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS Cloud Map.

에 대한 정책 리소스 AWS Cloud Map

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

AWS Cloud Map 리소스 유형 및 해당 ARNs 목록을 보려면 서비스 승인 참조의에서 정의한 리소스를 AWS Cloud Map 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 AWS Cloud Map 에서 정의한 작업을 참조하세요.

자격 AWS Cloud Map 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS Cloud Map.

에 대한 정책 조건 키 AWS Cloud Map

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

AWS Cloud Map 조건 키 목록을 보려면 서비스 승인 참조의에 대한 조건 키를 AWS Cloud Map 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 에서 정의한 작업을 AWS Cloud Map 참조하세요.

AWS Cloud Map 는 IAM 정책에 대한 세분화된 필터링을 제공하는 데 사용할 수 있는 다음과 같은 서비스별 조건 키를 지원합니다.

servicediscovery:NamespaceArn

관련 네임스페이스에 대해 Amazon 리소스 이름(ARN)을 지정하여 객체를 가져올 수 있는 필터입니다.

servicediscovery:NamespaceName

관련 네임스페이스의 이름을 지정하여 객체를 가져올 수 있는 필터입니다.

servicediscovery:ServiceArn

관련 서비스에 대해 Amazon 리소스 이름(ARN)을 지정하여 객체를 가져올 수 있는 필터입니다.

servicediscovery:ServiceName

관련 서비스의 이름을 지정하여 객체를 가져올 수 있는 필터입니다.

servicediscovery:ServiceCreatedByAccount

서비스를 생성한의 ID를 지정하여 객체를 가져올 수 AWS 계정 있는 필터입니다.

자격 AWS Cloud Map 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS Cloud Map.

ACLs AWS Cloud Map

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

를 사용한 ABAC AWS Cloud Map

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

에서 임시 자격 증명 사용 AWS Cloud Map

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 임시 보안 자격 증명 및 IAM으로 작업하는 AWS 서비스 섹션을 참조하세요.

에 대한 전달 액세스 세션 AWS Cloud Map

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

에 대한 서비스 역할 AWS Cloud Map

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스 AWS에 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 AWS Cloud Map 기능이 중단될 수 있습니다. 에서 관련 지침을 AWS Cloud Map 제공하는 경우에만 서비스 역할을 편집합니다.

에 대한 서비스 연결 역할 AWS Cloud Map

서비스 연결 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스를 참조하세요. 서비스 연결 역할 열에서 Yes가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

에 대한 자격 증명 기반 정책 예제 AWS Cloud Map

기본적으로 사용자 및 역할에는 AWS Cloud Map 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARNs 형식을 포함하여 AWS Cloud Map에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 에 사용되는 작업, 리소스 및 조건 키를 AWS Cloud Map 참조하세요.

주제

- 정책 모범 사례
- AWS Cloud Map 콘솔 사용
- AWS Cloud Map 콘솔 액세스 예제
- AWS Cloud Map 사용자가 자신의 권한을 볼 수 있도록 허용
- 모든 AWS Cloud Map 리소스에 대한 읽기 액세스 허용
- AWS Cloud Map 서비스 인스턴스 예제
- AWS Cloud Map 서비스 예제 생성
- AWS Cloud Map 네임스페이스 생성 예제

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS Cloud Map 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것

이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.

- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

AWS Cloud Map 콘솔 사용

AWS Cloud Map 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은의 AWS Cloud Map 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 AWS Cloud Map 콘솔을 계속 사용할 수 있도록 하려면 또는 *ReadOnly* AWS 관리형 AWS Cloud Map *ConsoleAccess* 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

AWS Cloud Map 콘솔 액세스 예제

AWS Cloud Map 콘솔에 대한 전체 액세스 권한을 부여하려면 다음 권한 정책에서 권한을 부여합니다.

JSON

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

다음은 권한이 필요한 이유입니다.

`servicediscovery:*`

모든 AWS Cloud Map 작업을 수행할 수 있습니다.

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

퍼블릭 및 프라이빗 DNS 네임스페이스를 생성하고 삭제할 때 호스팅 영역을 AWS Cloud Map 관리할 수 있습니다.

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

서비스를 생성할 때 Amazon Route 53 상태 확인을 포함할 때 상태 확인을 AWS Cloud Map 관리할 수 있습니다.

ec2:DescribeVpcs 및 **ec2:DescribeRegions**

가 프라이빗 호스팅 영역을 AWS Cloud Map 관리하도록 합니다.

AWS Cloud Map 사용자가 자신의 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여 줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
```

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

모든 AWS Cloud Map 리소스에 대한 읽기 액세스 허용

다음 권한 정책은 사용자에게 모든 AWS Cloud Map 리소스에 대한 읽기 전용 액세스 권한을 부여합니다.

JSON

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

AWS Cloud Map 서비스 인스턴스 예제

다음 예제에서는 사용자에게 서비스 인스턴스를 등록, 등록 취소 및 검색할 수 있는 권한을 부여하는 권한 정책을 보여줍니다. `Sid`(문 ID)는 선택 사항입니다.

JSON

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

이 정책은 서비스 인스턴스를 등록 및 관리하는 데 필요한 작업에 대한 권한을 부여합니다. 인스턴스를 등록 및 등록 취소할 때가 Route 53 레코드 및 상태 확인을 AWS Cloud Map 생성, 업데이트 및 삭제하기 때문에 퍼블릭 또는 프라이빗 DNS 네임스페이스를 사용하는 경우 Route 53 권한이 필요합니다. 의 와일드카드 문자(*)는 현재 AWS 계정이 소유한 모든 AWS Cloud Map 인스턴스와 Route 53 레코드 및 상태 확인에 대한 액세스 권한을 `Resource` 부여합니다.

AWS Cloud Map 서비스 예제 생성

IAM 자격 증명이 서비스를 생성 AWS Cloud Map 하도록 허용하는 권한 정책을 추가할 때 리소스 필드에 네임스페이스와 서비스의 Amazon 리소스 이름(ARN)을 모두 AWS Cloud Map 지정해야 합니다. ARN에는 리전, 계정 ID 및 네임스페이스 ID가 포함됩니다. 서비스의 서비스 ID가 무엇인지 아직 알 수 없으므로 와일드카드를 사용하는 것이 좋습니다. 다음은 정책 코드 조각의 예입니다.

JSON

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

```
            "Effect": "Allow",
            "Action": [
                "servicediscovery:CreateService"
            ],
            "Resource": [
                "arn:aws:servicediscovery:us-east-1:111122223333:namespace/ns-
p32123EXAMPLE",
                "arn:aws:servicediscovery:us-east-1:111122223333:service/*"
            ]
        }
    ]
}
```

AWS Cloud Map 네임스페이스 생성 예제

다음 권한 정책은 사용자가 모든 유형의 AWS Cloud Map 네임스페이스를 생성할 수 있도록 허용합니다.

JSON

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

AWS 에 대한 관리형 정책 AWS Cloud Map

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 줍니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 가이드의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AWSCloudMapDiscoverInstanceAccess

`AWSCloudMapDiscoverInstanceAccess`를 IAM 엔티티에 연결할 수 있습니다. AWS Cloud Map Discovery API에 대한 액세스를 제공합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSCloudMapDiscoverInstanceAccess를 확인하세요.

AWS 관리형 정책: AWSCloudMapReadOnlyAccess

`AWSCloudMapReadOnlyAccess`를 IAM 엔티티에 연결할 수 있습니다. 모든 AWS Cloud Map 작업에 대한 읽기 전용 액세스 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSCloudMapReadOnlyAccess를 확인하세요.

AWS 관리형 정책: AWSCloudMapRegisterInstanceAccess

`AWSCloudMapRegisterInstanceAccess`를 IAM 엔티티에 연결할 수 있습니다. 네임스페이스 및 서비스에 대한 읽기 전용 액세스 권한을 부여하고, 서비스 인스턴스를 등록 및 등록 취소하는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSCloudMapRegisterInstanceAccess를 확인하세요.

AWS 관리형 정책: AWSCloudMapFullAccess

`AWSCloudMapFullAccess`를 IAM 엔티티에 연결할 수 있습니다. 모든 AWS Cloud Map 작업에 대한 전체 액세스 권한을 제공합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSCloudMapFullAccess를 확인하세요.

AWS Cloud Map AWS 관리형 정책에 대한 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 AWS Cloud Map 이후부터의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 변경 사항에 대한 자동 알림을 받으려면 AWS Cloud Map 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경	설명	Date
AWSCloudMapDiscoverInstanceAccess, AWSCloudMapRegisterInstanceAccess, AWSCloudMapReadOnlyAccess – 기존 정책 업데이트.	AWS Cloud Map 는 새 AWS Cloud Map `DiscoverInstanceRevision` API 작업에 대한 액세스를 제공하기 위해 이러한 정책을 업데이트했습니다.	2023년 8월 15일

AWS Cloud Map API 권한 참조

액세스 제어를 설정하고 IAM 자격 증명에 연결할 수 있는 권한 정책(자격 증명 기반 정책)을 작성할 때 다음 목록을 참조로 사용할 수 있습니다. 목록에는 각 AWS Cloud Map API 작업과 액세스 권한을 부여해야 하는 작업이 포함됩니다. 정책의 `Action` 필드에 작업을 지정합니다. `Resource` 필드 또는 IAM 정책에서 지정해야 하는 리소스 값에 대한 자세한 내용은 서비스 승인 참조의에 대한 작업, 리소스 및 조건 키를 AWS Cloud Map 참조하세요.

일부 작업에 대해 IAM 정책에서 AWS Cloud Map특정 조건 키를 사용할 수 있습니다. 자세한 내용은 서비스 권한 부여 참조의 AWS Cloud Map조건 키를 참조하세요.

작업을 지정하려면 `servicediscovery` 접두사 다음에 API 작업 이름을 사용합니다(예: `servicediscovery:CreatePublicDnsNamespace` 및 `route53:CreateHostedZone`).

AWS Cloud Map 작업에 필요한 권한

CreateHttpNamespace

필수 권한(API 작업):

- `servicediscovery:CreateHttpNamespace`

CreatePrivateDnsNamespace

필수 권한(API 작업):

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

CreatePublicDnsNamespace

필수 권한(API 작업):

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

CreateService

필요한 권한(API 작업): `servicediscovery:CreateService`

DeleteNamespace

필수 권한(API 작업):

- `servicediscovery:DeleteNamespace`

DeleteService

필요한 권한(API 작업): `servicediscovery:DeleteService`

DeleteServiceAttributes

필요한 권한(API 작업): `servicediscovery:DeleteServiceAttributes`

DeregisterInstance

필수 권한(API 작업):

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

DiscoverInstances

필요한 권한(API 작업): servicediscovery:DiscoverInstances

GetInstance

필요한 권한(API 작업): servicediscovery:GetInstance

GetInstancesHealthStatus

필요한 권한(API 작업): servicediscovery:GetInstancesHealthStatus

GetNamespace

필요한 권한(API 작업): servicediscovery:GetNamespace

GetOperation

필요한 권한(API 작업): servicediscovery:GetOperation

GetService

필요한 권한(API 작업): servicediscovery:GetService

GetServiceAttributes

필요한 권한(API 작업): servicediscovery:GetServiceAttributes

ListInstances

필요한 권한(API 작업): servicediscovery:ListInstances

ListNamespaces

필요한 권한(API 작업): servicediscovery:ListNamespaces

ListOperations

필요한 권한(API 작업): servicediscovery:ListOperations

ListServices

필요한 권한(API 작업): `servicediscovery:ListServices`

ListTagsForResource

필요한 권한(API 작업): `servicediscovery:ListTagsForResource`

RegisterInstance

필수 권한(API 작업):

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

TagResource

필요한 권한(API 작업): `servicediscovery:TagResource`

UntagResource

필요한 권한(API 작업): `servicediscovery:UntagResource`

UpdateHttpNamespace

필요한 권한(API 작업): `servicediscovery:UpdateHttpNamespace`

UpdateInstanceCustomHealthStatus

필요한 권한(API 작업): `servicediscovery:UpdateInstanceCustomHealthStatus`

UpdatePrivateDnsNamespace

필수 권한(API 작업):

- `servicediscovery:UpdatePrivateDnsNamespace`
- `route53:ChangeResourceRecordSets`

UpdatePublicDnsNamespace

필수 권한(API 작업):

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

UpdateService

필수 권한(API 작업):

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

UpdateServiceAttributes

필요한 권한(API 작업): servicediscovery:UpdateServiceAttributes

AWS Cloud Map 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단 AWS Cloud Map 하고 수정할 수 있습니다.

주제

- 에서 작업을 수행할 권한이 없음 AWS Cloud Map
- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 AWS Cloud Map 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.

에서 작업을 수행할 권한이 없음 AWS Cloud Map

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 servicediscovery:*GetWidget* 권한이 없을 때 발생합니다.

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

이 경우, servicediscovery:*GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS Cloud Map에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 AWS Cloud Map에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

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

이 경우, Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 AWS Cloud Map 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- 에서 이러한 기능을 AWS Cloud Map 지원하는지 여부를 알아보려면 섹션을 참조하세요AWS Cloud Map 에서 IAM을 사용하는 방법.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

에 대한 규정 준수 검증 AWS Cloud Map

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 범위 내 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서를 AWS 서비스참조하세요.

의 복원력 AWS Cloud Map

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며,이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결됩니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS Cloud Map 는 주로 글로벌 서비스입니다. 그러나 AWS Cloud Map 를 사용하여 Amazon EC2 인스턴스 및 Elastic Load Balancing 로드 밸런서와 같은 특정 리전의 리소스 상태를 확인하는 Route 53 상태 확인을 생성할 수 있습니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

의 인프라 보안 AWS Cloud Map

관리형 서비스인는 AWS 글로벌 네트워크 보안으로 보호 AWS Cloud Map 됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요 AWS .

AWS 에서 게시한 API 호출을 사용하여 네트워크를 AWS Cloud Map 통해에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.

- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

인터페이스 VPC 엔드포인트를 사용하도록 AWS Cloud Map 를 구성하여 VPC의 보안 태세를 개선할 수 있습니다. 자세한 내용은 인터페이스 엔드포인트를 AWS Cloud Map 사용한 액세스(AWS PrivateLink) 단원을 참조하십시오.

인터페이스 엔드포인트를 AWS Cloud Map 사용한 액세스(AWS PrivateLink)

AWS PrivateLink 를 사용하여 VPC와 간에 프라이빗 연결을 생성할 수 있습니다 AWS Cloud Map. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 Direct Connect 연결을 사용하지 않고 VPC에 있는 AWS Cloud Map 것처럼에 액세스할 수 있습니다. VPC의 인스턴스에서 AWS Cloud Map API에 액세스하는 데는 퍼블릭 IP 주소가 필요하지 않습니다.

AWS PrivateLink에서 제공되는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 AWS Cloud Map로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은 AWS PrivateLink 안내서의 AWS PrivateLink를 통해 AWS 서비스 에 액세스를 참조하세요.

에 대한 고려 사항 AWS Cloud Map

에 대한 인터페이스 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의 고려 사항을 AWS Cloud Map검토하세요.

Amazon VPC에 인터넷 게이트웨이가 없고 작업에서 `awslogs` 로그 드라이버를 사용하여 로그 정보를 CloudWatch Logs로 전송하는 경우에는 CloudWatch Logs용 인터페이스 VPC 엔드포인트를 생성해야 합니다. 자세한 정보는 Amazon CloudWatch Logs 사용 설명서의 인터페이스 VPC 엔드포인트에서 CloudWatch Logs 사용을 참조하세요.

VPC 엔드포인트는 AWS 리전 간 요청을 지원하지 않습니다. API 호출을 AWS Cloud Map(으)로 발행할 계획인 동일 리전에서 엔드포인트를 생성해야 합니다.

VPC 엔드포인트는 Amazon Route 53을 통해 Amazon이 제공하는 DNS만 지원합니다. 자체 DNS를 사용하는 경우에는 조건부 DNS 전달을 사용할 수 있습니다. 자세한 정보는 Amazon VPC 사용 설명서의 DHCP 옵션 세트를 참조하세요.

VPC 엔드포인트에 연결된 보안 그룹은 Amazon VPC의 프라이빗 서브넷에서 443 포트로 들어오는 연결을 허용해야 합니다.

에 대한 인터페이스 엔드포인트 생성 AWS Cloud Map

Amazon VPC 콘솔 또는 AWS Command Line Interface ()를 AWS Cloud Map 사용하여 용 인터페이스 엔드포인트를 생성할 수 있습니다AWS CLI. 자세한 내용은 AWS PrivateLink 안내서의 인터페이스 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 AWS Cloud Map 사용하여 용 인터페이스 엔드포인트를 생성합니다.

> **Note**
>
> 이 두 엔드포인트에서는 `DiscoverInstances` API를 사용할 수 없습니다.

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

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

다음 서비스 이름을 사용하여 `DiscoverInstances` API에 액세스할 수 있도록 AWS Cloud Map 데이터 영역에 대한 인터페이스 엔드포인트를 생성합니다.

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

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

> **Note**
>
> 데이터 영역 엔드포인트의 리전 또는 영역 VPCE DNS 이름을 사용하여 `DiscoverInstances`를 호출할 때는 호스트 접두사 삽입을 비활성화해야 합니다. AWS CLI 및 AWS SDKs는 각 API 작업을 호출할 때 서비스 엔드포인트 앞에 다양한 호스트 접두사를 추가합니다. 그러면 VPC 엔드포인트를 지정할 때 잘못된 URL이 생성됩니다.

인터페이스 엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우, 리전에 대한 기본 DNS 이름(예: AWS Cloud Map)을 사용하여 에 API 요청을 할 수 있습니다. 예를 들어 `servicediscovery.us-east-1.amazonaws.com`입니다.

VPCE AWS PrivateLink 연결은가 AWS Cloud Map 지원되는 모든 리전에서 지원되지만 고객은 엔드포인트를 정의하기 전에 VPCE를 지원하는 가용 영역을 확인해야 합니다. 리전의 인터페이스 VPC 엔드포인트에서 지원되는 가용 영역을 확인하려면 describe-vpc-endpoint-services 명령을 사용하거나 AWS Management Console을 사용하세요. 예를 들어 다음 명령은 미국 동부(오하이오) 리전 내에 AWS Cloud Map 인터페이스 VPC 엔드포인트를 배포할 수 있는 가용 영역을 반환합니다.

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

모니터링 AWS Cloud Map

모니터링은 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 중요한 역할을 합니다. AWS 솔루션의 모든 부분에서 모니터링 데이터를 수집해야 다중 지점 장애가 발생할 경우 더 쉽게 디버깅할 수 있습니다. 하지만 모니터링을 시작하기 전에 다음 질문에 대한 답변을 포함하는 모니터링 계획을 작성해야 합니다.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

주제

- 를 사용하여 AWS Cloud Map API 호출 로깅 AWS CloudTrail

를 사용하여 AWS Cloud Map API 호출 로깅 AWS CloudTrail

AWS Cloud Map 는 사용자AWS CloudTrail, 역할 또는가 수행한 작업에 대한 레코드를 제공하는 서비스인와 통합됩니다 AWS 서비스. CloudTrail은에 대한 모든 API 호출을 이벤트 AWS Cloud Map 로 캡처합니다. 캡처되는 호출에는 AWS Cloud Map 콘솔의 호출과 AWS Cloud Map API 작업에 대한 코드 호출이 포함됩니다. CloudTrail에서 수집한 정보를 사용하여 수행된 요청, 요청이 수행된 AWS Cloud Map IP 주소, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트 사용자로 했는지 사용자 보안 인증으로 했는지 여부.
- IAM Identity Center 사용자를 대신하여 요청이 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화되며 CloudTrail 이벤트 기록에 자동으로 액세스할 수 있습니다. CloudTrail 이벤트 기록은 지난 90일 간 AWS 리전의 관리 이벤트에 대해 보기, 검색 및 다운로드가 가능하고, 수정이 불가능한 레코드를 제공합니다. 자세한 설명은 AWS CloudTrail 사용 설명서의 CloudTrail 이벤트 기록 작업을 참조하세요. Event history(이벤트 기록) 보기는 CloudTrail 요금이 부과되지 않습니다.

AWS 계정 지난 90일 동안의 이벤트를 지속적으로 기록하려면 추적 또는 CloudTrail Lake 이벤트 데이터 스토어를 생성합니다.

CloudTrail 추적

CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 를 사용하여 생성된 모든 추적 AWS Management Console 은 다중 리전입니다. AWS CLI를 사용하여 단일 리전 또는 다중 리전 추적을 생성할 수 있습니다. 계정의 모든 AWS 리전 에서 활동을 캡처하므로 다중 리전 추적을 생성하는 것이 좋습니다. 단일 리전 추적을 생성하는 경우 추적의 AWS 리전에 로깅된 이벤트만 볼 수 있습니다. 추적에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 Creating a trail for your AWS 계정 및 Creating a trail for an organization을 참조하세요.

CloudTrail에서 추적을 생성하여 진행 중인 관리 이벤트의 사본 하나를 Amazon S3 버킷으로 무료로 전송할 수는 있지만, Amazon S3 스토리지 요금이 부과됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요. Amazon S3 요금에 대한 자세한 내용은 Amazon S3 요금을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어

CloudTrail Lake를 사용하면 이벤트에 대해 SQL 기반 쿼리를 실행할 수 있습니다. CloudTrail Lake는 행 기반 JSON 형식의 기존 이벤트를 Apache ORC 형식으로 변환합니다. ORC는 빠른 데이터 검색에 최적화된 열 기반 스토리지 형식입니다. 이벤트는 이벤트 데이터 스토어로 집계되며, 이벤트 데이터 스토어는 고급 이벤트 선택기를 적용하여 선택한 기준을 기반으로 하는 변경 불가능한 이벤트 컬렉션입니다. 이벤트 데이터 스토어에 적용하는 선택기는 어떤 이벤트가 지속되고 쿼리할 수 있는지 제어합니다. CloudTrail Lake에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 AWS CloudTrail Lake 작업을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어 및 쿼리에는 비용이 발생합니다. 이벤트 데이터 스토어를 생성할 때 이벤트 데이터 스토어에 사용할 요금 옵션을 선택합니다. 요금 옵션에 따라 이벤트 모으기 및 저장 비용과 이벤트 데이터 스토어의 기본 및 최대 보존 기간이 결정됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요.

AWS Cloud Map CloudTrail의 데이터 이벤트

데이터 이벤트는 리소스에서 또는 리소스에서 수행되는 리소스 작업에 대한 정보를 제공합니다(예: 네임스페이스에서 등록된 인스턴스 검색). 이를 데이터 영역 작업이라고도 합니다. 데이터 이벤트가 대량 활동인 경우도 있습니다. 기본적으로 CloudTrail은 데이터 이벤트를 로깅하지 않습니다. CloudTrail 이벤트 기록은 데이터 이벤트를 기록하지 않습니다.

데이터 이벤트에는 추가 요금이 적용됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요.

CloudTrail 콘솔 AWS CLI또는 CloudTrail API 작업을 사용하여 AWS Cloud Map 리소스 유형에 대한 데이터 이벤트를 로깅할 수 있습니다. 데이터 이벤트를 로깅하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 Logging data events with the AWS Management Console 및 Logging data events with the AWS Command Line Interface를 참조하세요.

다음 표에는 데이터 이벤트를 로깅할 수 있는 AWS Cloud Map 리소스 유형이 나열되어 있습니다. 데이터 이벤트 유형(콘솔) 열에는 CloudTrail 콘솔의 데이터 이벤트 유형 목록에서 선택할 값이 표시됩니다. resources.type 값 열에는 AWS CLI 또는 CloudTrail APIs를 사용하여 고급 이벤트 선택기를 구성할 때 지정하는 `resources.type` 값이 표시됩니다. CloudTrail에 로깅되는 데이터 API 열에는 리소스 유형에 대해 CloudTrail에 로깅된 API 호출이 표시됩니다.

데이터 이벤트 유형(콘솔)	resources.type 값	CloudTrail에 로깅되는 데이터 API
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances • DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • DiscoverInstancesRevision • GetServiceAttributes

`eventName`, `readOnly` 및 `resources.ARN` 필드를 필터링하여 중요한 이벤트만 로깅하도록 고급 이벤트 선택기를 구성할 수 있습니다. 이러한 필드에 대한 자세한 내용은 AWS CloudTrail API 참조의 AdvancedFieldSelector 섹션을 참조하세요.

다음 예제에서는 모든 AWS Cloud Map 데이터 이벤트를 로깅하도록 고급 이벤트 선택기를 구성하는 방법을 보여줍니다.

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

AWS Cloud Map CloudTrail의 관리 이벤트

관리 이벤트는의 리소스에서 수행되는 관리 작업에 대한 정보를 제공합니다 AWS 계정. 이를 컨트롤 플레인 작업이라고도 합니다. 기본적으로 CloudTrail은 관리 이벤트를 로깅합니다.

AWS Cloud Map 는 모든 AWS Cloud Map 컨트롤 플레인 작업을 관리 이벤트로 기록합니다. CloudTrail에 AWS Cloud Map 로깅하는 AWS Cloud Map 컨트롤 플레인 작업 목록은 AWS Cloud Map API 참조를 참조하세요.

AWS Cloud Map 이벤트 예제

이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청된 API 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 추적이 아니므로 이벤트가 특정 순서로 표시되지 않습니다.

다음 예제에서는 `CreateHTTPNamespace` 작업을 보여주는 CloudTrail 관리 이벤트를 보여줍니다.

```
{
    "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"
}
```


다음 예제에서는 `DiscoverInstances` 작업을 보여주는 CloudTrail 데이터 이벤트를 보여줍니다.


```
{
```

```
            "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 레코드 콘텐츠에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail record contents를 참조하세요.

AWS Cloud Map 리소스에 태그 지정

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다.

태그를 사용하면 용도, 소유자 또는 환경을 기준으로 AWS 리소스를 분류할 수 있습니다. 동일한 유형의 리소스가 많은 경우 할당한 태그에 따라 특정 리소스를 빠르게 식별할 수 있습니다. 예를 들어 각 AWS Cloud Map 서비스의 소유자 및 스택 수준을 추적하는 데 도움이 되도록 서비스에 대한 태그 세트를 정의할 수 있습니다. 각 리소스 유형에 대해 일관된 태그 키 집합을 고안하는 것이 좋습니다.

태그가 리소스에 자동으로 할당되는 것은 아닙니다. 태그를 추가한 후에는 언제든지 태그 키와 값을 편집하거나 리소스에서 태그를 제거할 수 있습니다. 리소스를 삭제하면 리소스 태그도 삭제됩니다.

태그는에 의미가 없으며 엄격하게 문자열로 해석 AWS Cloud Map 됩니다. 태그의 값을 빈 문자열로 설정할 수 있지만 태그의 값을 Null로 설정할 수는 없습니다. 해당 리소스에 대해 키가 기존 태그와 동일한 태그를 추가하는 경우 새 값이 이전 값을 덮어씁니다.

AWS Management Console, AWS CLI및 AWS Cloud Map API를 사용하여 태그로 작업할 수 있습니다.

AWS Identity and Access Management (IAM)를 사용하는 경우 태그 생성, 편집 또는 삭제 권한이 있는 AWS 계정의 사용자를 제어할 수 있습니다.

리소스 태그 지정 방법

신규 또는 기존 AWS Cloud Map 네임스페이스 및 서비스에 태그를 지정할 수 있습니다.

AWS Cloud Map 콘솔을 사용하는 경우 새 리소스가 생성될 때 태그를 적용하거나 관련 리소스 페이지의 태그 탭을 사용하여 언제든지 기존 리소스에 태그를 적용할 수 있습니다.

AWS Cloud Map API AWS CLI, 또는 AWS SDK를 사용하는 경우 관련 API 작업의 `tags` 파라미터를 사용하여 새 리소스에 태그를 적용하거나 TagResource API 작업을 사용하여 기존 리소스에 태그를 적용할 수 있습니다. 자세한 내용은 TagResource를 참조하세요.

일부 리소스 생성 작업에서는 리소스 생성 시 리소스에 태그를 지정할 수 있습니다. 리소스 생성 중에 태그를 적용할 수 없는 경우 리소스 생성 프로세스는 실패합니다. 이로써 생성 중에 태그를 지정하려는 리소스는 지정된 태그와 함께 생성되거나 전혀 생성되지 않습니다. 생성 시 리소스에 태그를 지정하면 리소스 생성 후 사용자 지정 태그 지정 스크립트를 실행할 필요가 없습니다.

다음 표에서는 태그를 지정할 수 있는 AWS Cloud Map 리소스와 생성 시 태그를 지정할 수 있는 리소스를 설명합니다.

AWS Cloud Map 리소스에 대한 태그 지정 지원

리소스	태그 지원	태그 전달 지원	생성 시 태그 지정 지원(AWS Cloud Map API, AWS CLI, AWS SDK)
AWS Cloud Map 네임스페이스	예	아니요. 네임스페이스 태그는 네임스페이스에 연결된 다른 리소스로 전파되지 않습니다.	예
AWS Cloud Map 서비스	예	아니요. 서비스 태그는 서비스에 연결된 다른 리소스로 전파되지 않습니다.	예

제한 사항

태그에 적용되는 기본 제한은 다음과 같습니다.

- 각 리소스의 최대 태그 수는 50입니다.
- 각 리소스에 대해 각 태그 키는 고유하며 하나의 값만 가질 수 있습니다.
- 최대 키 길이 - UTF-8 형식의 유니코드 문자 128자
- 최대 값 길이 - UTF-8 형식의 유니코드 문자 256자
- 여러 AWS 서비스 및 리소스에서 태깅 스키마를 사용하는 경우 다른 서비스에 허용되는 문자에 대한 제한이 있을 수 있습니다. 일반적으로 허용되는 문자는 UTF-8로 표시할 수 있는 문자, 숫자 및 공백과 특수 문자 + - = . _ : / @입니다.
- 태그 키와 값은 대소문자를 구분합니다.
- `aws:`, `AWS:`또는 키 또는 값에 대한 접두사와 같은의 대문자 또는 소문자 조합을 사용하지 마십시오. AWS 사용을 위해 예약되어 있습니다. 이 접두사가 지정된 태그 키나 값은 편집하거나 삭제할 수 없습니다. 이 접두사가 지정된 태그는 리소스당 태그 수 제한에 포함되지 않습니다.

AWS Cloud Map 리소스에 대한 태그 업데이트

다음 AWS CLI 명령 또는 AWS Cloud Map API 작업을 사용하여 리소스에 대한 태그를 추가, 업데이트, 나열 및 삭제합니다.

AWS Cloud Map 리소스에 대한 태그 지정 지원

Task	API 작업	AWS CLI	AWS Tools for Windows PowerShell
하나 이상의 태그를 추가하거나 덮어씁니다.	TagResource	tag-resource	Add-SDResourceTag
하나 이상의 태그를 삭제합니다.	UntagResource	untag-resource	Remove-SDResourceTag
리소스에 대한 태그 나열	ListTagsForResource	list-tags-for-resource	Get-SDResourceTag

다음 예제는 AWS CLI를 사용하여 리소스에 태그를 지정하거나 태그를 제거하는 방법을 보여줍니다.

예제 1: 기존 리소스에 태그 지정

다음 명령은 기존 리소스에 태그를 지정합니다.

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

예제 2: 기존 리소스에서 태그 제거

다음 명령은 기존 리소스에서 태그를 삭제합니다.

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

예제 3: 리소스의 태그 나열

다음 명령은 기존 리소스와 연결된 태그를 나열합니다.

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

일부 리소스 생성 작업에서는 리소스를 생성할 때 태그를 지정할 수 있습니다. 다음 태스크는 생성 시 태그 지정을 지원합니다.

작업	API 작업	AWS CLI	AWS Tools for Windows PowerShell
HTTP 네임스페이스 생성	CreateHttpNamespace	create-http-namespace	New-SDHttpNamespace
DNS를 기반으로 프라이빗 네임스페이스 생성	CreatePrivateDnsNamespace	create-private-dns-namespace	New-SDPrivateDnsNamespace
DNS를 기반으로 공용 네임스페이스 생성	CreatePublicDnsNamespace	create-public-dns-namespace	New-SDPublicDnsNamespace
서비스 생성	CreateService	create-service	New-SDService

AWS Cloud Map 서비스 할당량

AWS Cloud Map 리소스에는 다음과 같은 계정 수준 서비스 할당량이 적용됩니다. 나열된 각 할당량은 AWS Cloud Map 리소스를 생성하는 각 AWS 리전에 적용됩니다.

명칭	기본값	조정 가능	설명
인스턴스당 사용자 지정 속성	지원되는 각 리전: 30개	아니요	인스턴스 등록 시 지정한 사용자 지정 속성의 최대 개수입니다.
계정 버스트 레이트당 DiscoverInstances 작업	지원되는 각 리전: 2,000	예	단일 계정에서 DiscoverInstances 작업을 호출할 수 있는 최대 버스트 속도입니다.
계정당 DiscoverInstances 작업의 일정한 속도	지원되는 각 리전: 1,000	예	단일 계정에서 DiscoverInstances 작업을 호출할 수 있는 최대 일정한 속도입니다.
계정 속도당 DiscoverInstancesRevision 작업	지원되는 각 리전: 3,000	예	단일 계정에서 DiscoverInstancesRevision 작업을 호출하는 최대 속도입니다.
네임스페이스당 인스턴스	지원되는 각 리전: 2,000	예	동일한 네임스페이스를 사용하여 등록할 수 있는 최대 서비스 인스턴스 수입니다.
서비스당 인스턴스	지원되는 각 리전: 1,000	아니요	동일한 서비스를 사용하여 리전에서 등록할 수 있는 최대 인스턴스 수.

명칭	기본값	조정 가능	설명
리전당 네임스페이스	지원되는 각 지역: 50	예	리전당 생성할 수 있는 최대 네임스페이스 수.

* 사용자가 네임스페이스를 생성하면 Amazon Route 53 호스팅 영역이 자동으로 생성됩니다. 이 호스팅 영역은 AWS 계정으로 생성할 수 있는 호스팅 영역 수의 할당량에 포함됩니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 호스팅 영역에 대한 할당량을 참조하세요.

** AWS Cloud Map 의 DNS 네임스페이스 인스턴스를 늘리려면 호스팅 영역 Route 53 한도당 레코드 수를 늘려야 하며, 이 경우 추가 요금이 발생합니다.

AWS Cloud Map 서비스 할당량 관리

AWS Cloud Map 는 중앙 위치에서 할당량을 보고 관리할 수 있는 AWS 서비스인 Service Quotas와 통합되었습니다. 자세한 내용은 Service Quotas 사용 설명서의 Service Quotas는 무엇입니까?를 참조하세요.

Service Quotas를 사용하면 AWS Cloud Map 서비스 할당량의 값을 쉽게 찾을 수 있습니다.

AWS Management Console

를 사용하여 AWS Cloud Map 서비스 할당량을 보려면 AWS Management Console

1. https://console.aws.amazon.com/servicequotas/에서 Service Quotas 콘솔을 엽니다.
2. 탐색 창에서 AWS 서비스를 선택합니다.
3. AWS 서비스 목록에서 AWS Cloud Map를 검색하여 선택합니다.
4. 서비스 할당량 목록에서 서비스 할당량 이름 AWS Cloud Map, 적용된 값(사용 가능한 경우), AWS 기본 할당량 및 할당량 값을 조정할 수 있는지 여부를 확인할 수 있습니다.

   설명과 같은 서비스 할당량에 대한 추가 정보를 보려면 할당량 이름을 선택하여 할당량 세부 정보를 가져옵니다.
5. (선택 사항) 할당량 증가를 요청하려면 늘릴 할당량을 선택하고 계정 수준에서 증가 요청을 선택합니다.

를 사용하여 서비스 할당량에 대해 자세히 알아보려면 Service Quotas 사용 설명서를 AWS Management Console 참조하세요.

AWS CLI

를 사용하여 AWS Cloud Map 서비스 할당량을 보려면 AWS CLI

다음 명령을 실행하여 기본 할당 AWS Cloud Map 량을 확인합니다.

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

다음 명령을 실행하여 적용된 할당 AWS Cloud Map 량을 확인합니다.

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

를 사용한 서비스 할당량 작업에 대한 자세한 내용은 Service Quotas AWS CLI 명령 AWS CLI참조를 참조하세요. 할당량 증가를 요청하려면 AWS CLI 명령 참조에서 `request-service-quota-increase` 명령을 참조하세요.

Handle AWS Cloud Map DiscoverInstances API 요청 제한

AWS Cloud Map 는 리전별로 각 AWS 계정에 대한 DiscoverInstances API 요청을 제한합니다. 조절은 서비스의 성능을 개선하고 모든 AWS Cloud Map 고객에게 공정한 사용을 제공하는 데 도움이 됩니다. 제한을 사용하면 AWS Cloud Map DiscoverInstances API에 대한 호출이 허용되는 최대 DiscoverInstances API 요청 할당량을 초과하지 않습니다. 다음 소스 중 하나에서 시작된 DiscoverInstances API 호출에는 요청 할당량이 적용됩니다.

- 타사 애플리케이션
- 명령줄 도구
- AWS Cloud Map 콘솔

API 제한 할당량을 초과하면 `RequestLimitExceeded` 오류 코드가 표시됩니다. 자세한 내용은 the section called “요청 속도 제한” 단원을 참조하십시오.

제한 적용 방법

AWS Cloud Map 는 토큰 버킷 알고리즘을 사용하여 API 제한을 구현합니다. 이 알고리즘을 사용하면 계정에 특정 수의 토큰을 보관하는 버킷이 있습니다. 버킷의 토큰 수는 지정된 초당 제한 할당량을 나타냅니다. 단일 리전에는 버킷이 하나 있으며 이는 해당 리전의 모든 엔드포인트에 적용됩니다.

요청 속도 제한

제한은 수행할 수 있는 DiscoverInstances API 요청 수를 제한합니다. 각 요청은 버킷에서 하나의 토큰을 제거합니다. 예를 들어 DiscoverInstances API 작업의 버킷 크기는 토큰 2,000개이므로 매초 최대 2,000개의 DiscoverInstances 요청을 할 수 있습니다. 매초 요청이 2,000개를 초과하면 병목 현상이 발생하고 해당 초 내에 나머지 요청은 실패합니다.

버킷은 설정된 속도로 자동으로 다시 채워집니다. 버킷 용량이 부족하면 버킷 용량에 도달할 때까지 매초마다 정해진 수의 토큰이 다시 추가됩니다. 다시 채우기 토큰이 도착했을 때 버킷 용량이 다 차면 해당 토큰은 폐기됩니다. DiscoverInstances API 작업의 버킷 크기는 토큰 2,000개이고 다시 채우기 속도는 초당 1,000개 토큰입니다. 1초에 2,000건의 DiscoverInstances API 요청을 하면 버킷의 즉시 토큰이 0개로 줄어듭니다. 그러면 최대 용량 2,000개에 도달할 때까지 매초마다 최대 1,000개의 토큰이 버킷에 다시 채워집니다.

버킷에 추가된 토큰은 그대로 사용할 수 있습니다. API를 요청하기 전에 버킷이 최대 용량이 될 때까지 기다릴 필요가 없습니다. 1초에 2,000건의 DiscoverInstances API 요청을 수행하여 버킷을 고갈시킨 경우에도 필요한 기간 동안 계속 매초마다 최대 1,000건의 DiscoverInstances API 요청을 할 수 있습니다. 즉, 다시 채우기 토큰이 버킷에 추가되면 즉시 사용할 수 있습니다. 버킷은 초당 API 요청 횟수가 다시 채우기 속도보다 적은 경우에만 최대 용량까지 다시 채워지기 시작합니다.

재시도 또는 일괄 처리

API 요청이 실패하는 경우 애플리케이션에서 요청을 재시도해야 할 수 있습니다. API 요청 수를 줄이려면 연속적인 요청 사이에 적절한 절전 간격을 사용하세요. 최상의 결과를 얻으려면 절전 시간 간격을 늘리거나 가변적으로 사용합니다.

휴면 간격 계산

API 요청을 폴링하거나 재시도해야 하는 경우 지수 백오프 알고리즘을 사용하여 API 호출 간 절전 시간 간격을 계산하는 것이 좋습니다. 연속적인 오류 응답에 대한 재시도 사이의 대기 시간을 점진적으로 늘리면 실패한 요청 수를 줄일 수 있습니다. 이 알고리즘의 자세한 내용과 구현 예제는 SDK 및 도구 참조 안내서의 재시도 동작을 참조하세요. AWS SDKs

API 제한 할당량 조정

AWS 계정에 대한 API 제한 할당량 증가를 요청할 수 있습니다. 할당량 조정을 요청하려면 AWS Support Center에 문의하세요.

에 대한 문서 기록 AWS Cloud Map

다음 표에서 AWS Cloud Map 개발자 설명서의 중요한 업데이트 및 새 기능이 나와 있습니다. 사용자로부터 받은 의견을 수렴하기 위해 설명서가 자주 업데이트됩니다.

변경 사항	설명	날짜
AWS Cloud Map 교차 계정 네임스페이스 공유	이제 간소화된 교차 계정 서비스 검색 및 레지스트리를 위해 AWS Resource Access Manager (AWS RAM)를 AWS Organizations 사용하여의 다른 AWS 계정 또는 조직 내에서 네임스페이스를 공유할 수 있습니다.	2025년 8월 14일
AWS Cloud Map 서비스 속성	이제 서비스 수준에서 속성을 지정하여 서비스에 등록된 인스턴스 간에 속성이 중복되지 않도록 할 수 있습니다. 이러한 속성을 복잡한 트래픽 라우팅, 제한 시간 및 재시도 값 설정, 서비스와 외부 통합 간의 조정에 사용할 수 있습니다.	2024년 12월 13일
자습서 추가	사용에 대한 일반적인 사용 사례를 보여주는 두 가지 자습서가 AWS Cloud Map 추가되었습니다.	2024년 3월 27일
CloudTrail 통합 설명서 업데이트	API 활동을 로깅하기 위한 CloudTrail과의 AWS Cloud Map 통합을 설명하는 설명서가 업데이트되었습니다.	2024년 3월 20일
관리형 정책 업데이트	`AWSCloudMapDiscoverInstanceAccess` ,	2023년 9월 20일

	`AWSCloudMapRegisterInstanceAccess` , `AWSCloudMapReadOnlyAccess` 정책이 업데이트되었습니다.	
Cloud Map 및 AWS PrivateLink	이제 AWS PrivateLink 를 사용하여 VPC와 간에 프라이빗 연결을 생성할 수 있습니다 AWS Cloud Map.	2023년 9월 15일
관리형 정책 업데이트	`AWSCloudMapDiscoverInstanceAccess` 정책이 업데이트되었습니다.	2023년 8월 15일
AWS Python용 SDK	Python 명령줄 예제가 추가되었습니다.	2022년 9월 13일
IPv6 지원	API 엔드포인트는 이제 `IPv6` 네트워크에서만 사용할 수 있습니다.	2022년 1월 28일
서비스 인스턴스 검색	AWS Cloud Map 는 DiscoverInstances API 작업만 사용하고 DNS 쿼리를 사용하지 않고 검색할 수 있는 DNS 쿼리를 지원하는 네임스페이스에서의 서비스 생성에 대한 지원을 추가했습니다.	2021년 3월 24일
리소스에 태깅	AWS Cloud Map 는를 사용하여 네임스페이스 및 서비스에 메타데이터 태그를 추가하는 지원을 추가했습니다 AWS Management Console.	2021년 2월 8일
리소스에 태깅	AWS Cloud Map 는 AWS CLI 및 APIs.	2020년 6월 22일

최초 릴리스	이 문서는 첫 번째 AWS Cloud Map 개발자 안내서 릴리스입니다.	2018년 11월 28일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.