aws

Entwicklerhandbuch

AWS Cloud Map

AWS Cloud Map: Entwicklerhandbuch

Table of Contents

Was ist AWS Cloud Map?

AWS Cloud Map ist eine vollständig verwaltete Lösung, mit der Sie logische Namen den Backend-Diensten und -Ressourcen zuordnen können, von denen Ihre Anwendungen abhängen. Sie hilft Ihren Anwendungen auch dabei AWS SDKs, Ressourcen mithilfe von RESTful API-Aufrufen oder DNS-Abfragen zu erkennen. AWS Cloud Map bedient nur fehlerfreie Ressourcen, bei denen es sich um Amazon DynamoDB-Tabellen (DynamoDB), Amazon Simple Queue Service (Amazon SQS) -Warteschlangen, beliebige Anwendungsservices auf höherer Ebene, die mit Amazon Elastic Compute Cloud (Amazon EC2) -Instances oder Amazon Elastic Container Service (Amazon ECS) -Aufgaben erstellt wurden, und mehr handeln kann.

Komponenten von AWS Cloud Map

Namespace

Zu Beginn erstellen Sie zunächst einen AWS Cloud Map Namespace, mit dem Dienste für eine Anwendung gruppiert werden können. Ein Namespace identifiziert den Namen, den Sie zum Auffinden Ihrer Ressourcen verwenden möchten, und gibt auch an, wie Sie Ressourcen suchen möchten: mithilfe von AWS Cloud Map DiscoverInstancesAPI-Aufrufen, DNS-Abfragen in einer VPC oder öffentlichen DNS-Abfragen. In den meisten Fällen enthält ein Namespace alle Dienste für eine Anwendung, z. B. eine Abrechnungsanwendung. Weitere Informationen finden Sie unter AWS Cloud Map Namespaces.

Service

Nachdem Sie einen Namespace erstellt haben, erstellen Sie einen AWS Cloud Map Dienst für jeden Ressourcentyp, den Sie zum Auffinden von Endpunkten verwenden AWS Cloud Map möchten. Sie können z. B. Services für Webserver und Datenbankserver erstellen.

Ein Dienst ist eine Vorlage, die AWS Cloud Map verwendet wird, wenn Ihre Anwendung eine weitere Ressource, z. B. einen weiteren Webserver, hinzufügt. Wenn Sie beim Erstellen des Namespace angegeben haben, dass Ressourcen per DNS gesucht werden sollen, enthält ein Service Informationen zu den Arten von Datensätzen, die Sie zum Suchen des Webservers verwenden möchten. Ein Service gibt auch an, ob Sie den Zustand der Ressource überprüfen möchten und ob Sie Amazon Route 53 Health Checks oder einen Health Checker eines Drittanbieters verwenden möchten. Weitere Informationen finden Sie unter AWS Cloud Map Dienstleistungen.

Service-Instance

Wenn Ihre Anwendung eine Ressource hinzufügt, können Sie die AWS Cloud Map RegisterInstanceAPI-Aktion im Code aufrufen, wodurch eine AWS Cloud Map Dienstinstanz in einem Service erstellt wird. Die Dienstinstanz enthält Informationen darüber, wie Ihre Anwendung die Ressource finden kann, unabhängig davon, ob sie DNS oder die AWS Cloud Map DiscoverInstancesAPI-Aktion verwendet.

Wenn Ihre Anwendung eine Verbindung zu einer Ressource herstellen muss, ruft sie öffentliche DiscoverInstancesoder private DNS-Abfragen auf oder verwendet sie, indem sie den Namespace und den Dienst angibt, die der Ressource zugeordnet sind. AWS Cloud Map gibt Informationen darüber zurück, wie eine oder mehrere Ressourcen gefunden werden können. Wenn Sie bei der Erstellung des Dienstes eine Integritätsprüfung angegeben haben, werden nur fehlerfreie Instanzen AWS Cloud Map zurückgegeben. Weitere Informationen finden Sie unter AWS Cloud Map Dienstinstanzen.

Zugreifen AWS Cloud Map

Sie können AWS Cloud Map auf folgende Arten zugreifen:

- AWS Management Console— Die Verfahren in diesem Handbuch erläutern, wie Sie mit AWS Management Console dem Aufgaben ausführen können.
- AWS SDKs— Wenn Sie eine Programmiersprache verwenden, die ein SDK für AWS bereitstellt, können Sie ein SDK für den Zugriff verwenden AWS Cloud Map. SDKs Vereinfachen Sie die Authentifizierung, lassen Sie sich problemlos in Ihre Entwicklungsumgebung integrieren und bieten Sie Zugriff auf AWS Cloud Map Befehle. Weitere Informationen finden Sie unter Tools für Amazon Web Services.
- AWS Command Line Interface— Weitere Informationen finden Sie unter Erste Schritte mit dem AWS CLI im AWS Command Line Interface Benutzerhandbuch.
- AWS Tools for Windows PowerShell— Weitere Informationen finden Sie unter Erste Schritte mit dem AWS Tools for Windows PowerShell im AWS -Tools für PowerShell Benutzerhandbuch.
- AWS Cloud Map API — Wenn Sie eine Programmiersprache verwenden, für die kein SDK verfügbar ist, finden Sie in der AWS Cloud Map API-Referenz Informationen zu API-Aktionen und zum Stellen von API-Anfragen.

Note

IPv6 Kundensupport — AWS Cloud Map Ab dem 22. Juni 2023 werden in allen neuen Regionen alle Befehle, an die `IPv6` Clients gesendet werden, an einen neuen Dual-Stack-Endpunkt () weitergeleitet. `servicediscovery.<region>.api.aws` AWS Cloud Map `IPv6`In den folgenden Regionen, die vor dem 22. Juni 2023 veröffentlicht wurden, sind nur Netzwerke sowohl für Legacy - (`servicediscovery.<region>.amazonaws.com`) als auch für Dual-Stack-Endgeräte erreichbar:

- USA Ost (Ohio) – us-east-2
- USA Ost (Nord-Virginia) – us-east-1
- USA West (Nordkalifornien) – us-west-1
- USA West (Oregon) – us-west-2
- Afrika (Kapstadt) – af-south-1
- Asien-Pazifik (Hongkong) – ap-east-1
- Asien-Pazifik (Hyderabad) — ap-south-2
- Asien-Pazifik (Jakarta) – ap-southeast-3
- Asien-Pazifik (Melbourne) — ap-southeast-4
- Asien-Pazifik (Mumbai) – ap-south-1
- Asien-Pazifik (Osaka) – ap-northeast-3
- Asien-Pazifik (Seoul) – ap-northeast-2
- Asien-Pazifik (Singapur) – ap-southeast-1
- Asien-Pazifik (Sydney) – ap-southeast-2
- Asien-Pazifik (Tokio) – ap-northeast-1
- Kanada (Zentral) – ca-central-1
- Europa (Frankfurt) – eu-central-1
- Europa (Irland) – eu-west-1
- Europa (London) – eu-west-2
- Europa (Mailand) – eu-south-1
- Europa (Paris) – eu-west-3
- Europa (Spanien) — eu-south-2
- Europa (Stockholm) – eu-north-1

- Europa (Zürich) — eu-central-2
- Naher Osten (Bahrain) – me-south-1
- Naher Osten (VAE) — me-central-1
- Südamerika (São Paulo) – sa-east-1
- AWS GovCloud (US-Ost) — -1 us-gov-east
- AWS GovCloud (US-West) — -1 us-gov-west

AWS Identity and Access Management

AWS Cloud Map ist in AWS Identity and Access Management (IAM) integriert, einen Dienst, den Ihre Organisation für die folgenden Aktionen verwenden kann:

- Erstellen Sie Benutzer und Gruppen unter dem Konto Ihrer Organisation AWS
- Teilen Sie Ihre AWS Kontoressourcen auf effiziente Weise unter den Benutzern im Konto
- Zuweisen eindeutiger Sicherheitsanmeldeinformationen zu jedem Benutzer
- Genaue Kontrolle des Zugriffs jedes Benutzers auf Dienste und Ressourcen

Sie können IAM with beispielsweise verwenden, AWS Cloud Map um zu kontrollieren, welche Benutzer in Ihrem AWS Konto einen neuen Namespace erstellen oder Instanzen registrieren können.

Allgemeine Informationen zu IAM finden Sie in den folgenden Ressourcen:

- Identity and Access Management für AWS Cloud Map
- AWS Identity and Access Management
- IAM Benutzerhandbuch

AWS Cloud Map Preisgestaltung

AWS Cloud Map Die Preisgestaltung basiert auf Ressourcen, die Sie in der Service-Registry registrieren, und auf API-Aufrufen, die Sie tätigen, um diese zu ermitteln. AWS Cloud Map Es fallen keine Vorauszahlungen an und Sie zahlen nur für das, was Sie tatsächlich nutzen.

Optional können Sie auch eine DNS-basierte Erkennung für Ressourcen mit IP-Adressen aktivieren. Sie können mithilfe von Amazon Route 53-Zustandsprüfungen auch die Zustandsprüfung für Ihre

Ressourcen aktivieren, unabhängig davon, ob Sie Instances mithilfe von API-Aufrufen oder DNS-Abfragen entdecken. Für die Nutzung von Route 53-DNS und Health Checks fallen zusätzliche Gebühren an.

Weitere Informationen finden Sie unter AWS Cloud Map – Preise.

AWS Cloud Map und AWS Cloud-Compliance

Informationen zur AWS Cloud Map Einhaltung verschiedener Sicherheitsvorschriften und Prüfungsstandards finden Sie auf den folgenden Seiten:

- AWS Cloud-Konformität
- AWS Dienstleistungen im Geltungsbereich des Compliance-Programms

Erste Schritte mit AWS Cloud Map

In den folgenden Anleitungen erfahren Sie, wie Sie die Verwendung von AWS Cloud Map Namespaces einrichten AWS Cloud Map und allgemeine Aufgaben ausführen.

Überblick über den Leitfaden	Weitere Informationen
Melden Sie sich für die Nutzung an AWS und bereiten Sie sich auf die Nutzung vor AWS Cloud Map	Zur Verwendung eingerichtet AWS Cloud Map
Verwenden von DNS-Abfragen und API-Aufrufen zur Erkennung von Back-End-Diensten.	Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit DNS-Abfragen und API-Aufrufen verwenden
Verwenden von DNS-Abfragen und API-Aufrufen zur Erkennung von Backend-Diensten mithilfe von. AWS CLI	Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit DNS-Abfragen und API-Aufrufen verwenden können, indem Sie AWS CLI
Erstellen einer Beispielanwendung und Verwenden von benutzerdefinierten Attributen im Code zur Erkennung von Ressourcen.	Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden
Erstellen einer Beispielanwendung und Verwenden von benutzerdefinierten Attributen im Code zur Erkennung von Ressourcen mithilfe von AWS CLI.	Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden können, indem Sie AWS CLI

Zur Verwendung eingerichtet AWS Cloud Map

Die Übersicht und die Verfahren in den folgenden Abschnitten sollen Ihnen den Einstieg erleichtern AWS und Sie darauf vorbereiten AWS Cloud Map.

Themen

- Melden Sie sich an für AWS
- Greifen Sie auf die API, AWS CLIAWS Tools for Windows PowerShell, oder die zu AWS SDKs

- Richten Sie das Oder AWS Command Line Interface ein AWS Tools for Windows PowerShell
- Laden Sie ein AWS SDK herunter

Melden Sie sich an für AWS

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Greifen Sie auf die API, AWS CLIAWS Tools for Windows PowerShell, oder die zu AWS SDKs

Um die API, die AWS CLI AWS Tools for Windows PowerShell, oder die verwenden zu können AWS SDKs, müssen Sie Zugriffsschlüssel erstellen. Diese Zugriffsschlüssel bestehen aus einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel. Diese werden zum Signieren der von Ihnen ausgeführten programmgesteuerten Anforderungen an AWS verwendet.

Benutzer benötigen programmgesteuerten Zugriff, wenn sie mit AWS außerhalb des AWS Management Console interagieren möchten. Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift. AWS

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zur Verwendung AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch AWS

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		SDKs und im Tools-Ref erenzhandbuch.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Folgen Sie den Anweisungen unter Verwenden temporäre r Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisung en für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM-Benutzeranmeld einformationen im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldein formationen im Referenzh andbuch AWS SDKs und im Tools-Referenzhandbuch. • Weitere Informationen finden Sie unter Verwaltun g von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benut zerhandbuch. AWS APIs

Richten Sie das Oder AWS Command Line Interface ein AWS Tools for Windows PowerShell

Das AWS Command Line Interface (AWS CLI) ist ein einheitliches Tool zur Verwaltung von AWS Diensten. Informationen zur Installation und Konfiguration von finden Sie AWS CLI im AWS Command Line Interface Benutzerhandbuch unter Installation oder Aktualisierung auf die neueste Version von. AWS CLI

Wenn Sie Erfahrung mit Windows haben PowerShell, bevorzugen Sie möglicherweise die Verwendung von AWS Tools for Windows PowerShell. Weitere Informationen finden Sie unter Einrichten von AWS Tools for Windows PowerShell im AWS -Tools für PowerShell - Benutzerhandbuch.

Laden Sie ein AWS SDK herunter

Wenn Sie eine Programmiersprache verwenden, die ein SDK für AWS bereitstellt, empfehlen wir Ihnen, anstelle der AWS Cloud Map API ein SDK zu verwenden. Die Verwendung eines SDK hat mehrere Vorteile. SDKs vereinfachen die Authentifizierung, lassen sich problemlos in Ihre Entwicklungsumgebung integrieren und bieten Zugriff auf AWS Cloud Map Befehle. Weitere Informationen finden Sie unter Tools für Amazon Web Services.

Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit DNS-Abfragen und API-Aufrufen verwenden

Das folgende Tutorial simuliert eine Microservice-Architektur mit zwei Backend-Diensten. Der erste Dienst wird mithilfe einer DNS-Abfrage auffindbar sein. Der zweite Dienst wird nur über die AWS Cloud Map API auffindbar sein.

> **Note**
>
> Die Ressourcendetails, wie Domainnamen und IP-Adressen, dienen nur zu Simulationszwecken. Sie können nicht über das Internet gelöst werden.

Eine end-to-end AWS CLI Version dieses Tutorials finden Sie unterErfahren Sie, wie Sie AWS Cloud Map Service Discovery mit DNS-Abfragen und API-Aufrufen verwenden können, indem Sie AWS CLI.

Voraussetzungen

Die folgenden Voraussetzungen müssen erfüllt sein, um das Tutorial erfolgreich abschließen zu können.

- Bevor Sie beginnen, führen Sie die Schritte in Zur Verwendung eingerichtet AWS Cloud Map aus.
- Wenn Sie das noch nicht installiert haben AWS Command Line Interface, folgen Sie den Schritten unter Installieren oder Aktualisieren der neuesten Version von AWS CLI, um es zu installieren.

  Das Tutorial erfordert zum Ausführen von Befehlen ein Befehlszeilenterminal oder eine Shell. Verwenden Sie unter Linux und macOS Ihre bevorzugte Shell und Ihren bevorzugten Paketmanager.

  > **Note**
  >
  > In Windows werden einige Bash-CLI-Befehle, die Sie häufig mit Lambda verwenden (z. B. `zip`), von den integrierten Terminals des Betriebssystems nicht unterstützt. Um eine in Windows integrierte Version von Ubuntu und Bash zu erhalten, installieren Sie das Windows-Subsystem für Linux.

- Für das Tutorial ist eine lokale Umgebung mit dem Befehl `dig` DNS Lookup Utility erforderlich.

Schritt 1: Erstellen Sie einen AWS Cloud Map Namespace

In diesem Schritt erstellen Sie einen öffentlichen AWS Cloud Map Namespace. AWS Cloud Map erstellt in Ihrem Namen eine Route 53-Hosting-Zone mit demselben Namen. Auf diese Weise können Sie die in diesem Namespace erstellten Dienstinstanzen entweder mithilfe von öffentlichen DNS-Einträgen oder mithilfe von AWS Cloud Map API-Aufrufen ermitteln.

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie Create namespace (Namespace erstellen) aus.
3. Geben `cloudmap-tutorial.com` Sie als Namespace-Name an.

   > **Note**
   >
   > Wenn Sie dies in der Produktion verwenden möchten, sollten Sie sicherstellen, dass Sie den Namen einer Domain angegeben haben, die Sie besitzen oder auf die Sie Zugriff

hatten. Für die Zwecke dieses Tutorials ist es jedoch nicht erforderlich, dass es sich um eine tatsächliche Domain handelt, die verwendet wird.

4. (Optional) Geben Sie unter Namespace-Beschreibung eine Beschreibung dafür an, wofür Sie den Namespace verwenden möchten.
5. Wählen Sie für die Instanzerkennung API-Aufrufe und öffentliche DNS-Abfragen aus.
6. Behalten Sie die restlichen Standardwerte bei und wählen Sie Create Namespace.

Schritt 2: Erstellen Sie die Dienste AWS Cloud Map

In diesem Schritt erstellen Sie zwei Dienste. Der erste Dienst wird über öffentliche DNS- und API-Aufrufe auffindbar sein. Der zweite Dienst wird nur über API-Aufrufe auffindbar sein.

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie im linken Navigationsbereich Namespaces aus, um die Namespaces aufzulisten, die Sie erstellt haben.
3. Wählen Sie aus der Liste der Namespaces den Namespace aus und klicken Sie auf Details anzeigen. **`cloudmap-tutorial.com`**
4. Wählen Sie im Abschnitt Dienste die Option Dienst erstellen aus und gehen Sie wie folgt vor, um den ersten Dienst zu erstellen.

   a. Geben Sie unter Servicename `public-service` ein. Der Dienstname wird auf die DNS-Einträge angewendet, die AWS Cloud Map erstellt werden. Das verwendete Format ist *`<service-name>`*`.`*`<namespace-name>`*.

   b. Wählen Sie für Service Discovery-Konfiguration die Optionen API und DNS aus.

   c. Wählen Sie im Abschnitt DNS-Konfiguration für Routing-Richtlinie die Option Mehrwertiges Antwort-Routing aus.

   > **Note**
   >
   > Die Konsole übersetzt dies nach der Auswahl in MULTIVALUE. Weitere Informationen zu den verfügbaren Routing-Optionen finden Sie unter Auswahl einer Routing-Richtlinie im Route 53-Entwicklerhandbuch.

   d. Behalten Sie die restlichen Standardwerte bei und wählen Sie Dienst erstellen aus, um zur Seite mit den Namespace-Details zurückzukehren.

5. Wählen Sie im Abschnitt Dienste die Option Dienst erstellen aus und gehen Sie wie folgt vor, um den zweiten Dienst zu erstellen.

   a. Geben Sie unter Servicename `backend-service` ein.

   b. Wählen Sie für Service Discovery-Konfiguration die Option Nur API aus.

   c. Behalten Sie die restlichen Standardwerte bei und wählen Sie Service erstellen aus.

Schritt 3: Registrieren Sie die AWS Cloud Map Dienstinstanzen

In diesem Schritt erstellen Sie zwei Dienstinstanzen, eine für jeden Dienst in unserem Namespace.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie aus der Liste der Namespaces den Namespace aus, den Sie in Schritt 1 erstellt haben, und wählen Sie Details anzeigen aus.
3. Wählen Sie auf der Seite mit den Namespace-Details aus der Liste der Dienste den **`public-service`** Dienst aus und klicken Sie auf Details anzeigen.
4. Wählen Sie im Abschnitt Dienstinstanzen die Option Dienstinstanz registrieren aus und gehen Sie wie folgt vor, um die erste Dienstinstanz zu erstellen.

   a. Geben Sie als Dienstinstanz-ID Folgendes an`first`.

   b. Geben Sie als IPv4 Adresse an`192.168.2.1`.

   c. Behalten Sie die restlichen Standardwerte bei und wählen Sie Dienstinstanz registrieren aus.

5. Wählen Sie mithilfe des Breadcrumbs oben auf der Seite cloudmap-tutorial.com aus, um zur Namespace-Detailseite zurückzukehren.
6. Wählen Sie auf der Seite mit den Namespace-Details aus der Liste der Dienste den Backend-Service aus und klicken Sie auf Details anzeigen.
7. Wählen Sie im Abschnitt Dienstinstanzen die Option Dienstinstanz registrieren aus und gehen Sie wie folgt vor, um die zweite Dienstinstanz zu erstellen.

   a. Geben Sie unter Dienstinstanz-ID `second` an, dass dies die zweite Dienstinstanz ist.

b. Wählen Sie als Instanztyp die Option Identifizierungsinformationen für eine andere Ressource aus.

c. Fügen Sie für benutzerdefinierte Attribute ein Schlüssel-Wert-Paar mit `service-name` als Schlüssel und `backend` als Wert hinzu.

d. Wählen Sie Register service instance (Service-Instance registrieren) aus.

Schritt 4: Entdecken Sie die Dienstinstanzen AWS Cloud Map

Nachdem der AWS Cloud Map Namespace, die Dienste und die Dienstinstanzen erstellt wurden, können Sie überprüfen, ob alles funktioniert, indem Sie die Instanzen ermitteln. Verwenden Sie den `dig` Befehl, um die öffentlichen DNS-Einstellungen zu überprüfen, und die AWS Cloud Map API, um den Back-End-Dienst zu verifizieren. Weitere Informationen zu diesem `dig` Befehl finden Sie unter dig — DNS-Suchprogramm.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.
2. Wählen Sie in der linken Navigation Hosted Zones (Gehostete Zonen).
3. Wählen Sie die gehostete Zone cloudmap-tutorial.com aus. Dadurch werden die Details der gehosteten Zone in einem separaten Bereich angezeigt. Notieren Sie sich die Nameserver, die mit Ihrer Hosting-Zone verknüpft sind, da wir diese im nächsten Schritt verwenden werden.
4. Fragen Sie mit dem Befehl dig und einem der Route 53-Nameserver für Ihre gehostete Zone die DNS-Einträge für Ihre Service-Instanz ab.

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

   `ANSWER SECTION`In der Ausgabe sollte die IPv4 Adresse angezeigt werden, die Sie mit Ihrem `public-service` Service verknüpft haben.

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

5. Fragen Sie mithilfe von die AWS CLI Attribute für Ihre zweiten Dienstinstanzen ab.

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

In der Ausgabe werden die Attribute, die Sie dem Service zugeordnet haben, als Schlüssel-Wert-Paare angezeigt.

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

Schritt 5: Bereinigen Sie die Ressourcen

Sobald Sie das Tutorial abgeschlossen haben, können Sie die Ressourcen löschen. AWS Cloud Map erfordert, dass Sie sie in umgekehrter Reihenfolge bereinigen, zuerst die Dienstinstanzen, dann die Dienste und schließlich den Namespace. AWS Cloud Map bereinigt die Route 53-Ressourcen in Ihrem Namen, wenn Sie diese Schritte ausführen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie aus der Liste der Namespaces den **`cloudmap-tutorial.com`** Namespace aus und klicken Sie auf Details anzeigen.
3. Wählen Sie auf der Seite mit den Namespace-Details aus der Liste der Dienste den **`public-service`** Dienst aus und klicken Sie auf Details anzeigen.
4. Wählen Sie im Abschnitt Dienstinstanzen die `first` Instanz aus und klicken Sie auf Abmelden.
5. Wählen Sie mithilfe des Breadcrumbs oben auf der Seite cloudmap-tutorial.com aus, um zur Namespace-Detailseite zurückzukehren.
6. Wählen Sie auf der Namespace-Detailseite aus der Liste der Dienste den öffentlichen Dienst aus und klicken Sie auf Löschen.
7. Wiederholen Sie die Schritte 3-6 für die. `backend-service`

8. Wählen Sie in der linken Navigationsleiste Namespaces aus.
9. Wählen Sie den **`cloudmap-tutorial.com`** Namespace aus und wählen Sie Löschen.

> **Note**
>
> Obwohl die Route 53-Ressourcen in Ihrem Namen AWS Cloud Map bereinigt werden, können Sie zur Route 53-Konsole navigieren, um zu überprüfen, ob die `cloudmap-tutorial.com` gehostete Zone gelöscht wurde.

Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit DNS-Abfragen und API-Aufrufen verwenden können, indem Sie AWS CLI

Dieses Tutorial zeigt, wie Sie AWS Cloud Map Service Discovery mit der AWS Command Line Interface (CLI) verwenden. Sie erstellen eine Microservice-Architektur mit zwei Back-End-Diensten — einem, der mithilfe von DNS-Abfragen auffindbar ist, und einem anderen, der nur über die API auffindbar ist. AWS Cloud Map

Ein Tutorial, das Schritte für die AWS Cloud Map Konsole beinhaltet, finden Sie unter. Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit DNS-Abfragen und API-Aufrufen verwenden

Voraussetzungen

Die folgenden Voraussetzungen müssen erfüllt sein, um das Tutorial erfolgreich abschließen zu können.

- Bevor Sie beginnen, führen Sie die Schritte in Zur Verwendung eingerichtet AWS Cloud Map aus.
- Wenn Sie das noch nicht installiert haben AWS Command Line Interface, folgen Sie den Schritten unter Installieren oder Aktualisieren der neuesten Version von AWS CLI, um es zu installieren.

  Das Tutorial erfordert zum Ausführen von Befehlen ein Befehlszeilenterminal oder eine Shell. Verwenden Sie unter Linux und macOS Ihre bevorzugte Shell und Ihren bevorzugten Paketmanager.

> **Note**
>
> In Windows werden einige Bash-CLI-Befehle, die Sie häufig mit Lambda verwenden (z. B. `zip`), von den integrierten Terminals des Betriebssystems nicht unterstützt. Um eine in Windows integrierte Version von Ubuntu und Bash zu erhalten, installieren Sie das Windows-Subsystem für Linux.

- Für das Tutorial ist eine lokale Umgebung mit dem Befehl `dig` DNS Lookup Utility erforderlich.

Erstellen Sie einen AWS Cloud Map Namespace

Zunächst erstellen Sie einen öffentlichen AWS Cloud Map Namespace. AWS Cloud Map erstellt eine Route 53-Hosting-Zone mit demselben Namen, wodurch die Diensterkennung sowohl über DNS-Einträge als auch über API-Aufrufe ermöglicht wird.

1. Erstellen Sie den öffentlichen DNS-Namespace:

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

   Der Befehl gibt eine Vorgangs-ID zurück, mit der Sie den Status der Namespace-Erstellung überprüfen können:

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

2. Überprüfen Sie den Vorgangsstatus, um zu bestätigen, dass der Namespace erfolgreich erstellt wurde:

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

3. Sobald der Vorgang erfolgreich ist, rufen Sie die Namespace-ID ab:

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

Dieser Befehl gibt die Namespace-ID zurück, die Sie für nachfolgende Schritte benötigen:

```
ns-abcd1234xmplefgh
```

Erstellen Sie die Dienste AWS Cloud Map

Erstellen Sie jetzt zwei Dienste in Ihrem Namespace. Der erste Dienst wird sowohl über DNS- als auch über API-Aufrufe auffindbar sein, während der zweite nur über API-Aufrufe auffindbar sein wird.

1. Erstellen Sie den ersten Dienst mit aktivierter DNS-Erkennung:

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

Der Befehl gibt Details zum erstellten Dienst zurück:

```
{
    "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. Erstellen Sie den zweiten Dienst mit reiner API-Erkennung:

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

Der Befehl gibt Details zum erstellten Dienst zurück:

```
{
    "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"
    }
}
```

Registrieren Sie die AWS Cloud Map Dienstinstanzen

Als Nächstes registrieren Sie Dienstinstanzen für jeden Ihrer Dienste. Diese Instanzen stellen die tatsächlichen Ressourcen dar, die entdeckt werden.

1. Registrieren Sie die erste Instanz mit einer IPv4 Adresse für die DNS-Erkennung:

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

```
    --region us-east-2
```

Der Befehl gibt eine Vorgangs-ID zurück:

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

2. Überprüfen Sie den Betriebsstatus, um zu bestätigen, dass die Instanz erfolgreich registriert wurde:

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

3. Registrieren Sie die zweite Instanz mit benutzerdefinierten Attributen für die API-Erkennung:

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

Der Befehl gibt eine Vorgangs-ID zurück:

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

4. Überprüfen Sie den Betriebsstatus, um zu bestätigen, dass die Instanz erfolgreich registriert wurde:

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

Entdecken Sie die AWS Cloud Map Dienstinstanzen

Nachdem Sie Ihre Dienstinstanzen erstellt und registriert haben, können Sie überprüfen, ob alles funktioniert, indem Sie sie sowohl mithilfe von DNS-Abfragen als auch mithilfe der AWS Cloud Map API ermitteln.

1. Rufen Sie zunächst die ID der gehosteten Route 53-Zone ab:

   ```
   aws route53 list-hosted-zones-by-name \
       --dns-name cloudmap-tutorial.com \
       --query "HostedZones[0].Id" \
       --output text
   ```

   Dies gibt die ID der gehosteten Zone zurück:

   ```
   /hostedzone/Z1234ABCDXMPLEFGH
   ```

2. Rufen Sie die Nameserver für Ihre gehostete Zone ab:

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

   Dies gibt einen der Nameserver zurück:

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

3. Verwenden Sie den `dig` Befehl, um die DNS-Einträge für Ihren öffentlichen Dienst abzufragen:

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

   In der Ausgabe sollte die IPv4 Adresse angezeigt werden, die Sie mit Ihrem Dienst verknüpft haben:

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

4. Verwenden Sie den AWS CLI , um die Back-End-Dienstinstanz zu ermitteln:

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

In der Ausgabe werden die Attribute angezeigt, die Sie dem Service zugeordnet haben:

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

Bereinigen Sie die Ressourcen

Wenn Sie das Tutorial abgeschlossen haben, bereinigen Sie die Ressourcen, um Gebühren zu vermeiden. AWS Cloud Map erfordert, dass Sie sie in umgekehrter Reihenfolge bereinigen: zuerst die Dienstinstanzen, dann die Dienste und schließlich den Namespace.

1. Die erste Dienstinstanz deregistrieren:

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

2. Die zweite Dienstinstanz deregistrieren:

```
aws servicediscovery deregister-instance \
    --service-id srv-ijkl5678xmplmnop \
```

```
    --instance-id second \
    --region us-east-2
```

3. Löschen Sie den öffentlichen Dienst:

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

4. Löschen Sie den Backend-Dienst:

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

5. Löschen Sie den -Namespace:

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

6. Stellen Sie sicher, dass die von Route 53 gehostete Zone gelöscht wurde:

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

Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden

Das folgende Tutorial zeigt, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden können, die über die AWS Cloud Map API auffindbar sind. Das Tutorial führt Sie durch das Erstellen und Ausführen von Client-Anwendungen mithilfe von AWS CloudShell. Die Anwendungen verwenden zwei Lambda-Funktionen, um Daten in eine DynamoDB-Tabelle zu schreiben und dann aus der Tabelle zu lesen. Die Lambda-Funktionen und die DynamoDB-Tabelle sind AWS Cloud Map als Dienstinstanzen registriert. Der Code in den Client-Anwendungen und Lambda-Funktionen verwendet AWS Cloud Map benutzerdefinierte Attribute, um die Ressourcen zu ermitteln, die für die Ausführung des Jobs benötigt werden.

Eine AWS CLI basierte Version dieses Tutorials finden Sie unterErfahren Sie, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden können, indem Sie AWS CLI.

> **Important**
>
> Während des Workshops werden Sie AWS Ressourcen erstellen, für die Kosten auf Ihrem AWS Konto anfallen. Es wird empfohlen, die Ressourcen zu bereinigen, sobald Sie den Workshop beendet haben, um die Kosten zu minimieren.

Voraussetzungen

Bevor Sie beginnen, führen Sie die Schritte in Zur Verwendung eingerichtet AWS Cloud Map aus.

Schritt 1: Erstellen Sie einen Namespace AWS Cloud Map

In diesem Schritt erstellen Sie einen AWS Cloud Map Namespace. Ein Namespace ist ein Konstrukt, das verwendet wird, um Dienste für eine Anwendung zu gruppieren. Wenn Sie den Namespace erstellen, geben Sie an, wie die Ressourcen auffindbar sein sollen. Die Ressourcen, die in dem in diesem Schritt erstellten Namespace erstellt wurden, können mit AWS Cloud Map API-Aufrufen unter Verwendung benutzerdefinierter Attribute gefunden werden.

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter. https://console.aws.amazon.com/cloudmap/
2. Wählen Sie Create namespace (Namespace erstellen) aus.
3. Geben `cloudmap-tutorial` Sie als Namespace-Name an.
4. (Optional) Geben Sie unter Namespace-Beschreibung eine Beschreibung dafür an, wofür Sie den Namespace verwenden möchten.
5. Wählen Sie für Instance Discovery die Option API-Aufrufe aus.
6. Behalten Sie die restlichen Standardwerte bei und wählen Sie Create Namespace.

Schritt 2: Erstellen Sie eine DynamoDB-Tabelle

In diesem Schritt erstellen Sie eine DynamoDB-Tabelle. Die Tabelle wird zum Speichern und Abrufen von Daten für die Beispielanwendung verwendet, die Sie in den folgenden Schritten erstellen werden.

Informationen zum Erstellen einer DynamoDB finden Sie unter Schritt 1: Erstellen einer Tabelle in DynamoDB im DynamoDB Developer Guide. Ermitteln Sie anhand der folgenden Tabelle, welche Optionen angegeben werden müssen.

Option	Wert	
Tabellenname	Cloudmap	
Partitionsschlüssel	id	

Behalten Sie die Standardwerte für die restlichen Einstellungen bei und erstellen Sie die Tabelle.

Schritt 3: Erstellen Sie einen AWS Cloud Map Datendienst und registrieren Sie die DynamoDB-Tabelle als Instanz

In diesem Schritt erstellen Sie einen AWS Cloud Map Service und registrieren dann die im letzten Schritt erstellte DynamoDB-Tabelle als Dienstinstanz.

1. Öffnen Sie die Konsole unter AWS Cloud Map https://console.aws.amazon.com/cloudmap/
2. Wählen Sie aus der Liste der Namespaces den **`cloudmap-tutorial`** Namespace aus und klicken Sie auf Details anzeigen.
3. Wählen Sie im Abschnitt Dienste die Option Dienst erstellen aus und gehen Sie wie folgt vor.
   a. Geben Sie unter Servicename `data-service` ein.
   b. Behalten Sie die restlichen Standardwerte bei und wählen Sie Dienst erstellen aus.
4. Wählen Sie im Abschnitt Dienste den `data-service` Dienst aus und klicken Sie auf Details anzeigen.
5. Wählen Sie im Abschnitt Dienstinstanzen die Option Dienstinstanz registrieren aus.
6. Gehen Sie auf der Seite Dienstinstanz registrieren wie folgt vor.
   a. Wählen Sie als Instanztyp die Option Identifizierungsinformationen für eine andere Ressource aus.
   b. Geben Sie für Service-Instanz-ID Folgendes and`data-instance`.
   c. Geben Sie im Abschnitt Benutzerdefinierte Attribute das folgende Schlüssel-Wert-Paar an: key =`tablename`, value =. `cloudmap`

Schritt 4: Erstellen Sie eine Ausführungsrolle AWS Lambda

In diesem Schritt erstellen Sie eine IAM-Rolle, die von der AWS Lambda Funktion im nächsten Schritt verwendet wird. Sie können die IAM-Rolle benennen `cloudmap-tutorial-role` und die Berechtigungsgrenze weglassen, da die Rolle nur für dieses Tutorial verwendet wird und Sie sie anschließend löschen können.

So erstellen Sie die Servicerolle für Lambda (IAM-Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen, und wählen Sie dann Rolle erstellen.
3. Wählen Sie für Vertrauenswürdige Entität die Option AWS-Service aus.
4. Wählen Sie für Service oder Anwendungsfall Lambda und dann den Lambda-Anwendungsfall aus.
5. Wählen Sie Weiter aus.
6. Suchen Sie nach der Richtlinie, wählen Sie das Kästchen neben der **`PowerUserAccess`** Richtlinie aus und wählen Sie dann Weiter aus.
7. Wählen Sie Weiter aus.
8. Geben Sie als Rollenname an`cloudmap-tutorial-role`.
9. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).

Schritt 5: Erstellen Sie die Lambda-Funktion zum Schreiben von Daten

In diesem Schritt erstellen Sie eine von Grund auf neu erstellte Lambda-Funktion, die Daten in die DynamoDB-Tabelle schreibt, indem Sie die AWS Cloud Map API verwenden, um den von Ihnen erstellten Service abzufragen. AWS Cloud Map

Informationen zum Erstellen einer Lambda-Funktion finden Sie unter Erstellen einer Lambda-Funktion mit der Konsole im AWS Lambda Entwicklerhandbuch. Ermitteln Sie anhand der folgenden Tabelle, welche Optionen angegeben oder ausgewählt werden müssen.

Option	Wert	
Funktionsname	Funktion schreiben	

Option	Wert
Laufzeit	Python 3.12
Architektur	x86_64
Berechtigungen	Verwenden Sie eine bestehende Rolle
Vorhandene Rolle	cloudmap-tutorial-role

Nachdem Sie die Funktion erstellt haben, aktualisieren Sie den Beispielcode, sodass er den folgenden Python-Code wiedergibt, und stellen Sie dann die Funktion bereit. Beachten Sie, dass Sie das `datatable` benutzerdefinierte Attribut angeben, das Sie der AWS Cloud Map Dienstinstanz zugeordnet haben, die Sie für die DynamoDB-Tabelle erstellt haben. Die Funktion generiert einen Schlüssel, der eine Zufallszahl zwischen 1 und 100 ist, und verknüpft ihn mit einem Wert, der an die Funktion übergeben wird, wenn sie aufgerufen wird.

```
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)
    }
```

Um Timeoutfehler zu vermeiden, aktualisieren Sie das Funktionstimeout nach der Bereitstellung der Funktion auf 5 Sekunden. Weitere Informationen finden Sie unter Configure Lambda function timeout im AWS Lambda Developer Guide.

Schritt 6: Erstellen Sie einen AWS Cloud Map App-Dienst und registrieren Sie die Lambda-Schreibfunktion als Instanz

In diesem Schritt erstellen Sie einen AWS Cloud Map Dienst und registrieren dann die Lambda-Schreibfunktion als Dienstinstanz.

1. Öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/
2. Wählen Sie in der linken Navigationsleiste Namespaces aus.
3. Wählen Sie aus der Liste der Namespaces den Namespace aus und klicken Sie auf Details anzeigen. **`cloudmap-tutorial`**
4. Wählen Sie im Abschnitt Dienste die Option Dienst erstellen aus und gehen Sie wie folgt vor.
   a. Geben Sie unter Servicename `app-service` ein.
   b. Behalten Sie die restlichen Standardwerte bei und wählen Sie Dienst erstellen aus.
5. Wählen Sie im Abschnitt Dienste den `app-service` Dienst aus und klicken Sie auf Details anzeigen.
6. Wählen Sie im Abschnitt Dienstinstanzen die Option Dienstinstanz registrieren aus.
7. Gehen Sie auf der Seite Dienstinstanz registrieren wie folgt vor.
   a. Wählen Sie als Instanztyp die Option Identifizierungsinformationen für eine andere Ressource aus.
   b. Geben Sie für Service-Instanz-ID Folgendes an`write-instance`.
   c. Geben Sie im Abschnitt Benutzerdefinierte Attribute die folgenden Schlüssel-Wert-Paare an.
      - Schlüssel =**`action`**, Wert = `write`
      - Schlüssel =`functionname`, Wert = `writefunction`

Schritt 7: Erstellen Sie die Lambda-Funktion zum Lesen von Daten

In diesem Schritt erstellen Sie eine Lambda-Funktion, die von Grund auf neu erstellt wurde und Daten in die von Ihnen erstellte DynamoDB-Tabelle schreibt.

Informationen zum Erstellen einer Lambda-Funktion finden Sie unter Erstellen einer Lambda-Funktion mit der Konsole im AWS Lambda Entwicklerhandbuch. Ermitteln Sie anhand der folgenden Tabelle, welche Optionen angegeben oder ausgewählt werden müssen.

Option	Wert
Funktionsname	Funktion lesen
Laufzeit	Python 3.12
Architektur	x86_64
Berechtigungen	Verwenden Sie eine bestehende Rolle
Vorhandene Rolle	cloudmap-tutorial-role

Nachdem Sie die Funktion erstellt haben, aktualisieren Sie den Beispielcode, sodass er den folgenden Python-Code wiedergibt, und stellen Sie dann die Funktion bereit. Die Funktion scannt die Tabelle und gibt alle Elemente zurück.

```
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)
    }
```

Um Timeout-Fehler zu vermeiden, aktualisieren Sie das Funktions-Timeout nach der Bereitstellung der Funktion auf 5 Sekunden. Weitere Informationen finden Sie unter Configure Lambda function timeout im AWS Lambda Developer Guide.

Schritt 8: Registrieren Sie die Lambda-Lesefunktion als AWS Cloud Map Dienstinstanz

In diesem Schritt registrieren Sie die Lambda-Lesefunktion als Dienstinstanz in dem `app-service` Service, den Sie zuvor erstellt haben.

1. Öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/
2. Wählen Sie in der linken Navigationsleiste Namespaces aus.
3. Wählen Sie aus der Liste der Namespaces den Namespace aus und klicken Sie auf Details anzeigen. **`cloudmap-tutorial`**
4. Wählen Sie im Abschnitt Dienste den **`app-service`** Dienst aus und klicken Sie auf Details anzeigen.
5. Wählen Sie im Abschnitt Dienstinstanzen die Option Dienstinstanz registrieren aus.
6. Gehen Sie auf der Seite Dienstinstanz registrieren wie folgt vor.

    a. Wählen Sie als Instanztyp die Option Identifizierungsinformationen für eine andere Ressource aus.

    b. Geben Sie für Service-Instanz-ID Folgendes an`read-instance`.

    c. Geben Sie im Abschnitt Benutzerdefinierte Attribute die folgenden Schlüssel-Wert-Paare an.

    - Schlüssel =**`action`**, Wert = `read`
    - Schlüssel =`functionname`, Wert = `readfunction`

Schritt 9: Lese- und Schreibclients erstellen und ausführen AWS CloudShell

Sie können Clientanwendungen erstellen und ausführen AWS CloudShell , die Code verwenden, um die Dienste zu ermitteln, in denen Sie konfiguriert haben, AWS Cloud Map und diese Dienste aufzurufen.

1. Öffnen Sie die AWS CloudShell Konsole unter https://console.aws.amazon.com/cloudshell/
2. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen zu erstellen`writefunction.py`.

   ```
   vim writeclient.py
   ```

3. Rufen Sie in der `writeclient.py` Datei den Einfügemodus auf, indem Sie die `i` Taste drücken. Kopieren Sie dann den folgenden Code und fügen Sie ihn ein. Dieser Code erkennt die Lambda-Funktion zum Schreiben von Daten, indem er `name=writeservice` im Dienst nach dem benutzerdefinierten Attribut sucht. `app-service` Der Name der Lambda-Funktion, die für das Schreiben von Daten in die DynamoDB-Tabelle verantwortlich ist, wird zurückgegeben. Dann wird die Lambda-Funktion aufgerufen und eine Beispielnutzlast übergeben, die als Wert in die Tabelle geschrieben wird.

   ```
   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. Drücken Sie die Escape-Taste`:wq`, geben Sie ein und drücken Sie die Eingabetaste, um die Datei zu speichern und zu beenden.
5. Verwenden Sie den folgenden Befehl, um den Python-Code auszuführen.

```
python3 writeclient.py
```

Die Ausgabe sollte eine `200` Antwort sein, die der folgenden ähnelt.

```
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. Um zu überprüfen, ob der Schreibvorgang im vorherigen Schritt erfolgreich war, erstellen Sie einen Leseclient.

   a. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen zu erstellen`readfunction.py`.

   ```
   vim readclient.py
   ```

   b. Drücken Sie in der `readclient.py` Datei die `i` Taste, um in den Einfügemodus zu wechseln. Kopieren Sie dann den folgenden Code und fügen Sie ihn ein. Dieser Code scannt die Tabelle und gibt den Wert zurück, den Sie im vorherigen Schritt in die Tabelle geschrieben haben.

   ```
   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. Drücken Sie die Escape-Taste`:wq`, geben Sie ein, und drücken Sie die Eingabetaste, um die Datei zu speichern und zu beenden.

d. Verwenden Sie den folgenden Befehl, um den Python-Code auszuführen.

```
python3 readclient.py
```

Die Ausgabe sollte wie folgt aussehen und den Wert auflisten, der durch Ausführen in die Tabelle geschrieben wurde, `writefunction.py` und den zufälligen Schlüssel, der in der Lambda-Schreibfunktion generiert wurde.

```
  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}}"}'
```

Schritt 10: Bereinigen Sie die Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, löschen Sie die Ressourcen, um zusätzliche Kosten zu vermeiden. AWS Cloud Map erfordert, dass Sie sie in umgekehrter Reihenfolge bereinigen, zuerst die Dienstinstanzen, dann die Dienste und schließlich den Namespace. Die folgenden Schritte führen Sie durch die Bereinigung der im Tutorial verwendeten AWS Cloud Map Ressourcen.

Um die AWS Cloud Map Ressourcen zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie aus der Liste der Namespaces den **`cloudmap-tutorial`** Namespace aus und klicken Sie auf Details anzeigen.
3. Wählen Sie auf der Seite mit den Namespace-Details aus der Liste der Dienste den **`data-service`** Dienst aus und klicken Sie auf Details anzeigen.
4. Wählen Sie im Abschnitt Dienstinstanzen die `data-instance` Instanz aus und klicken Sie auf Abmelden.

5. Wählen Sie mithilfe des Breadcrumbs oben auf der Seite cloudmap-tutorial.com aus, um zur Namespace-Detailseite zurückzukehren.
6. Wählen Sie auf der Namespace-Detailseite aus der Liste der Dienste den Datendienstdienst aus und klicken Sie auf Löschen.
7. Wiederholen Sie die Schritte 3-6 für den `app-service` Dienst `write-instance` und `read-instance` die Dienstinstanzen.
8. Wählen Sie in der linken Navigationsleiste Namespaces aus.
9. Wählen Sie den **`cloudmap-tutorial`** Namespace aus und wählen Sie Löschen.

In der folgenden Tabelle sind Verfahren aufgeführt, mit denen Sie die anderen im Tutorial verwendeten Ressourcen löschen können.

Ressource	Schritte	
DynamoDB-Tabelle	Schritt 6: (Optional) Löschen Sie Ihre DynamoDB-Tabelle, um Ressourcen zu bereinigen im Amazon DynamoDB DynamoDB-Entwickle rhandbuch	
Lambda-Funktionen und zugehörige IAM-Ausfü hrungsrolle	Im Developer Guide finden Sie Ordnung AWS Lambda	

Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden können, indem Sie AWS CLI

Dieses Tutorial zeigt, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden können. Sie erstellen eine Microservices-Anwendung, die Ressourcen mithilfe benutzerdefinierter Attribute dynamisch erkennt. AWS Cloud Map Die Anwendung besteht aus zwei Lambda-Funktionen, die Daten in eine DynamoDB-Tabelle schreiben und aus einer DynamoDB-Tabelle lesen, in der alle Ressourcen registriert sind. AWS Cloud Map

Eine AWS Management Console Version des Tutorials finden Sie unter. Erfahren Sie, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, führen Sie die Schritte unter durchZur Verwendung eingerichtet AWS Cloud Map.

Erstellen Sie einen AWS Cloud Map Namespace

Ein Namespace ist ein Konstrukt, das verwendet wird, um Dienste für eine Anwendung zu gruppieren. In diesem Schritt erstellen Sie einen Namespace, der es ermöglicht, Ressourcen über AWS Cloud Map API-Aufrufe auffindbar zu machen.

1. Führen Sie den folgenden Befehl aus, um einen HTTP-Namespace zu erstellen:

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

   Der Befehl gibt eine Vorgangs-ID zurück. Sie können den Status des Vorgangs mit dem folgenden Befehl überprüfen:

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

2. Sobald der Namespace erstellt wurde, können Sie seine ID zur Verwendung in nachfolgenden Befehlen abrufen:

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

3. Speichern Sie die Namespace-ID zur späteren Verwendung in einer Variablen:

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

Erstellen einer DynamoDB-Tabelle

Erstellen Sie als Nächstes eine DynamoDB-Tabelle, in der Daten für Ihre Anwendung gespeichert werden:

1. Führen Sie den folgenden Befehl aus, um die Tabelle zu erstellen:

```
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. Warten Sie, bis die Tabelle aktiv wird, bevor Sie fortfahren:

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

   Dieser Befehl wartet, bis die Tabelle vollständig erstellt und einsatzbereit ist.

Erstellen Sie einen AWS Cloud Map Datendienst und registrieren Sie die DynamoDB-Tabelle

Erstellen Sie nun einen Dienst in Ihrem Namespace, der Datenspeicherressourcen repräsentiert:

1. Führen Sie den folgenden Befehl aus, um einen AWS Cloud Map Dienst für Datenspeicherressourcen zu erstellen:

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

2. Rufen Sie die Dienst-ID für den Datendienst ab:

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

3. Registrieren Sie die DynamoDB-Tabelle als Dienstinstanz mit einem benutzerdefinierten Attribut, das den Tabellennamen angibt:

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

Das benutzerdefinierte Attribut `tablename=cloudmap` ermöglicht es anderen Diensten, den DynamoDB-Tabellennamen dynamisch zu ermitteln.

Erstellen Sie eine IAM-Rolle für Lambda-Funktionen

Erstellen Sie eine IAM-Rolle, die die Lambda-Funktionen für den Zugriff auf AWS Ressourcen verwenden:

1. Erstellen Sie das Dokument mit der Vertrauensrichtlinie für die IAM-Rolle:

```
cat > lambda-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

2. Führen Sie den folgenden Befehl aus, um die IAM-Rolle mithilfe der Vertrauensrichtlinie zu erstellen:

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

3. Erstellen Sie eine Datei für eine benutzerdefinierte IAM-Richtlinie mit den geringsten Rechten:

```
cat > cloudmap-policy.json << EOF
```

```
{
  "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"
    }
  ]
}
EOF
```

4. Erstellen Sie die Richtlinie und fügen Sie sie der IAM-Rolle hinzu:

```
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
```

Erstellen Sie die Lambda-Funktion zum Schreiben von Daten

Gehen Sie folgendermaßen vor, um eine Lambda-Funktion zu erstellen, die Daten in die DynamoDB-Tabelle schreibt:

1. Erstellen Sie die Python-Datei für die Schreibfunktion:

```
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
```

Diese Funktion ermittelt AWS Cloud Map den DynamoDB-Tabellennamen aus dem benutzerdefinierten Attribut und schreibt dann Daten in die Tabelle.

2. Package und implementieren Sie die Lambda-Funktion:

```
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. Aktualisieren Sie das Funktions-Timeout, um Timeout-Fehler zu vermeiden:

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

```
  --timeout 5
```

Erstellen Sie einen AWS Cloud Map App-Dienst und registrieren Sie die Lambda-Schreibfunktion

Gehen Sie folgendermaßen vor, um in Ihrem Namespace einen weiteren Dienst zur Darstellung von Anwendungsfunktionen zu erstellen:

1. Erstellen Sie einen Dienst für Anwendungsfunktionen:

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

2. Rufen Sie die Service-ID für den App-Dienst ab:

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

3. Registrieren Sie die Lambda-Schreibfunktion als Dienstinstanz mit benutzerdefinierten Attributen:

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

   Die benutzerdefinierten Attribute `action=write` `functionname=writefunction` ermöglichen es den Clients, diese Funktion anhand ihres Zwecks zu erkennen.

Erstellen Sie die Lambda-Funktion zum Lesen von Daten

Gehen Sie folgendermaßen vor, um eine Lambda-Funktion zu erstellen, die Daten aus der DynamoDB-Tabelle liest:

1. Erstellen Sie die Python-Datei für die Lesefunktion:

```
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
```

Diese Funktion ermittelt AWS Cloud Map auch den DynamoDB-Tabellennamen und liest dann Daten aus der Tabelle. Sie umfasst Kommentare zur Fehlerbehandlung und Seitennummerierung.

2. Package und implementieren Sie die Lambda-Funktion:

```
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. Aktualisieren Sie das Funktions-Timeout:

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

Registrieren Sie die Lambda-Lesefunktion als Dienstinstanz

Gehen Sie wie folgt vor, um die Lambda-Lesefunktion als weitere Dienstinstanz im App-Dienst zu registrieren:

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

```
  --attributes action=read,functionname=readfunction
```

Die benutzerdefinierten Attribute `action=read` `functionname=readfunction` ermöglichen es den Clients, diese Funktion anhand ihres Zwecks zu erkennen.

Client-Anwendungen erstellen und ausführen

Gehen Sie folgendermaßen vor, um eine Python-Client-Anwendung AWS Cloud Map zu erstellen, mit der die Schreibfunktion erkannt und aufgerufen wird:

1. Erstellen Sie eine Python-Datei für die Schreibclient-Anwendung:

```
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
```

Dieser Client verwendet die `QueryParameters` Option, um Dienstinstanzen mit dem `action=write` Attribut zu finden.

2. Erstellen Sie eine Python-Datei für die Readclient-Anwendung:

```
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. Führen Sie den Schreibclient aus, um der DynamoDB-Tabelle Daten hinzuzufügen:

```
python3 writeclient.py
```

   Die Ausgabe sollte eine erfolgreiche Antwort mit dem HTTP-Statuscode 200 anzeigen.

4. Führen Sie den Leseclient aus, um Daten aus der DynamoDB-Tabelle abzurufen:

```
python3 readclient.py
```

   Die Ausgabe sollte die Daten enthalten, die in die Tabelle geschrieben wurden, einschließlich der zufällig generierten ID und des Werts „Dies sind Testdaten“.

Bereinigen von -Ressourcen

Wenn Sie mit dem Tutorial fertig sind, bereinigen Sie die Ressourcen, um zusätzliche Kosten zu vermeiden.

1. Führen Sie zunächst den folgenden Befehl aus, um die Registrierung der Dienstinstanzen aufzuheben:

```
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. Führen Sie den folgenden Befehl aus, um die Dienste zu löschen:

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

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

3. Führen Sie den folgenden Befehl aus, um den Namespace zu löschen:

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

4. Führen Sie den folgenden Befehl aus, um die Lambda-Funktionen zu löschen:

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

5. Führen Sie den folgenden Befehl aus, um die IAM-Rolle und -Richtlinie zu löschen:

```
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. Führen Sie den folgenden Befehl aus, um die DynamoDB-Tabelle zu löschen:

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

7. Führen Sie den folgenden Befehl aus, um temporäre Dateien zu bereinigen:

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

AWS Cloud Map Namespaces

Ein Namespace ist eine logische Einheit, die verwendet wird AWS Cloud Map , um die Dienste einer Anwendung unter einem gemeinsamen Namen und einer gemeinsamen Erkennbarkeitsebene zu gruppieren. Wenn Sie einen Namespace erstellen, geben Sie Folgendes an:

- Ein Name, den Ihre Anwendung verwenden soll, um Instanzen zu erkennen.
- Die Methode, mit der Dienstinstanzen, bei denen Sie sich registrieren, ermittelt werden AWS Cloud Map können. Sie können entscheiden, ob Ihre Ressourcen öffentlich über das Internet, privat in einer bestimmten Virtual Private Cloud (VPC) oder nur durch API-Aufrufe entdeckt werden müssen.

Im Folgenden finden Sie allgemeine Konzepte zu Namespaces.

- Namespaces sind spezifisch für das, in dem AWS-Region sie erstellt wurden. Um sie AWS Cloud Map in mehreren Regionen verwenden zu können, müssen Sie in jeder Region Namespaces erstellen.
- Wenn Sie einen Namespace erstellen, der beispielsweise die Erkennung durch DNS-Abfragen in einer VPC ermöglicht, AWS Cloud Map wird automatisch eine private, von Route 53 gehostete Zone erstellt. Diese gehostete Zone kann mehreren zugeordnet werden. VPCs Weitere Informationen finden Sie unter Associate VPCWith HostedZone in der Amazon Route 53 API-Referenz.

Themen

Einen AWS Cloud Map Namespace zur Gruppierung von Anwendungsdiensten erstellen

Sie können einen Namespace erstellen, um Dienste für Ihre Anwendung unter einem benutzerfreundlichen Namen zu gruppieren, der die Erkennung von Anwendungsressourcen über API-Aufrufe oder DNS-Abfragen ermöglicht.

Optionen für die Instanzensuche

In der folgenden Tabelle sind die verschiedenen Optionen zur Instanzerkennung AWS Cloud Map und der entsprechende Namespace-Typ zusammengefasst, den Sie je nach den Diensten und der Konfiguration Ihrer Anwendung erstellen können.

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
HTTP	API-Aufrufe	Ressourcen in Ihrer Anwendung können andere Ressourcen nur ermitteln, indem sie die `DiscoverInstances` API aufrufen.	• DiscoverInstances • CreateHttpNamespace
Privates DNS	API-Aufrufe und DNS-Abfragen in einer VPC	Ressourcen in Ihrer Anwendung können andere Ressourcen ermitteln, indem sie die `DiscoverInstances` API aufrufen und die Nameserver in der privaten Route 53-Hosting-Zone abfragen, die automatisch erstellt wird. AWS Cloud Map Die gehostete Zone, die von erstellt wurde, AWS Cloud Map hat denselben Namen wie der Namespace	• DiscoverInstances • CreatePrivateDnsNamespace

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
		und enthält DNS-Einträge mit Namen im folgenden Format. *service-name* *namespace-name* . **Note** Route 53 Resolver löst DNS-Abfragen, die ihren Ursprung in der VPC haben, mithilfe von Datensätzen in der privaten Hosting-Zone auf. Wenn die private gehostete Zone keinen Datensatz enthält, der dem Domänennamen in einer DNS-Abfrage entspricht, antwortet Route 53 auf die Anfrage	

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
		mit `NXDOMAIN` (nicht vorhandene Domäne).	

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
Öffentliches DNS	API-Aufrufe und öffentliche DNS-Abfragen	Ressourcen in Ihrer Anwendung können andere Ressourcen ermitteln, indem sie die `DiscoverInstances` API aufrufen und die Nameserver in der öffentlichen Route 53-Hosting-Zone abfragen, AWS Cloud Map die automatisch erstellt wird. Die öffentlich gehostete Zone hat denselben Namen wie der Namespace und enthält DNS-Einträge mit Namen im folgenden Format. *`service-name namespace-name`* . **Note** Der Namespace-Name muss in diesem Fall ein Domainname sein, den	• DiscoverInstances • CreatePublicDnsNamespace

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
		Sie registriert haben.	

Verfahren

Sie können diesen Schritten folgen, um einen Namespace mit dem AWS CLI AWS Management Console, oder dem SDK für Python zu erstellen.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie Create namespace (Namespace erstellen) aus.
3. Geben Sie unter Namespace-Name einen Namen ein, der zur Erkennung von Instances verwendet wird.

   > **Note**
   >
   > - Namespaces, die für öffentliche DNS-Abfragen konfiguriert sind, müssen mit einer Top-Level-Domain enden. Beispiel, `.com`.
   > - Sie können einen internationalisierten Domänennamen (IDN) angeben, wenn Sie den Namen zuerst in Punycode umwandeln. Informationen zu Onlinekonvertern erhalten Sie, indem Sie eine Internetsuche nach „punycode converter“ durchführen.
   >
   >   Sie können auch einen internationalisierten Domänennamen in Punycode konvertieren, wenn Sie Namespaces programmgesteuert erstellen. Wenn Sie z. B. mit Java arbeiten, können Sie einen Unicode-Wert in Punycode umwandeln, indem Sie die Methode `toASCII` der java.net.IDN-Bibliothek verwenden.

4. (Optional) Geben Sie für die Namespace-Beschreibung Informationen über den Namespace ein, die auf der Seite Namespaces und unter Namespace-Informationen angezeigt werden. Sie können diese Informationen verwenden, um einen Namespace einfach zu identifizieren.

5. Bei der Instanzerkennung können Sie zwischen API-Aufrufen, API-Aufrufen und DNS-Abfragen in und API-Aufrufen und öffentlichen DNS-Abfragen wählen VPCs, um jeweils einen HTTP-, privaten DNS- oder öffentlichen DNS-Namespace zu erstellen. Weitere Informationen finden Sie unter Optionen für die Instanzensuche.

   Gehen Sie je nach Ihrer Auswahl wie folgt vor.

   - Wenn Sie API-Aufrufe und DNS-Abfragen in wählen VPCs, wählen Sie für VPC eine Virtual Private Cloud (VPC), der Sie den Namespace zuordnen möchten.
   - Wenn Sie API-Aufrufe und DNS-Abfragen in VPCs oder API-Aufrufe und öffentliche DNS-Abfragen wählen, geben Sie für TTL einen numerischen Wert in Sekunden an. Der TTL-Wert (Time to Live) bestimmt, wie lange der DNS-Resolver die Informationen für den SOA-DNS-Eintrag (Start of Authority) der Route 53-Hosting-Zone zwischenspeichert, die mit Ihrem Namespace erstellt wurde. Weitere Informationen zu TTL finden Sie unter TTL (Sekunden) im Amazon Route 53 Developer Guide.
6. (Optional) Wählen Sie unter Tags die Option Tags hinzufügen aus und geben Sie dann einen Schlüssel und einen Wert an, um Ihren Namespace zu kennzeichnen. Sie können ein oder mehrere Tags angeben, die Ihrem Namespace hinzugefügt werden sollen. Mithilfe von Tags können Sie Ihre AWS Ressourcen kategorisieren, sodass Sie sie einfacher verwalten können. Weitere Informationen finden Sie unter Verschlagworten Sie Ihre Ressourcen AWS Cloud Map.
7. Wählen Sie Create namespace (Namespace erstellen) aus. Sie können den Status des Vorgangs mithilfe ListOperationsvon anzeigen. Weitere Informationen finden Sie ListOperationsin der AWS Cloud Map API-Referenz

AWS CLI

- Erstellen Sie einen Namespace mit dem Befehl für den Instance-Discovery-Typ, den Sie bevorzugen (ersetzen Sie die *red* Werte durch Ihre eigenen).
  - Erstellen Sie einen HTTP-Namespace mit. `create-http-namespace` Dienstinstanzen, die mit einem HTTP-Namespace registriert wurden, können mithilfe einer `DiscoverInstances` Anfrage ermittelt werden, sie können jedoch nicht mithilfe von DNS ermittelt werden.

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

- Erstellen Sie einen privaten Namespace, der auf DNS basiert und nur innerhalb einer bestimmten Amazon VPC sichtbar ist, indem Sie. `create-private-dns-namespace` Sie können Instances, die in einem privaten DNS-Namespace registriert wurden, entweder mithilfe einer `DiscoverInstances` Anfrage oder mithilfe von DNS ermitteln

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

- Erstellen Sie einen öffentlichen Namespace, der auf DNS basiert und im Internet sichtbar ist, indem Sie. `create-public-dns-namespace` Sie können Instances erkennen, die bei einem öffentlichen DNS-Namespace registriert wurden, indem Sie entweder eine `DiscoverInstances`-Anforderung oder DNS verwenden.

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

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Erstellen Sie einen Namespace mit dem Befehl für den Instance-Discovery-Typ, den Sie bevorzugen würden (ersetzen Sie die *red* Werte durch Ihre eigenen):

- Erstellen Sie einen HTTP-Namespace mit. `create_http_namespace()` Dienstinstanzen, die mit einem HTTP-Namespace registriert wurden, können mithilfe von `discover_instances()` DNS ermittelt werden, sie können jedoch nicht ermittelt werden.

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

- Erstellen Sie einen privaten Namespace, der auf DNS basiert und nur innerhalb einer bestimmten Amazon VPC sichtbar ist, indem Sie. `create_private_dns_namespace()` Sie können Instances, die in einem privaten DNS-Namespace registriert wurden, entweder mithilfe von oder `discover_instances()` mithilfe von DNS ermitteln

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

- Erstellen Sie einen öffentlichen Namespace, der auf DNS basiert und im Internet sichtbar ist, indem Sie. `create_public_dns_namespace()` Sie können Instanzen, die in einem öffentlichen DNS-Namespace registriert wurden, entweder mithilfe von `discover_instances()` oder mithilfe von DNS ermitteln.

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

- Beispiel für eine Antwortausgabe

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

Nächste Schritte

Nachdem Sie einen Namespace erstellt haben, können Sie Dienste im Namespace erstellen, um Anwendungsressourcen zu gruppieren, die zusammen einem bestimmten Zweck in Ihrer Anwendung dienen. Ein Dienst dient als Vorlage für die Registrierung von Anwendungsressourcen als Instanzen. Weitere Informationen zum Erstellen von AWS Cloud Map Diensten finden Sie unterAWS Cloud Map Dienst für eine Anwendungskomponente erstellen.

AWS Cloud Map Namespaces auflisten

Nachdem Sie Namespaces erstellt haben, können Sie eine Liste der Namespaces anzeigen, die Sie erstellt haben, indem Sie die folgenden Schritte ausführen.

AWS Management Console

1. Melden Sie sich bei der an und öffnen Sie die Konsole unter AWS Management Console . AWS Cloud Map https://console.aws.amazon.com/cloudmap/
2. Wählen Sie im Navigationsbereich Namespaces aus, um eine Liste von Namespaces anzuzeigen. Sie können Namespaces nach Name, Beschreibung, Instanzerkennungsmodus oder Namespace-ID sortieren. Sie können auch einen Namespace-Namen oder eine Namespace-ID in das Suchfeld eingeben, um einen bestimmten Namespace zu finden und anzuzeigen.

AWS CLI

- Listet Namespaces mit dem Befehl auf. `list-namespaces`

```
aws servicediscovery list-namespaces
```

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie hier Anweisungen zur Installation, Konfiguration und Verwendung. `Boto3`
2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Namespaces auflisten mit. `list_namespaces()`

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

Beispiel für eine Antwortausgabe

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

Löschen eines AWS Cloud Map Namespaces

Wenn Sie einen Namespace nicht mehr verwenden, können Sie ihn löschen. Wenn Sie einen Namespace löschen, können Sie ihn nicht mehr verwenden, um Service-Instances zu registrieren oder zu erkennen.

> **Note**
>
> Wenn Sie beim Erstellen eines Namespace angeben, dass Sie Service-Instances entweder mithilfe von öffentlichen DNS-Abfragen oder DNS-Abfragen in ermitteln möchten VPCs, AWS Cloud Map wird eine öffentliche oder private gehostete Zone von Amazon Route 53 erstellt. Wenn Sie den Namespace löschen, wird die entsprechende gehostete Zone AWS Cloud Map gelöscht.

Bevor Sie einen Namespace löschen, müssen Sie alle Dienstinstanzen deregistrieren und anschließend alle Dienste löschen, die im Namespace erstellt wurden. Weitere Informationen erhalten Sie unter Abmeldung einer Dienstinstanz AWS Cloud Map und AWS Cloud Map Dienst löschen.

Nachdem Sie die Registrierung von Instanzen aufgehoben und Dienste gelöscht haben, die in einem Namespace erstellt wurden, gehen Sie wie folgt vor, um den Namespace zu löschen.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Konsole unter. AWS Cloud Map https://console.aws.amazon.com/cloudmap/
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie den Namespace aus, den Sie löschen möchten, und wählen Sie dann Löschen.
4. Bestätigen Sie, dass Sie den Dienst löschen möchten, indem Sie erneut Löschen wählen.

AWS CLI

- Löschen Sie einen Namespace mit dem `delete-namespace` Befehl (ersetzen Sie den *red* Wert durch Ihren eigenen). Wenn der Namespace immer noch einen oder mehrere Dienste enthält, schlägt die Anfrage fehl.

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

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Löschen Sie einen Namespace mit `delete_namespace()` (ersetzen Sie den *red* Wert durch Ihren eigenen). Wenn der Namespace immer noch einen oder mehrere Dienste enthält, schlägt die Anfrage fehl.

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

Beispiel für eine Antwortausgabe

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

AWS Cloud Map Dienstleistungen

Ein AWS Cloud Map Dienst ist eine Vorlage für die Registrierung von Dienstinstanzen, die aus dem Dienstnamen und gegebenenfalls der DNS-Konfiguration für den Dienst besteht. Sie können auch eine Integritätsprüfung einrichten, um den Integritätsstatus der Instanzen im Service zu ermitteln und fehlerhafte Ressourcen herauszufiltern. Ein Dienst kann eine Komponente Ihrer Anwendung darstellen. Sie können beispielsweise einen Dienst für Ressourcen erstellen, die Zahlungen für Ihre Anwendung abwickeln, und einen anderen für Ressourcen, die Benutzer verwalten.

Ein Dienst ermöglicht es Ihnen, die Ressourcen für eine Anwendung zu finden, indem Sie einen oder mehrere Endpunkte abrufen, die für die Verbindung mit der Ressource verwendet werden können. Der Speicherort der Ressourcen erfolgt mithilfe von DNS-Abfragen oder der AWS Cloud Map `DiscoverInstances`API-Aktion, je nachdem, wie Sie den Namespace konfiguriert haben. Sie können die AWS Cloud Map Konsole verwenden, um die Instanzerkennung auf Service-Ebene zu spezifizieren.

Mithilfe der API können Sie auch benutzerdefinierte Metadaten als Attribute auf Serviceebene angeben. `UpdateServiceAttributes` Sie können Serviceattribute festlegen, um doppelte Attribute zwischen Instanzen zu vermeiden, und diese Attribute ändern, ohne Änderungen an den Instanzattributen vornehmen zu müssen. Zu den Informationen, die Sie auf Serviceebene als Attribute angeben können, gehören unter anderem die folgenden:

- Gewichtungen von Endpunkten für die Verlagerung des Datenverkehrs bei schrittweisen Bereitstellungen.
- Diensteinstellungen wie API-Timeouts und vorgeschlagene Richtlinien für Wiederholungsversuche.

Weitere Informationen finden Sie UpdateServiceAttributesin der AWS Cloud Map API-Referenz.

Die folgenden Themen beschreiben Integritätsprüfungen und DNS-Konfigurationen für Dienste und enthalten Anweisungen zum Erstellen, Auflisten, Aktualisieren und Löschen eines Dienstes.

Themen

- AWS Cloud Map Dienste in einem Namespace auflisten
- AWS Cloud Map Dienst löschen

AWS Cloud Map Konfiguration der Service-Integritätsprüfung

Mithilfe von Integritätsprüfungen kann festgestellt werden, ob Serviceinstanzen fehlerfrei sind oder nicht. Wenn Sie bei der Diensterstellung keine Integritätsprüfung konfigurieren, wird der Datenverkehr unabhängig vom Integritätsstatus der Instances an die Serviceinstanzen weitergeleitet. Wenn Sie eine Integritätsprüfung konfigurieren, werden standardmäßig intakte Ressourcen AWS Cloud Map zurückgegeben. Sie können den `HealthStatus` `DiscoverInstances` API-Parameter verwenden, um Ressourcen nach dem Integritätsstatus zu filtern und eine Liste mit fehlerhaften Ressourcen abzurufen. Sie können die `GetInstancesHealthStatus` API auch verwenden, um den Integritätsstatus einer bestimmten Dienstinstanz abzurufen.

Sie können entweder eine Route 53-Zustandsprüfung oder eine benutzerdefinierte Integritätsprüfung eines Drittanbieters konfigurieren, wenn Sie einen AWS Cloud Map Dienst erstellen.

Route 53 Zustandsprüfungen

Wenn Sie Einstellungen für eine Amazon Route 53-Zustandsprüfung angeben, AWS Cloud Map erstellt es bei jeder Registrierung einer Instance eine Route 53-Zustandsprüfung und löscht die Zustandsprüfung, wenn Sie die Instance abmelden.

Ordnet bei öffentlichen DNS-Namespaces die Zustandsprüfung dem Route 53-Datensatz zu, AWS Cloud Map der AWS Cloud Map erstellt wird, wenn Sie eine Instance registrieren. Wenn Sie in der DNS-Konfiguration eines Dienstes `A` sowohl `AAAA` Eintragstypen als auch Eintragstypen angeben, wird eine Integritätsprüfung AWS Cloud Map erstellt, bei der anhand der Adresse der Zustand der IPv4 Ressource überprüft wird. Wenn der durch die IPv4 Adresse angegebene Endpunkt fehlerhaft ist, betrachtet Route 53 sowohl die als auch die Datensätze als fehlerhaft. `A` `AAAA` Wenn Sie in der DNS-Konfiguration eines `CNAME` Dienstes einen Eintragstyp angeben, können Sie keine Route 53-Zustandsprüfung konfigurieren.

Für Namespaces, für die Sie API-Aufrufe verwenden, um Instanzen zu ermitteln, AWS Cloud Map erstellt eine Route 53-Zustandsprüfung. Es gibt jedoch keinen DNS-Eintrag, mit dem die AWS Cloud Map Zustandsprüfung verknüpft werden könnte. Um festzustellen, ob eine Zustandsprüfung fehlerfrei ist, können Sie die Überwachung entweder mit der Route 53-Konsole oder mit Amazon konfigurieren CloudWatch. Weitere Informationen zur Verwendung der Route 53-Konsole finden Sie unter Get Notification When a Health Check Fails im Amazon Route 53-Entwicklerhandbuch. Weitere

Informationen zur Verwendung CloudWatch finden Sie PutMetricAlarmin der Amazon CloudWatch API-Referenz.

> **Note**
>
> - Sie können keine Amazon Route 53-Zustandsprüfung für einen Service konfigurieren, der in einem privaten DNS-Namespace erstellt wurde.
> - Ein Route 53-Zustandsprüfer AWS-Region sendet bei jeder Zustandsprüfung alle 30 Sekunden eine Integritätsprüfungsanfrage an einen Endpunkt. Im Durchschnitt erhält Ihr Endpunkt etwa alle zwei Sekunden eine Health Check-Anfrage. Zustandsprüfer stimmen sich jedoch nicht aufeinander ab. Daher sehen Sie manchmal mehrere Anforderungen in einer Sekunde, gefolgt von wenigen Sekunden ohne Zustandsprüfungen. Eine Liste der Regionen, in denen die Systemintegrität überprüft wird, finden Sie unter Regionen.

Informationen zu den Gebühren für Route 53-Gesundheitschecks finden Sie unter Route 53-Preise.

Benutzerdefinierte Zustandsprüfungen

Wenn Sie bei der Registrierung einer Instance die Verwendung einer benutzerdefinierten Integritätsprüfung konfigurieren AWS Cloud Map , müssen Sie eine Integritätsprüfung eines Drittanbieters verwenden, um den Zustand Ihrer Ressourcen zu bewerten. Benutzerdefinierte Zustandsprüfungen sind in folgenden Fällen nützlich:

- Sie können keine Route 53-Zustandsprüfung verwenden, da die Ressource nicht über das Internet verfügbar ist. Nehmen wir zum Beispiel an, Sie haben eine Instance, die sich in einer Amazon VPC befindet. Sie können eine benutzerdefinierte Zustandsprüfung für diese Instance verwenden. Damit der Health Check funktioniert, muss sich Ihr Health Checker jedoch auch in derselben VPC wie Ihre Instance befinden.
- Sie möchten eine Drittanbieter-Zustandsprüfung unabhängig vom Standort Ihrer Ressourcen verwenden.

Wenn Sie eine benutzerdefinierte Integritätsprüfung verwenden, AWS Cloud Map wird der Zustand einer bestimmten Ressource nicht direkt überprüft. Stattdessen überprüft der Integritätsprüfer eines Drittanbieters den Zustand der Ressource und gibt einen Status an Ihre Anwendung zurück. Ihre Bewerbung muss dann eine `UpdateInstanceCustomHealthStatus` Anfrage einreichen, die diesen Status an weiterleitet. AWS Cloud Map Wenn der ursprüngliche Status „Weitergeleitet“

lautet UNHEALTHY und UpdateInstanceCustomHealthStatus innerhalb von 30 Sekunden kein weiterer Status übermittelt wird, wird bestätigtHEALTHY, dass die Ressource fehlerhaft ist. AWS Cloud Map beendet die Weiterleitung des Datenverkehrs zu dieser Ressource.

AWS Cloud Map Dienst-DNS-Konfiguration

Wenn Sie einen Dienst in einem Namespace erstellen, der die Instanzerkennung durch DNS-Abfragen unterstützt, werden Route 53-DNS-Einträge AWS Cloud Map erstellt. Sie müssen eine Route 53-Routingrichtlinie und einen DNS-Eintragstyp angeben, die für alle Route 53-DNS-Einträge gelten, die AWS Cloud Map erstellt werden.

Routing-Richtlinie

Eine Routingrichtlinie bestimmt, wie Route 53 auf die DNS-Abfragen reagiert, die für die Erkennung von Dienstinstanzen verwendet werden. Die unterstützten Routingrichtlinien und wie sie sich darauf beziehen, AWS Cloud Map lauten wie folgt.

Gewichtetes Routing

Route 53 gibt den entsprechenden Wert von einer zufällig ausgewählten AWS Cloud Map Dienstinstanz aus den Instanzen zurück, die Sie mit demselben AWS Cloud Map Dienst registriert haben. Alle Datensätze haben die gleiche Gewichtung. Sie können also nicht mehr oder weniger Datenverkehr zu einer Instance weiterleiten.

Nehmen wir beispielsweise an, der Service umfasst Konfigurationen für einen A-Datensatz und eine Integritätsprüfung, und Sie verwenden den Dienst, um 10 Instanzen zu registrieren. Route 53 antwortet auf DNS-Abfragen mit der IP-Adresse für eine zufällig ausgewählte Instance aus der Liste der fehlerfreien Instances. Wenn keine Instanzen fehlerfrei sind, reagiert Route 53 auf DNS-Abfragen, als ob alle Instanzen fehlerfrei wären.

Wenn Sie keine Integritätsprüfung für den Service definieren, nimmt Route 53 an, dass alle Instances fehlerfrei sind, und gibt den entsprechenden Wert für eine zufällig ausgewählte Instance zurück.

Weitere Informationen finden Sie unter Weighted Routing im Amazon Route 53 Developer Guide.

Mehrwertiges Antwort-Routing

Wenn Sie eine Zustandsprüfung für den Service definieren und das Ergebnis der Zustandsprüfung fehlerfrei ist, gibt Route 53 den entsprechenden Wert für bis zu acht Instances zurück.

Nehmen wir beispielsweise an, dass der Service Konfigurationen für einen A-Datensatz und eine Integritätsprüfung umfasst. Sie verwenden den Dienst, um 10 Instances zu registrieren. Route 53 beantwortet DNS-Abfragen mit IP-Adressen nur für maximal acht fehlerfreie Instanzen. Wenn weniger als acht Instanzen fehlerfrei sind, beantwortet Route 53 jede DNS-Anfrage mit den IP-Adressen aller fehlerfreien Instanzen.

Wenn Sie keine Integritätsprüfung für den Service definieren, nimmt Route 53 an, dass alle Instances fehlerfrei sind, und gibt die Werte für bis zu acht Instances zurück.

Weitere Informationen finden Sie unter Mehrwertiges Answer Routing im Amazon Route 53 Developer Guide.

Datensatztyp

Ein Route 53-DNS-Eintragstyp bestimmt den Werttyp, den Route 53 als Antwort auf die DNS-Abfragen zurückgibt, die für die Erkennung von Service-Instances verwendet werden. Die verschiedenen DNS-Eintragstypen, die Sie angeben können, und die zugehörigen Werte, die von Route 53 als Antwort auf Abfragen zurückgegeben werden, lauten wie folgt.

A

Wenn Sie diesen Typ angeben, gibt Route 53 die IP-Adresse der Ressource in einem IPv4 Format zurück, z. B. 192.0.2.44.

AAAA

Wenn Sie diesen Typ angeben, gibt Route 53 die IP-Adresse der Ressource im IPv6 Format 2001:0 db 8:85 a 3:0000:0000:abcd: 0001:2345 zurück.

CNAME

Wenn Sie diesen Typ angeben, gibt Route 53 den Domänennamen der Ressource zurück (z. B. www.example.com).

> **Note**
>
> - Um einen CNAME-DNS-Eintrag zu konfigurieren, müssen Sie die Routing-Richtlinie für gewichtetes Routing angeben.
> - Wenn Sie einen CNAME-DNS-Eintrag konfigurieren, können Sie keine Route 53-Zustandsprüfung konfigurieren.

SRV

Wenn Sie diesen Typ angeben, gibt Route 53 den Wert für einen SRV Datensatz zurück. Der Wert für einen SRV-Datensatz verwendet die folgenden Werte:

`priority weight port service-hostname`

Berücksichtigen Sie dabei Folgendes:

- Die Werte `priority` und `weight` sind beide auf 1 gesetzt und können nicht geändert werden.
- For AWS Cloud Map verwendet den Wert`port`, den Sie für Port (AWS_INSTANCE_PORT) angeben, wenn Sie eine Instanz registrieren.
- Der Wert `service-hostname` setzt sich aus den folgenden Werten zusammen:
  - Der Wert, den Sie für die Service-Instanz-ID (InstanceID) angeben, wenn Sie eine Instanz registrieren
  - Name des Service
  - Name des Namespace

  Nehmen wir beispielsweise an, Sie geben test als Instanz-ID an, wenn Sie eine Instanz registrieren. Der Name des Dienstes ist Backend und der Name des Namespaces ist example.com. AWS Cloud Map weist dem **`service-hostname`** Attribut im SRV-Datensatz den folgenden Wert zu:

  `test.backend.example.com`

> **Note**
>
> Wenn Sie bei der Registrierung einer IPv4 Instanz Werte wie eine IPv6 Adresse, eine Adresse oder beides angeben, AWS Cloud Map werden automatisch A - und/oder AAAA-Einträge erstellt, die denselben Namen wie der Wert **`service-hostname`** im SRV-Datensatz haben.

Sie können Datensatztypen in den folgenden Kombinationen angeben:

- A
- AAAA
- A und AAAA

- CNAME
- SRV

Wenn Sie die Eintragstypen A und AAAA angeben, können Sie bei der Registrierung einer Instance eine IPv6 IP-Adresse, eine IP-Adresse oder beides angeben. IPv4

AWS Cloud Map Dienst für eine Anwendungskomponente erstellen

Nachdem Sie einen Namespace erstellt haben, können Sie Dienste erstellen, um verschiedene Komponenten Ihrer Anwendung darzustellen, die bestimmten Zwecken dienen. Sie können beispielsweise einen Dienst für Ressourcen in Ihrer Anwendung erstellen, die Zahlungen verarbeiten.

> **Note**
>
> Sie können nicht mehrere Dienste erstellen, auf die über DNS-Abfragen zugegriffen werden kann, deren Namen sich nur in der Groß- und Kleinschreibung unterscheiden (wie EXAMPLE und example). Der Versuch, dies zu tun, führt dazu, dass diese Dienste denselben DNS-Namen haben. Wenn Sie einen Namespace verwenden, auf den nur über API-Aufrufe zugegriffen werden kann, können Sie Dienste mit Namen erstellen, die sich nur durch Groß- und Kleinschreibung unterscheiden.

Gehen Sie wie folgt vor, um einen Service mit dem AWS Management Console AWS CLI, und SDK für Python zu erstellen.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie auf der Seite Namespaces den Namespace aus, dem Sie den Service hinzufügen möchten.
4. Wählen Sie auf der *namespace-name* Seite Namespace: die Option Dienst erstellen aus.
5. Geben Sie unter Dienstname einen Namen ein, der die Instanzen beschreibt, die Sie registrieren, wenn Sie diesen Dienst verwenden. Der Wert wird verwendet, um AWS Cloud Map Dienstinstanzen entweder in API-Aufrufen oder in DNS-Abfragen zu ermitteln.

> **Note**
>
> Wenn Sie bei der Registrierung einer Instanz einen SRV-Eintrag erstellen möchten AWS Cloud Map und ein System verwenden, das ein bestimmtes SRV-Format (z. B. HAProxy) erfordert, geben Sie Folgendes für den Dienstnamen an:
>
> - Beginnen Sie den Namen mit einem Unterstrich (_), zum Beispiel _exampleservice.
> - Beenden Sie den Namen beispielsweise mit ***._protocol***. _tcp.
>
> Wenn Sie eine Instanz registrieren, AWS Cloud Map erstellt sie einen SRV-Eintrag und weist ihnen einen Namen zu, indem der Dienstname und der Namespace-Name verkettet werden, zum Beispiel: _exampleservice. _tcp.beispiel.com

6. (Optional) Geben Sie unter Dienstbeschreibung eine Beschreibung für den Dienst ein. Die Beschreibung, die Sie hier eingeben, wird auf der Seite Dienste und auf der Detailseite für jeden Dienst angezeigt.
7. Wenn der Namespace DNS-Abfragen unterstützt, können Sie unter Konfiguration der Diensterkennung die Auffindbarkeit auf Dienstebene konfigurieren. Wählen Sie, ob Sie sowohl API-Aufrufe als auch DNS-Abfragen oder nur API-Aufrufe für die Erkennung von Instanzen in diesem Service zulassen möchten.

> **Note**
>
> Wenn Sie API-Aufrufe wählen, AWS Cloud Map werden bei der Registrierung einer Instanz keine SRV-Einträge erstellt.

Wenn Sie API und DNS wählen, gehen Sie wie folgt vor, um DNS-Einträge zu konfigurieren. Sie können DNS-Einträge hinzufügen oder entfernen.

1. Wählen Sie unter Routing-Richtlinie die Amazon Route 53-Routing-Richtlinie für die DNS-Einträge aus, die bei der Registrierung von Instances AWS Cloud Map erstellt werden. Sie können zwischen gewichtetem Routing und mehrwertigem Antwort-Routing wählen. Weitere Informationen finden Sie unter Routing-Richtlinie.

> **Note**
>
> Sie können die Konsole nicht verwenden, um AWS Cloud Map zu konfigurieren, dass bei der Registrierung einer Instanz ein Route 53-Aliaseintrag erstellt wird. Wenn Sie Aliaseinträge für einen Elastic Load Balancing Load Balancer erstellen möchten AWS Cloud Map , wenn Sie Instances programmgesteuert registrieren, wählen Sie Weighted Routing für die Routing-Richtlinie.

2. Wählen Sie unter Datensatztyp den DNS-Eintragstyp aus, der bestimmt, welche Route 53 als Antwort auf DNS-Abfragen zurückgibt. AWS Cloud Map Weitere Informationen finden Sie unter Datensatztyp.
3. Geben Sie für TTL einen numerischen Wert an, um die Gültigkeitsdauer (Time to Live, TTL) in Sekunden auf Service-Ebene zu definieren. Der Wert von TTL bestimmt, wie lange DNS-Resolver Informationen für diesen Datensatz zwischenspeichert, bevor die Resolver eine weitere DNS-Anfrage an Amazon Route 53 weiterleiten, um die Einstellungen zu aktualisieren.

8. Wählen Sie unter Konfiguration der Integritätsprüfung für Optionen zur Integritätsprüfung die Art der Zustandsprüfung aus, die für Dienstinstanzen gilt. Sie können wählen, ob Sie keine Zustandsprüfungen konfigurieren möchten, oder Sie können zwischen einer Route 53-Zustandsprüfung oder einer externen Zustandsprüfung für Ihre Instances wählen. Weitere Informationen finden Sie unter AWS Cloud Map Konfiguration der Service-Integritätsprüfung.

> **Note**
>
> Route 53-Zustandsprüfungen können nur für Dienste in öffentlichen DNS-Namespaces konfiguriert werden.

Wenn Sie Route 53-Zustandsprüfungen wählen, geben Sie die folgenden Informationen an.

1. Geben Sie für den Schwellenwert für Fehler eine Zahl zwischen 1 und 10 ein, die die Anzahl der aufeinanderfolgenden Route 53-Zustandsprüfungen definiert, die eine Dienstinstanz bestehen oder nicht bestehen muss, damit sich ihr Integritätsstatus ändert.
2. Wählen Sie für Health Check Protocol die Methode aus, mit der Route 53 den Zustand der Dienstinstanzen überprüft.

3. Wenn Sie das HTTP - oder HTTPS-Zustandsprüfungsprotokoll wählen, geben Sie für Health Check Path einen Pfad an, den Amazon Route 53 bei der Durchführung von Zustandsprüfungen anfordern soll. Der Pfad kann ein beliebiger Wert sein, z. B. die Datei`/docs/route53-health-check.html`. Wenn die Ressource fehlerfrei ist, ist der zurückgegebene Wert ein HTTP-Statuscode im 2xx- oder 3xx-Format. Sie können auch Abfragezeichenfolgenparameter einschließen, z. B. `/welcome.html?language=jp&login=y`. Die AWS Cloud Map -Konsole fügt automatisch einen vorangestellten Schrägstrich (/) hinzu.

Weitere Informationen zu Route 53-Zustandsprüfungen finden Sie unter So bestimmt Amazon Route 53, ob eine Zustandsprüfung fehlerfrei ist im Amazon Route 53-Entwicklerhandbuch.

9. (Optional) Wählen Sie unter Tags die Option Tags hinzufügen aus und geben Sie dann einen Schlüssel und einen Wert an, um Ihren Namespace zu kennzeichnen. Sie können ein oder mehrere Tags angeben, die Ihrem Namespace hinzugefügt werden sollen. Mithilfe von Tags können Sie Ihre AWS Ressourcen kategorisieren, sodass Sie sie einfacher verwalten können. Weitere Informationen finden Sie unter Verschlagworten Sie Ihre Ressourcen AWS Cloud Map.
10. Wählen Sie Create service.

AWS CLI

- Erstellen Sie einen Dienst mit dem `create-service` Befehl. Ersetzen Sie die *red* Werte durch Ihre eigenen.

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

Ausgabe:

```
{
        "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)

Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.

1. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

2. Erstellen Sie einen Dienst mit`create_service()`. Ersetzen Sie die *red* Werte durch Ihre eigenen. Weitere Informationen finden Sie unter create_service.

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

```
    NamespaceId='ns-xxxxxxxxxxx',
)
```

Beispiel für eine Antwortausgabe

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

Nächste Schritte

Nachdem Sie einen Service erstellt haben, können Sie Ihre Anwendungsressourcen als Dienstinstanzen registrieren, die Informationen darüber enthalten, wie Ihre Anwendung die Ressource finden kann. Weitere Informationen zur Registrierung von AWS Cloud Map Dienstinstanzen finden Sie unterEine Ressource als Dienstinstanz registrieren AWS Cloud Map.

Sie können auch benutzerdefinierte Metadaten wie Endpunktgewichte, API-Timeouts und Wiederholungsrichtlinien als Dienstattribute angeben, nachdem Sie einen Service erstellt haben. Weitere Informationen finden Sie unter ServiceAttributes und `UpdateServiceAttributes` in der AWS Cloud Map -API-Referenz.

Aktualisierung eines AWS Cloud Map Dienstes

Abhängig von der Konfiguration eines Dienstes können Sie dessen Tags, den Schwellenwert für Fehler bei der Zustandsprüfung von Route 53 und die Gültigkeitsdauer (TTL) für DNS-Resolver aktualisieren. Gehen Sie wie folgt vor, um einen Dienst zu aktualisieren.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie auf der Seite Namespaces den Namespace aus, in dem der Dienst erstellt wurde.
4. Wählen Sie auf der ***namespace-name*** Seite Namespace: den Service aus, den Sie bearbeiten möchten, und klicken Sie auf Details anzeigen.
5. Wählen Sie auf der *service-name* Seite Service: die Option Bearbeiten aus.

   > **Note**
   >
   > Sie können den Workflow „Schaltfläche bearbeiten“ nicht verwenden, um Werte für Dienste zu bearbeiten, die nur API-Aufrufe für die Instanzerkennung zulassen. Sie können jedoch Tags auf der *service-name* Seite Service: hinzufügen oder entfernen.

6. Auf der Seite Service bearbeiten können Sie unter Servicebeschreibung jede zuvor festgelegte Beschreibung für den Service aktualisieren oder eine neue Beschreibung hinzufügen. Sie können auch Tags hinzufügen und TTL für DNS-Resolver aktualisieren.
7. Unter DNS-Konfiguration können Sie für TTL einen aktualisierten Zeitraum in Sekunden angeben, der bestimmt, wie lange DNS-Resolver Informationen für diesen Datensatz zwischenspeichern, bevor die Resolver eine weitere DNS-Anfrage an Amazon Route 53 weiterleiten, um aktualisierte Einstellungen zu erhalten.
8. Wenn Sie Route 53-Zustandsprüfungen eingerichtet haben, können Sie für den Schwellenwert für Fehler eine neue Zahl zwischen 1 und 10 angeben, die die Anzahl der aufeinanderfolgenden Route 53-Zustandsprüfungen definiert, die eine Dienstinstanz bestehen oder fehlschlagen muss, damit sich ihr Integritätsstatus ändert.
9. Wählen Sie Service aktualisieren.

AWS CLI

- Aktualisieren Sie einen Dienst mit dem `update-service` Befehl (ersetzen Sie den *red* Wert durch Ihren eigenen).

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

Ausgabe:

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

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Aktualisieren Sie einen Service mit `update_service()` (ersetzen Sie den *red* Wert durch Ihren eigenen).

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

```
    }
)
```

Beispiel für eine Antwortausgabe

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

AWS Cloud Map Dienste in einem Namespace auflisten

Um eine Liste der Services anzuzeigen, die Sie in einem Namespace erstellt haben, gehen Sie wie folgt vor.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie den Namen des Namespace aus, der die gewünschten Services enthält. Unter Dienste können Sie eine Liste aller Dienste anzeigen und den Dienstnamen oder die ID in das Suchfeld eingeben, um einen bestimmten Dienst zu finden.

AWS CLI

- Dienste mit dem `list-services` Befehl auflisten. Der folgende Befehl listet alle Dienste in einem Namespace auf, wobei die Namespace-ID als Filter verwendet wird. Ersetzen Sie den Wert *red* durch Ihren eigenen.

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

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht Boto3 installiert haben, finden Sie hier Anweisungen zur Installation, Konfiguration und VerwendungBoto3.

2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Dienste auflisten mit`list_services()`.

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

Beispiel für eine Antwortausgabe

```
{
    '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-xxxxxxxxxxxxxxxx',
            'Name': 'myservice',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

AWS Cloud Map Dienst löschen

Bevor Sie einen Service löschen können, müssen Sie alle Service-Instances abmelden, die mit dem Service registriert wurden. Weitere Informationen finden Sie unter Abmeldung einer Dienstinstanz AWS Cloud Map.

Nachdem Sie alle mit dem Service registrierten Instanzen deregistriert haben, führen Sie das folgende Verfahren durch, um den Service zu löschen.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter. https://console.aws.amazon.com/cloudmap/
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie die Option für den Namespace aus, der den Service enthält, den Sie löschen möchten.
4. Wählen Sie auf der *namespace-name* Seite Namespace: die Option für den Dienst aus, den Sie löschen möchten.
5. Wählen Sie Löschen.
6. Bestätigen Sie, dass Sie den Service löschen möchten.

AWS CLI

- Löschen Sie einen Dienst mit dem `delete-service` Befehl (ersetzen Sie den *red* Wert durch Ihren eigenen).

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

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Löschen Sie einen Dienst mit `delete_service()` (ersetzen Sie den *red* Wert durch Ihren eigenen).

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

Beispiel für eine Antwortausgabe

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

AWS Cloud Map Dienstinstanzen

Eine Service-Instance enthält Informationen dazu, wie Sie eine Ressource für eine Anwendung finden, z. B. einen Webserver. Nachdem Sie Instances registriert haben, finden Sie sie mithilfe von DNS-Abfragen oder der AWS Cloud Map DiscoverInstancesAPI-Aktion. Zu den Ressourcen, die Sie registrieren können, gehören unter anderem die folgenden:

- EC2 Amazon-Instanzen
- Amazon-DynamoDB-Tabellen
- Amazon-S3-Buckets
- Amazon-Simple-Queue-Service-(Amazon-SQS)-Warteschlangen
- APIs auf Amazon API Gateway bereitgestellt

Sie können Attributwerte für Services-Instances angeben, und Kunden können diese Attribute verwenden, um die zurückgegebenen Ressourcen zu filtern. AWS Cloud Map Beispiel: Eine Anwendung kann Ressourcen in einer bestimmten Bereitstellungsphase anfordern, z. B. BETA oder PROD. Sie können Attribute auch für die Versionierung verwenden.

In den folgenden Verfahren wird beschrieben, wie Sie Ressourcen in Ihrer Anwendung als Dienstinstanzen registrieren, eine Liste der registrierten Instanzen in einem Dienst anzeigen, bestimmte Instanzparameter bearbeiten und die Registrierung einer Instanz aufheben können.

Themen

Eine Ressource als Dienstinstanz registrieren AWS Cloud Map

Sie können die Ressourcen Ihrer Anwendung als Instanzen in einem AWS Cloud Map Dienst registrieren. Nehmen wir beispielsweise an, Sie haben einen Dienst erstellt, der `users` für alle Anwendungsressourcen aufgerufen wird, die Benutzerdaten verwalten. Anschließend können Sie eine DynamoDB-Tabelle, die zum Speichern von Benutzerdaten verwendet wird, als Instanz in diesem Dienst registrieren.

> **Note**
>
> Die folgenden Funktionen sind auf der AWS Cloud Map Konsole nicht verfügbar:
>
> - Wenn Sie eine Service-Instance über die Konsole registrieren, können Sie keinen Aliaseintrag erstellen, der den Traffic an einen Elastic Load Balancing (ELB) -Load Balancer weiterleitet. Wenn Sie eine Instance registrieren, müssen Sie das Attribut `AWS_ALIAS_DNS_NAME` einschließen. Weitere Informationen finden Sie unter RegisterInstance in der AWS Cloud Map -API-Referenz.
> - Wenn Sie eine Instance mit einem Service registrieren, der eine benutzerdefinierte Zustandsprüfung enthält, können Sie nicht den anfänglichen Status für die benutzerdefinierte Zustandsprüfung angeben. Standardmäßig lautet der anfängliche Status einer benutzerdefinierten Zustandsprüfung Fehlerfrei. Wenn Sie möchten, dass der anfängliche Status Fehlerhaft lautet, registrieren Sie die Instance programmgesteuert und schließen Sie das Attribut `AWS_INIT_HEALTH_STATUS` ein. Weitere Informationen finden Sie unter RegisterInstance in der AWS Cloud Map -API-Referenz.

Gehen Sie folgendermaßen vor, um eine Instance in einem Service zu registrieren.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie auf der Seite Namespaces den Namespace aus, der den Service enthält, den Sie als Vorlage für die Registrierung einer Service-Instance verwenden möchten.
4. Wählen Sie auf der *`namespace-name`* Seite Namespace: den Dienst aus, den Sie verwenden möchten.
5. Wählen Sie auf der *`service-name`* Seite Service: die Option Dienstinstanz registrieren aus.
6. Wählen Sie auf der Seite Dienstinstanz registrieren einen Instanztyp aus. Abhängig von der Konfiguration der Namespace-Instance Discovery können Sie wählen, ob Sie eine IP-Adresse, eine EC2 Amazon-Instance-ID oder andere identifizierende Informationen für eine Ressource angeben möchten, die keine IP-Adresse hat.

> **Note**
>
> Sie können eine EC2 Instance nur in HTTP-Namespaces auswählen.

7. Geben Sie für die Dienstinstanz-ID einen Bezeichner an, der der Dienstinstanz zugeordnet ist.

> **Note**
>
> Wenn Sie eine bestehende Instanz aktualisieren möchten, geben Sie die ID an, die der Instanz zugeordnet ist, die Sie aktualisieren möchten. Verwenden Sie dann die nächsten Schritte, um Werte zu aktualisieren und die Instanz erneut zu registrieren.

8. Führen Sie je nach Wahl des Instanztyps die folgenden Schritte aus.

> **Important**
>
> Sie können das `AWS_` Präfix (ohne Berücksichtigung von Groß- und Kleinschreibung) nicht in einem Schlüssel verwenden, wenn Sie ein benutzerdefiniertes Attribut angeben.

Instance-Typ	Schritte
IP-Adresse	a. Geben Sie unter Standardattribute für IPv4Adresse gegebenen falls eine IPv4 Adresse an, unter der Ihre Anwendung auf die Ressource zugreifen kann, die dieser Dienstinstanz zugeordne t ist. b. Geben Sie als IPv6 Adresse gegebenenfalls

Instance-Typ	Schritte	
	eine IPv6 IP-Adresse an, über die Ihre Anwendung en auf die Ressource zugreifen können, die dieser Dienstinstanz zugeordnet ist. c. Geben Sie für Port einen beliebigen Port an, den Ihre Anwendung enthalten muss, um auf die Ressource zuzugreif en, die dieser Dienstins tanz zugeordnet ist. Ein Port ist erforderl ich, wenn der Service einen SRV-Eintrag oder eine Amazon Route 53-Zustandsprüfung umfasst. d. (Optional) Geben Sie unter Benutzerdefinierte Attribute alle Schlüssel -Wert-Paare an, die Sie der Ressource zuordnen möchten.	

Instance-Typ	Schritte	
EC2 Instanz	a. Wählen Sie zum EC2 Beispiel Instance-ID die ID der EC2 Amazon-In stance aus, die Sie als AWS Cloud Map Service-Instance registrieren möchten. b. (Optional) Geben Sie unter Benutzerdefinierte Attribute alle Schlüssel -Wert-Paare an, die Sie der Ressource zuordnen möchten.	

Instance-Typ	Schritte	
Identifying information for another resource (Identifizierende Informationen für eine andere Ressource)	a. Wenn die Dienstkonfiguration einen CNAME-DNS-Eintrag enthält, wird unter Standardattribute ein CNAME-Feld angezeigt. Geben Sie für CNAME den Domainnamen an, den Route 53 als Antwort auf DNS-Abfragen zurückgeben soll (z. B.). `example.com` b. Geben Sie unter Benutzerdefinierte Attribute alle identifizierenden Informationen für eine Ressource, bei der es sich nicht um eine IP-Adresse oder eine EC2 Amazon-Instance-ID handelt, als Schlüssel-Wert-Paar an. Sie können beispielsweise eine Lambda-Funktion registrieren, indem Sie einen aufgerufenen Schlüssel angeben `function` und den Namen der Lambda-Funktion als Wert angeben. Sie können auch einen Schlüssel angeben, der aufgerufen wird, `name` und einen Namen angeben, den Sie für	

Instance-Typ	Schritte	
	die programmatische Instanzerkennung verwenden können.	

9. Wählen Sie Register service instance (Service-Instance registrieren) aus.

AWS CLI

- Wenn Sie eine `RegisterInstance` Anfrage einreichen:
  - Für jeden DNS-Eintrag, den Sie in dem von angegebenen Dienst definieren`ServiceId`, wird ein Eintrag in der Hosting-Zone erstellt oder aktualisiert, der dem entsprechenden Namespace zugeordnet ist.
  - Wenn der Dienst Folgendes umfasst`HealthCheckConfig`, wird eine Integritätsprüfung auf der Grundlage der Einstellungen in der Integritätsprüfungskonfiguration erstellt.
  - Alle Integritätsprüfungen sind jedem der neuen oder aktualisierten Datensätze zugeordnet.

  Registrieren Sie eine Dienstinstanz mit dem `register-instance` Befehl (ersetzen Sie die *red* Werte durch Ihre eigenen).

```
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. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Wenn Sie eine `RegisterInstance` Anfrage einreichen:

- Für jeden DNS-Eintrag, den Sie in dem von angegebenen Dienst definieren`ServiceId`, wird ein Eintrag in der Hosting-Zone erstellt oder aktualisiert, der dem entsprechenden Namespace zugeordnet ist.
- Wenn der Dienst Folgendes umfasst`HealthCheckConfig`, wird eine Integritätsprüfung auf der Grundlage der Einstellungen in der Integritätsprüfungskonfiguration erstellt.
- Alle Integritätsprüfungen sind jedem der neuen oder aktualisierten Datensätze zugeordnet.

Registrieren Sie eine Dienstinstanz mit `register_instance()` (ersetzen Sie die *red* Werte durch Ihre eigenen).

```
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)
```

Beispiel für eine Antwortausgabe

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

AWS Cloud Map Dienstinstanzen auflisten

Um eine Liste der Service-Instances anzuzeigen, die Sie mit einem Service registriert haben, gehen Sie wie folgt vor.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie den Namen des Namespace aus, der den Service enthält, für den Sie Service-Instances auflisten möchten.
4. Wählen Sie den Namen des Service aus, mit dem Sie die Service-Instances erstellt haben. Unter Serviceinstanzen wird eine Liste der Instanzen angezeigt. Sie können die Instanz-ID in das Suchfeld eingeben, um eine bestimmte Instanz aufzulisten.

AWS CLI

- Listet Dienstinstanzen mit dem `list-instances` Befehl auf (ersetzen Sie den *red* Wert durch Ihren eigenen).

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

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als Ihren Service verwenden.

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

3. Listet Dienstinstanzen auf mit `list_instances()` (ersetze den *red* Wert durch deinen eigenen).

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

Beispiel für eine Antwortausgabe

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

Eine AWS Cloud Map Dienstinstanz aktualisieren

Sie können Service-Instances auf zwei Arten aktualisieren, je nachdem, welche Werte Sie aktualisieren möchten:

- Alle Werte aktualisieren: Wenn Sie Werte aktualisieren möchten, die Sie bei der Registrierung für eine Serviceinstanz angegeben haben, einschließlich benutzerdefinierter Attribute, müssen Sie die Dienstinstanz erneut registrieren und alle Werte neu angeben. Gehen Sie wie unter beschrieben vor Eine Ressource als Dienstinstanz registrieren AWS Cloud Map und geben Sie die Instanz-ID der vorhandenen Dienstinstanz als Dienstinstanz-ID an.

  Alternativ können Sie die `RegisterInstance`API verwenden. Sie können die ID der vorhandenen Instanz und des Dienstes mithilfe der `ServiceId` Parameter `InstanceId` und angeben und andere Werte erneut angeben.
- Nur benutzerdefinierte Attribute aktualisieren: Wenn Sie nur die benutzerdefinierten Attribute für eine Service-Instance aktualisieren möchten, müssen Sie die Instance nicht erneut registrieren. Sie können nur diese Werte aktualisieren. Siehe Aktualisierung der benutzerdefinierten Attribute für eine Dienstinstanz.

Aktualisierung der benutzerdefinierten Attribute für eine Dienstinstanz

So aktualisieren Sie nur benutzerdefinierte Attribute für eine Service-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie auf der Seite Namespaces den Namespace aus, der den Service enthält, den Sie ursprünglich für die Registrierung der Service-Instance verwendet haben.
4. Wählen Sie auf der *namespace-name* Seite Namespace: den Dienst aus, mit dem Sie die Dienstinstanz registriert haben.
5. Wählen Sie auf der *service-name* Seite Service: den Namen der Dienstinstanz aus, die Sie aktualisieren möchten.
6. Wählen Sie im Abschnitt Custom attributes (Benutzerdefinierte Attribute) die Option Edit (Bearbeiten) aus.
7. Fügen Sie auf der *instance-name* Seite Dienstinstanz bearbeiten: benutzerdefinierte Attribute hinzu, entfernen oder aktualisieren Sie sie. Sie können Schlüssel und Werte für vorhandene Attribute aktualisieren.
8. Wählen Sie Update service instance (Service-Instance aktualisieren) aus.

Abmeldung einer Dienstinstanz AWS Cloud Map

Bevor Sie einen Service löschen können, müssen Sie alle Service-Instances abmelden, die mit dem Service registriert wurden.

Um eine Service-Instance abzumelden, gehen Sie wie folgt vor.

AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter. https://console.aws.amazon.com/cloudmap/
2. Wählen Sie im Navigationsbereich Namespaces aus.
3. Wählen Sie die Option für den Namespace aus, der die Service-Instance enthält, die Sie abmelden möchten.
4. Wählen Sie auf der *namespace-name* Seite Namespace: den Dienst aus, mit dem Sie die Dienstinstanz registriert haben.

5. Wählen Sie auf der *service-name* Seite Service: die Dienstinstanz aus, deren Registrierung Sie aufheben möchten.
6. Wählen Sie Deregister.
7. Bestätigen Sie, dass Sie die Service-Instance abmelden möchten.

AWS CLI

- Melden Sie eine Dienstinstanz mit dem `deregister-instance` Befehl ab (ersetzen Sie die *red* Werte durch Ihre eigenen). Dieser Befehl löscht die Amazon Route 53 53-DNS-Einträge und alle Integritätsprüfungen, die für die angegebene Instance AWS Cloud Map erstellt wurden.

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

AWS SDK for Python (Boto3)

1. Falls Sie es noch nicht `Boto3` installiert haben, finden Sie `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als Service verwenden.

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

3. Deregistrieren Sie eine Dienstinstanz mit `deregister-instance()` (ersetzen Sie die *red* Werte durch Ihre eigenen). Dieser Befehl löscht die Amazon Route 53 53-DNS-Einträge und alle Integritätsprüfungen, die für die angegebene Instance AWS Cloud Map erstellt wurden.

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

Beispiel für eine Antwortausgabe

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

Sicherheit in AWS Cloud Map

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS -Compliance-Programme regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für gelten AWS Cloud Map, finden Sie unter AWS Services in Umfang nach Compliance-Programmen.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Die folgende Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können AWS Cloud Map. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen AWS Cloud Map , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer AWS Cloud Map Ressourcen unterstützen.

Themen

Identity and Access Management für AWS Cloud Map

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. AWS Cloud Map IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. AWS Cloud Map

Dienstbenutzer — Wenn Sie den AWS Cloud Map Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr AWS Cloud Map Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Unter Fehlerbehebung bei AWS Cloud Map Identität und Zugriff finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Feature in AWS Cloud Map haben.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die AWS Cloud Map Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf AWS Cloud Map. Es ist Ihre Aufgabe, zu bestimmen, auf welche AWS Cloud Map Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen kann AWS Cloud Map, finden Sie unterWie AWS Cloud Map funktioniert mit IAM.

IAM-Administrator: Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf AWS Cloud Map verfassen können. Beispiele für AWS Cloud Map identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Cloud Map

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung von Anforderungen finden Sie unter AWS Signature Version 4 für API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und AWS Multi-Faktor-Authentifizierung (MFA) in IAM im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-

Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Methoden für die Übernahme einer Rolle im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu

gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verwenden einer

IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe

oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer Inline-Richtlinie wählen, finden Sie unter Auswählen zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer

IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Ressourcenkontrollrichtlinien (RCPs) — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter Resource Control Policies (RCPs) im AWS Organizations Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird,

ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

Wie AWS Cloud Map funktioniert mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren AWS Cloud Map, mit welchen IAM-Funktionen Sie arbeiten können. AWS Cloud Map

IAM-Feature	AWS Cloud Map Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Forward Access Sessions (FAS)	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie AWS Cloud Map und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für AWS Cloud Map

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für AWS Cloud Map

Beispiele für AWS Cloud Map identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Cloud Map

Ressourcenbasierte Richtlinien finden Sie in AWS Cloud Map

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie

ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für AWS Cloud Map

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der AWS Cloud Map Aktionen finden Sie unter Aktionen definiert von AWS Cloud Map in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix AWS Cloud Map verwendet:

```
servicediscovery
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

```
        ]
```

Beispiele für AWS Cloud Map identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Cloud Map

Politische Ressourcen für AWS Cloud Map

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der AWS Cloud Map Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Resources defined by AWS Cloud Map in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von AWS Cloud Map definierte Aktionen.

Beispiele für AWS Cloud Map identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Cloud Map

Bedingungsschlüssel für Richtlinien für AWS Cloud Map

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der AWS Cloud Map Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für AWS Cloud Map in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Aktionen definiert von AWS Cloud Map.

AWS Cloud Map unterstützt die folgenden dienstspezifischen Bedingungsschlüssel, mit denen Sie Ihre IAM-Richtlinien detailliert filtern können.

`servicediscovery:NamespaceArn`

Ein Filter, mit dem Sie Objekte abrufen, indem Sie den Amazon-Ressourcennamen (ARN) für den zugehörigen Namespace angeben

`servicediscovery:NamespaceName`

Ein Filter, mit dem Sie Objekte abrufen, indem Sie den Namen des zugehörigen Namespace angeben

`servicediscovery:ServiceArn`

Ein Filter, mit dem Sie Objekte abrufen, indem Sie den Amazon-Ressourcennamen (ARN) für den entsprechenden Service angeben

`servicediscovery:ServiceName`

Ein Filter, mit dem Sie Objekte abrufen, indem Sie den Namen des zugehörigen Service angeben

Beispiele für AWS Cloud Map identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Cloud Map

ACLs in AWS Cloud Map

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit AWS Cloud Map

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit AWS Cloud Map

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln von einer Benutzerrolle zu einer IAM-Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Zugriffssitzungen weiterleiten für AWS Cloud Map

Unterstützt Forward Access Sessions (FAS): Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für AWS Cloud Map

Unterstützt Servicerollen: Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Rollen zum Delegieren von Berechtigungen an einen AWS-Service erstellen im IAM-Benutzerhandbuch.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle könnte die AWS Cloud Map -Funktionalität beeinträchtigen. Bearbeiten Sie Servicerollen nur, AWS Cloud Map wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für AWS Cloud Map

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS Cloud Map

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, AWS Cloud Map -Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden AWS Cloud Map, einschließlich des Formats von ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Cloud Map in der Service Authorization Reference.

Themen

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS Cloud Map Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer

Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der AWS Cloud Map -Konsole

Um auf die AWS Cloud Map Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS Cloud Map Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die AWS Cloud Map Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die AWS Cloud Map *ConsoleAccess* oder die *ReadOnly* AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

AWS Cloud Map Beispiel für einen Konsolenzugriff

Um vollen Zugriff auf die AWS Cloud Map Konsole zu gewähren, erteilen Sie die Berechtigungen in der folgenden Berechtigungsrichtlinie:

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":"*"
      }
   ]
}
```

Gründe, warum die Berechtigungen erforderlich sind

`servicediscovery:*`

Ermöglicht das Ausführen aller AWS Cloud Map Aktionen.

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

Lassen Sie uns gehostete Zonen AWS Cloud Map verwalten, wenn Sie öffentliche und private DNS-Namespaces erstellen und löschen.

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

Lassen Sie uns die Zustandsprüfungen AWS Cloud Map verwalten, wenn Sie bei der Erstellung eines Service Amazon Route 53-Zustandsprüfungen einbeziehen.

`ec2:DescribeVpcs` und **`ec2:DescribeRegions`**

Lassen Sie uns private gehostete Zonen AWS Cloud Map verwalten.

Erlauben Sie AWS Cloud Map Benutzern, ihre eigenen Berechtigungen einzusehen

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI AWS OR-API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
```

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

Erlauben Sie Lesezugriff auf alle Ressourcen AWS Cloud Map

Die folgende Berechtigungsrichtlinie gewährt dem Benutzer Lesezugriff auf alle AWS Cloud Map - Ressourcen:

JSON

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

AWS Cloud Map Beispiel für eine Dienstinstanz

Das folgende Beispiel zeigt eine Berechtigungsrichtlinie, die einem Benutzer die Erlaubnis erteilt, Dienstinstanzen zu registrieren, zu deregistrieren und zu ermitteln. Der Abschnitt `Sid` (die Anweisungs-ID) ist optional:

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": "*"
      }
   ]
}
```

Die Richtlinie gewährt Berechtigungen für Aktionen, die erforderlich sind, Service-Instances zu registrieren und zu verwalten. Die Route 53-Berechtigung ist erforderlich, wenn Sie öffentliche oder private DNS-Namespaces verwenden, da Route 53-Datensätze und Zustandsprüfungen AWS Cloud Map erstellt, aktualisiert und gelöscht werden, wenn Sie Instances registrieren und deregistrieren. Das Platzhalterzeichen (*) in `Resource` gewährt Zugriff auf alle AWS Cloud Map Instances und Route 53-Datensätze und Zustandsprüfungen, die dem aktuellen Konto gehören. AWS

Beispiel für einen AWS Cloud Map Service erstellen

Wenn Sie eine Berechtigungsrichtlinie hinzufügen, die es einer IAM-Identität ermöglicht, einen AWS Cloud Map Service zu erstellen, müssen Sie den Amazon-Ressourcennamen (ARN) sowohl des AWS Cloud Map Namespace als auch des Services im Ressourcenfeld angeben. Der ARN umfasst die Region, die Konto-ID und die Namespace-ID. Da Sie noch nicht wissen, wie die Service-ID des Dienstes lautet, empfehlen wir die Verwendung eines Platzhalters. Im Folgenden finden Sie ein Beispiel für einen Richtlinienausschnitt.

JSON

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

Beispiel für ein Namespaces erstellen AWS Cloud Map

Die folgende Berechtigungsrichtlinie ermöglicht es Benutzern, alle Arten von Namespaces zu erstellen: AWS Cloud Map

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 verwaltete Richtlinien für AWS Cloud Map

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AWSCloud MapDiscoverInstanceAccess

Sie können `AWSCloudMapDiscoverInstanceAccess` an Ihre IAM-Entitäten anhängen. Bietet Zugriff auf die AWS Cloud Map Discovery-API.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSCloudMapDiscoverInstanceAccess in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSCloud MapReadOnlyAccess

Sie können `AWSCloudMapReadOnlyAccess` an Ihre IAM-Entitäten anhängen. Gewährt Nur-Lese-Zugriff auf alle AWS Cloud Map Aktionen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSCloudMapReadOnlyAccess in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSCloud MapRegisterInstanceAccess

Sie können `AWSCloudMapRegisterInstanceAccess` an Ihre IAM-Entitäten anhängen. Gewährt schreibgeschützten Zugriff auf Namespaces und Dienste und erteilt die Berechtigung, Dienstinstanzen zu registrieren und zu deregistrieren.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSCloudMapRegisterInstanceAccess in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSCloud MapFullAccess

Sie können `AWSCloudMapFullAccess` an Ihre IAM-Entitäten anhängen. Bietet vollen Zugriff auf alle AWS Cloud Map Aktionen

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSCloudMapFullAccess in der Referenz zu von AWS verwalteten Richtlinien.

AWS Cloud Map Aktualisierungen der AWS verwalteten Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien, die AWS Cloud Map seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Abonnieren Sie den RSS-Feed auf der Seite mit dem AWS Cloud Map Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen zu erhalten.

Änderung	Beschreibung	Datum
AWSCloudMapDiscoverInstanceAccess, AWSCloudMapRegisterInstanceAccess, AWSCloudMapReadOnlyAccess— Aktualisierungen vorhandener Richtlinien.	AWS Cloud Map hat diese Richtlinien aktualisiert, um den Zugriff auf die neuen AWS Cloud Map `DiscoverInstanceRevision` API-Operationen zu ermöglichen.	15. August 2023

AWS Cloud Map Referenz zu API-Berechtigungen

Wenn Sie die Zugriffskontrolle einrichten und eine Berechtigungsrichtlinie schreiben, die Sie an eine IAM-Identität anhängen können (identitätsbasierte Richtlinien), können Sie die folgende Liste als Referenz verwenden. Die Liste enthält jede AWS Cloud Map API-Aktion und die Aktionen, für die Sie Zugriffsberechtigungen gewähren müssen. Sie geben die Aktionen im `Action` Feld für die Richtlinie an. Einzelheiten zu dem Ressourcenwert, den Sie im `Resource` Feld oder in der IAM-Richtlinie angeben müssen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Cloud Map in der Service Authorization Reference.

Für einige Operationen können AWS Cloud Map Sie in Ihren IAM-Richtlinien spezifische Bedingungsschlüssel verwenden. Weitere Informationen finden Sie unter Bedingungsschlüssel für AWS Cloud Map in der Service Authorization Reference.

Um eine Aktion anzugeben, verwenden Sie das Präfix `servicediscovery`, gefolgt vom Namen der API-Aktion (z. B. `servicediscovery:CreatePublicDnsNamespace` und `route53:CreateHostedZone`).

Erforderliche Berechtigungen für AWS Cloud Map -Aktionen

CreateHttpNamespace

Erforderliche Berechtigungen (API-Aktion):

- `servicediscovery:CreateHttpNamespace`

CreatePrivateDnsNamespace

Erforderliche Berechtigungen (API-Aktion):

- `servicediscovery:CreatePrivateDnsNamespace`

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

CreatePublicDnsNamespace

Erforderliche Berechtigungen (API-Aktion):

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

CreateService

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:CreateService`

DeleteNamespace

Erforderliche Berechtigungen (API-Aktion):

- `servicediscovery:DeleteNamespace`

DeleteService

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:DeleteService`

DeleteServiceAttributes

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:DeleteServiceAttributes`

DeregisterInstance

Erforderliche Berechtigungen (API-Aktion):

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

DiscoverInstances

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:DiscoverInstances`

GetInstance

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:GetInstance`

GetInstancesHealthStatus

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:GetInstancesHealthStatus`

GetNamespace

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:GetNamespace`

GetOperation

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:GetOperation`

GetService

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:GetService`

GetServiceAttributes

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:GetServiceAttributes`

ListInstances

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:ListInstances`

ListNamespaces

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:ListNamespaces`

ListOperations

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:ListOperations`

ListServices

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:ListServices`

ListTagsForResource

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:ListTagsForResource`

RegisterInstance

Erforderliche Berechtigungen (API-Aktion):

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

TagResource

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:TagResource`

UntagResource

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:UntagResource`

UpdateHttpNamespace

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:UpdateHttpNamespace`

UpdateInstanceCustomHealthStatus

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:UpdateInstanceCustomHealthStatus`

UpdatePrivateDnsNamespace

Erforderliche Berechtigungen (API-Aktion):

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

UpdatePublicDnsNamespace

Erforderliche Berechtigungen (API-Aktion):

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

UpdateService

Erforderliche Berechtigungen (API-Aktion):

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

UpdateServiceAttributes

Erforderliche Berechtigungen (API-Aktion): `servicediscovery:UpdateServiceAttributes`

Fehlerbehebung bei AWS Cloud Map Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS Cloud Map IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS Cloud Map
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS Cloud Map Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS Cloud Map

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `servicediscovery:`*`GetWidget`*-Berechtigungen verfügt.

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

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `servicediscovery:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS Cloud Mapübergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS Cloud Map auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS Cloud Map Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen AWS Cloud Map unterstützt werden, finden Sie unter. Wie AWS Cloud Map funktioniert mit IAM

- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Konformitätsprüfung für AWS Cloud Map

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- Referenz für berechtigte HIPAA-Services – Listet berechtigte HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den

Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.

- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerelementreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in AWS Cloud Map

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

AWS Cloud Map ist in erster Linie ein globaler Service. Sie können sie jedoch verwenden, AWS Cloud Map um Route 53-Zustandsprüfungen zu erstellen, die den Zustand von Ressourcen in bestimmten Regionen überprüfen, z. B. EC2 Amazon-Instances und Elastic Load Balancing Load Balancers.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Infrastruktursicherheit in AWS Cloud Map

Als verwalteter Dienst AWS Cloud Map ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff AWS Cloud Map über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Sie können den Sicherheitsstatus Ihrer VPC verbessern, indem Sie die Verwendung eines AWS Cloud Map VPC-Endpunkts mit Schnittstelle konfigurieren. Weitere Informationen finden Sie unter Zugriff AWS Cloud Map über einen Schnittstellenendpunkt (AWS PrivateLink).

Zugriff AWS Cloud Map über einen Schnittstellenendpunkt (AWS PrivateLink)

Sie können verwenden AWS PrivateLink , um eine private Verbindung zwischen Ihrer VPC und AWS Cloud Map herzustellen. Sie können darauf zugreifen, AWS Cloud Map als ob es in Ihrer VPC wäre, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung zu verwenden. Instances in Ihrer VPC benötigen für den Zugriff AWS Cloud Map keine öffentlichen IP-Adressen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellen-Endpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Hierbei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Eingangspunkt für den Datenverkehr dienen, der für AWS Cloud Map bestimmt ist.

Weitere Informationen finden Sie unter Zugriff auf AWS-Services über AWS PrivateLink im AWS PrivateLink -Leitfaden.

Überlegungen zu AWS Cloud Map

Bevor Sie einen Schnittstellen-Endpunkt für einrichten AWS Cloud Map, lesen Sie die Überlegungen im Handbuch.AWS PrivateLink

Wenn Ihre Amazon VPC kein Internet-Gateway hat und Ihre Aufgaben den `awslogs` Protokolltreiber verwenden, um CloudWatch Protokollinformationen an Logs zu senden, müssen Sie einen VPC-Schnittstellen-Endpunkt für CloudWatch Logs erstellen. Weitere Informationen finden Sie unter Using CloudWatch Logs with Interface VPC Endpoints im Amazon CloudWatch Logs-Benutzerhandbuch.

VPC-Endpunkte unterstützen keine AWS regionsübergreifenden Anfragen. Stellen Sie sicher, dass Sie Ihren Endpunkt innerhalb derselben Region erstellen, in der Sie Ihre API-Aufrufe an AWS Cloud Map ausgeben möchten.

VPC-Endpunkte unterstützen nur von Amazon bereitgestellten DNS über Amazon Route 53. Wenn Sie Ihre eigene DNS verwenden möchten, können Sie die bedingte DNS-Weiterleitung nutzen. Weitere Informationen finden Sie unter DHCP-Optionssätze im Amazon VPC-Benutzerhandbuch.

Die mit dem VPC-Endpunkt verbundene Sicherheitsgruppe muss eingehende Verbindungen über Port 443 aus dem privaten Subnetz der Amazon VPC zulassen.

Erstellen Sie einen Schnittstellenendpunkt für AWS Cloud Map

Sie können einen Schnittstellenendpunkt für die AWS Cloud Map Verwendung entweder der Amazon VPC-Konsole oder der AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im AWS PrivateLink -Leitfaden.

Erstellen Sie einen Schnittstellenendpunkt für die AWS Cloud Map Verwendung der folgenden Servicenamen:

> **Note**
>
> `DiscoverInstances`Die API wird über diese beiden Endpunkte nicht verfügbar sein.

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

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

Erstellen Sie einen Schnittstellenendpunkt für die AWS Cloud Map Datenebene, um mithilfe der folgenden Dienstnamen auf die `DiscoverInstances` API zuzugreifen:

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

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

> **Note**
>
> Sie müssen die Hostpräfixinjektion deaktivieren, wenn Sie `DiscoverInstances` mit den regionalen oder zonalen VPCE-DNS-Namen für Datenebenen-Endpunkte anrufen. Der AWS SDKs Service-Endpunkt AWS CLI und stellt dem Service-Endpunkt verschiedene Host-Präfixe voran, wenn Sie jeden API-Vorgang aufrufen, wodurch ungültige URLS erzeugt werden, wenn Sie einen VPC-Endpunkt angeben.

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an die AWS Cloud Map Verwendung des standardmäßigen regionalen DNS-Namens stellen. Beispiel, `servicediscovery.us-east-1.amazonaws.com`.

Die AWS PrivateLink VPCE-Verbindung wird in jeder Region unterstützt, in der sie unterstützt AWS Cloud Map wird. Ein Kunde muss jedoch überprüfen, welche Availability Zones VPCE unterstützen, bevor er einen Endpunkt definiert. Um herauszufinden, welche Availability Zones mit VPC-Schnittstellen-Endpunkten in einer Region unterstützt werden, verwenden Sie den describe-vpc-endpoint-services Befehl oder den. AWS Management Console Mit den folgenden Befehlen werden beispielsweise die Availability Zones zurückgegeben, in denen Sie VPC-Endpunkte mit AWS Cloud Map Schnittstelle in der Region USA Ost (Ohio) bereitstellen können:

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

Überwachung AWS Cloud Map

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Verfügbarkeit und Performance Ihrer AWS -Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen Ausfall an mehreren Stellen leichter debuggen können, falls einer auftritt. Aber bevor Sie mit der Überwachung beginnen, sollten Sie einen Überwachungsplan mit Antworten auf die folgenden Fragen erstellen:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungstools werden verwendet?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Themen

AWS Cloud Map API-Aufrufe protokollieren mit AWS CloudTrail

AWS Cloud Map ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS-Service ausgeführten Aktionen bereitstellt. CloudTrail erfasst alle API-Aufrufe AWS Cloud Map als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS Cloud Map Konsole und Codeaufrufen für die AWS Cloud Map API-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an die die Anfrage gestellt wurde AWS Cloud Map, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Die Anforderung wurde im Namen eines IAM-Identity-Center-Benutzers erstellt.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.

- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

CloudTrail ist in Ihrem aktiv AWS-Konto , wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem. AWS-Region Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Arbeiten mit dem CloudTrail Ereignisverlauf. Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder CloudTrailLake-Event-Datenspeicher.

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS Management Console sind regionsübergreifend. Sie können mithilfe von AWS CLI einen Einzel-Region- oder einen Multi-Region-Trail erstellen. Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten AWS-Regionen in Ihrem Konto erfassen. Wenn Sie einen Einzel-Region-Trail erstellen, können Sie nur die Ereignisse anzeigen, die im AWS-Region des Trails protokolliert wurden. Weitere Informationen zu Trails finden Sie unter Erstellen eines Trails für Ihr AWS-Konto und Erstellen eines Trails für eine Organisation im AWS CloudTrail -Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise. Informationen zu Amazon-S3-Preisen finden Sie unter Amazon S3 – Preise.

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON-Format in das Apache ORC-Format. ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von erweiterten Ereignisselektoren auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter Arbeiten mit AWS CloudTrail Lake im AWS CloudTrail Benutzerhandbuch.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die Preisoption aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise.

AWS Cloud Map Datenereignisse in CloudTrail

Datenereignisse liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. das Erkennen einer registrierten Instanz in einem Namespace). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Protokolliert standardmäßig CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preisgestaltung.

Sie können Datenereignisse für die AWS Cloud Map Ressourcentypen mithilfe der CloudTrail Konsole oder CloudTrail API-Operationen protokollieren. AWS CLI Weitere Informationen zum Protokollieren von Datenereignissen finden Sie unter Protokollieren von Datenereignissen mit dem AWS Management Console und Protokollieren von Datenereignissen mit dem AWS Command Line Interface im AWS CloudTrail -Benutzerhandbuch.

In der folgenden Tabelle sind die AWS Cloud Map Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, den Sie in der Liste Datenereignistyp auf der CloudTrail Konsole auswählen können. In der Wertspalte resources.type wird der `resources.type` Wert angezeigt, den Sie bei der Konfiguration erweiterter Event-Selektoren mithilfe von oder angeben würden. AWS CLI CloudTrail APIs In der CloudTrail Spalte APIs Protokollierte Daten werden die API-Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
		• DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • DiscoverInstancesRevision

Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den Feldern `eventName`, `readOnly` und `resources.`ARN filtern, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Feldern finden Sie unter AdvancedFieldSelector in der API-Referenz zu AWS CloudTrail

Das folgende Beispiel zeigt, wie erweiterte Ereignisselektoren konfiguriert werden, um alle AWS Cloud Map Datenereignisse zu protokollieren.

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

AWS Cloud Map Verwaltungsereignisse in CloudTrail

Verwaltungsereignisse bieten Informationen über Verwaltungsvorgänge, die an Ressourcen in Ihrem ausgeführt werden AWS-Konto. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. CloudTrail Protokolliert standardmäßig Verwaltungsereignisse.

AWS Cloud Map protokolliert alle Operationen auf der AWS Cloud Map Steuerungsebene als Verwaltungsereignisse. Eine Liste der Vorgänge auf der AWS Cloud Map Steuerungsebene, bei denen eine AWS Cloud Map Anmeldung erfolgt CloudTrail, finden Sie in der AWS Cloud Map API-Referenz.

AWS Cloud Map Beispiele für Ereignisse

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API-Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt ein CloudTrail Verwaltungsereignis, das den `CreateHTTPNamespace` Vorgang demonstriert.


```
{
    "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"
}
```

Das folgende Beispiel zeigt ein CloudTrail Datenereignis, das den `DiscoverInstances` Vorgang demonstriert.

```
{
            "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"
```

```
        }
```

Informationen zu CloudTrail Datensatzinhalten finden Sie im AWS CloudTrail Benutzerhandbuch unter CloudTrailDatensatzinhalt.

Verschlagworten Sie Ihre Ressourcen AWS Cloud Map

Ein Tag ist eine Bezeichnung, die Sie einer AWS Ressource zuweisen. Jeder Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert, beides können Sie bestimmen.

Mithilfe von Tags können Sie Ihre AWS Ressourcen beispielsweise nach Zweck, Eigentümer oder Umgebung kategorisieren. Wenn Sie viele Ressourcen desselben Typs haben, können Sie bestimmte Ressourcen basierend auf den zugewiesenen Tags schnell bestimmen. Sie können beispielsweise eine Reihe von Tags für Ihre AWS Cloud Map Services definieren, um Ihnen zu helfen, den Besitzer und die Stack-Ebene der einzelnen Services nachzuverfolgen. Sie sollten für jeden Ressourcentyp einen konsistenten Satz von Tag-Schlüsseln entwickeln.

Tags werden nicht automatisch Ihren Ressourcen zugewiesen. Nachdem Sie ein Tag hinzugefügt haben, können Sie jederzeit Tag-Schlüssel und -Werte bearbeiten oder Tags aus einer Ressource entfernen. Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.

Tags haben keine semantische Bedeutung AWS Cloud Map und werden ausschließlich als Zeichenfolge interpretiert. Sie können den Wert eines Tags (Markierung) zwar auf eine leere Zeichenfolge, jedoch nicht null festlegen. Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.

Sie können mit Tags arbeiten, indem Sie die AWS Management Console AWS CLI, und die AWS Cloud Map API verwenden.

Wenn Sie AWS Identity and Access Management (IAM) verwenden, können Sie steuern, welche Benutzer in Ihrem AWS Konto berechtigt sind, Tags zu erstellen, zu bearbeiten oder zu löschen.

So werden Ressourcen markiert

Sie können neue oder bestehende AWS Cloud Map Namespaces und Dienste taggen.

Wenn Sie die AWS Cloud Map Konsole verwenden, können Sie Tags auf neue Ressourcen anwenden, wenn diese erstellt werden, oder jederzeit auf bestehende Ressourcen, indem Sie die Registerkarte „Tags“ auf der entsprechenden Ressourcenseite verwenden.

Wenn Sie die AWS Cloud Map API, das AWS CLI oder ein AWS SDK verwenden, können Sie mithilfe des `tags` Parameters der entsprechenden API-Aktion Tags auf neue Ressourcen oder mithilfe der

TagResourceAPI-Aktion auf vorhandene Ressourcen anwenden. Weitere Informationen finden Sie unter TagResource.

Bei einigen Aktionen zur Ressourcenerstellung können Sie Tags für eine Ressource angeben, wenn die Ressource erstellt wird. Wenn Tags während der Ressourcenerstellung nicht angewendet werden können, schlägt die Ressourcenerstellung fehl. Auf diese Weise wird sichergestellt, dass Ressourcen, die Sie bei der Erstellung markieren möchten, entweder mit angegebenen Tags oder gar nicht erstellt werden. Wenn Sie Ressourcen zum Zeitpunkt der Erstellung markieren, müssen Sie nach der Ressourcenerstellung keine benutzerdefinierten Tagging-Skripts ausführen.

In der folgenden Tabelle werden die AWS Cloud Map Ressourcen beschrieben, die markiert werden können, und die Ressourcen, die bei der Erstellung markiert werden können.

Unterstützung für AWS Cloud Map das Markieren von Ressourcen

Ressource	Unterstützt Tags (Markierungen)	Unterstützt Tag-Propagierung	Unterstützt das Taggen bei der Erstellung (AWS Cloud Map API AWS CLI, AWS SDK)
AWS Cloud Map Namespaces	Ja	Nein. Namespace-Tags werden nicht auf andere Ressourcen übertragen, die dem Namespace zugeordnet sind.	Ja
AWS Cloud Map Dienste	Ja	Nein. Service-Tags werden nicht auf andere Ressourcen übertragen, die mit dem Service verknüpft sind.	Ja

Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Maximale Anzahl von Tags für jede Ressource — 50
- Jeder Tag (Markierung) muss für jede Ressource eindeutig sein. Jeder Tag (Markierung) kann nur einen Wert haben.
- Maximale Schlüssellänge: 128 Unicode-Zeichen in UTF-8
- Maximale Wertlänge: 256 Unicode-Zeichen in UTF-8
- Wenn Ihr Tagging-Schema für mehrere AWS Dienste und Ressourcen verwendet wird, denken Sie daran, dass für andere Dienste möglicherweise Einschränkungen hinsichtlich der zulässigen Zeichen gelten. Allgemein erlaubte Zeichen sind Buchstaben, Zahlen, Leerzeichen, die in UTF-8 darstellbar sind, sowie die folgenden Zeichen: + - = . _ : / @.
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden.
- Verwenden `aws:` Sie weder für Schlüssel noch für Werte eine Kombination aus Groß- oder Kleinbuchstaben, z. B. ein Präfix, da es für AWS die Verwendung reserviert ist. `AWS:` Sie können keine Tag-Schlüssel oder -Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht auf Ihr tags-per-resource Limit angerechnet.

Tags für AWS Cloud Map Ressourcen werden aktualisiert

Verwenden Sie die folgenden AWS CLI Befehle oder AWS Cloud Map API-Operationen, um die Tags für Ihre Ressourcen hinzuzufügen, zu aktualisieren, aufzulisten und zu löschen.

Tagging-Unterstützung für Ressourcen AWS Cloud Map

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Fügen Sie einen oder mehrere Tags hinzu oder überschreiben Sie sie.	TagResource	tag-resource	Hinzufügen — Tag SDResource
Löschen Sie ein oder mehrere Tags.	UntagResource	untag-resource	Entfernen — SDResource Tag

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Listet Tags für eine Ressource auf	ListTagsForResource	list-tags-for-resource	Abrufen — SDResource Tag

Die folgenden Beispiele zeigen, wie man Tags an Ressourcen mithilfe der AWS CLI hinzufügt oder entfernt.

Beispiel 1: Markieren einer vorhandenen Ressource

Der folgende Befehl markiert eine vorhandene Ressource.

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

Beispiel 2: Entfernen der Markierung einer vorhandenen Ressource

Der folgende Befehl löscht ein Tag aus einer vorhandenen Ressource.

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

Beispiel 3: Tags für eine Ressource auflisten

Der folgende Befehl listet die Tags auf, die einer vorhandenen Ressource zugeordnet sind.

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

Mit einigen Aktionen zur Ressourcenerstellung können Sie Tags beim Erstellen der Ressource angeben. Die folgenden Aktionen unterstützen das Markieren bei der Erstellung.

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Erstellen eines HTTP-Namespace	CreateHttpNamespace	create-http-namespace	Neu — SDHttp Namespace
Erstellen eines privaten Namespace auf DNS-Basis	CreatePrivateDnsNamespace	create-private-dns-namespace	Neu- SDPrivate DnsNamespace

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Erstellen eines öffentlichen Namespace auf DNS-Basis	CreatePublicDnsNamespace	create-public-dns-namespace	Neu- SDPublicDnsNamespace
Einen Service erstellen	CreateService	create-service	Neu- SDService

AWS Cloud Map Servicekontingenten

AWS Cloud Map Ressourcen unterliegen den folgenden Servicekontingenten auf Kontoebene. Jedes aufgeführte Kontingent gilt für jede AWS Region, in der Sie Ressourcen erstellen AWS Cloud Map .

Name	Standard	Anpas	Beschreibung
Benutzerdefinierte Attribute pro Instance	Jede unterstützte Region: 30	Nein	Die maximale Anzahl der benutzerdefinierten Attribute, die Sie bei der Registrierung einer Instance angeben können.
DiscoverInstances Burst-Rate pro Konto	Jede unterstützte Region: 2.000	Ja	Die maximale Burst-Rate für den DiscoverInstances Aufrufvorgang von einem einzelnen Konto aus.
DiscoverInstances Betrieb pro Konto, konstante Rate	Jede unterstützte Region: 1 000	Ja	Die maximale konstante Rate für den DiscoverInstances Anrufbetrieb von einem einzigen Konto aus.
DiscoverInstancesRevision Tarif pro Konto	Jede unterstützte Region: 3 000	Ja	Die maximale Rate für Anrufe DiscoverInstancesRevision von einem einzigen Konto aus.
Instances pro Namespace	Jede unterstützte Region: 2 000	Ja	Die maximale Anzahl von Service-Instances, die Sie mit demselben Namespace registrieren können.

Name	Standard	Anpas	Beschreibung
Instances pro Service	Jede unterstützte Region: 1 000	Nein	Die maximale Anzahl der Instances, die Sie mit demselben Service in einer Region registrieren können.
Namespaces pro Region	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von Namespaces, die Sie pro Region erstellen können.

* Wenn Sie einen Namespace erstellen, erstellen wir automatisch eine von Amazon Route 53 gehostete Zone. Diese gehostete Zone wird auf das Kontingent für die Anzahl der Hosting-Zonen angerechnet, die Sie mit einem AWS Konto erstellen können. Weitere Informationen finden Sie unter Kontingente für gehostete Zonen im Amazon Route 53-Entwicklerhandbuch.

** Die Erhöhung der Instances für DNS-Namespaces für AWS Cloud Map erfordert eine Erhöhung des Route-53-Limits für Datensätze pro gehosteter Zone, was zusätzliche Gebühren verursacht.

Verwaltung Ihrer Servicekontingenten AWS Cloud Map

AWS Cloud Map ist in Service Quotas integriert, einen AWS Dienst, mit dem Sie Ihre Kontingente von einem zentralen Ort aus einsehen und verwalten können. Weitere Informationen zu Service Quotas finden Sie unter Was sind Service Quotas? im Benutzerhandbuch für Service Quotas.

Mit Service Quotas können Sie ganz einfach den Wert Ihrer AWS Cloud Map Servicekontingenten nachschlagen.

AWS Management Console

Um AWS Cloud Map Servicekontingenten anzuzeigen, verwenden Sie den AWS Management Console

1. Öffnen Sie die Service Quotas-Konsole unter https://console.aws.amazon.com/servicequotas/.
2. Wählen Sie im Navigationsbereich AWS -Services.
3. Suchen Sie in der Liste der AWS -Services nach AWS Cloud Map und wählen Sie es aus.

4. In der Liste der Dienstkontingente für AWS Cloud Map finden Sie den Namen des Servicekontingents, den angewendeten Wert (falls verfügbar), das AWS Standardkontingent und ob der Kontingentwert anpassbar ist.

   Um zusätzliche Informationen zu einem Servicekontingent, wie z. B. die Beschreibung, anzuzeigen, wählen Sie den Kontingentnamen aus, um die Kontingentdetails aufzurufen.

5. (Optional) Um eine Kontingenterhöhung zu beantragen, wählen Sie das Kontingent aus, das Sie erhöhen möchten, und wählen Sie Erhöhung auf Kontoebene beantragen.

Weitere Informationen zum Umgang mit Servicekontingenten AWS Management Console finden Sie im Service Quotas User Guide.

AWS CLI

Zur Anzeige von AWS Cloud Map Servicekontingenten verwenden Sie den AWS CLI

Führen Sie den folgenden Befehl aus, um die AWS Cloud Map Standardkontingente anzuzeigen.

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

Führen Sie den folgenden Befehl aus, um Ihre angewendeten AWS Cloud Map Kontingente anzuzeigen.

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

Weitere Informationen zum Arbeiten mit Service Quotas mithilfe von finden Sie in der AWS CLIAWS CLI Befehlsreferenz für Dienstkontingente. Informationen zum Beantragen einer Kontingenterhöhung finden Sie unter dem `request-service-quota-increase`-Befehl in der AWS CLI -Befehlsreferenz.

Behandeln Sie die Drosselung von AWS Cloud Map DiscoverInstances API-Anfragen

AWS Cloud Map drosselt DiscoverInstancesAPI-Anfragen für jedes AWS Konto pro Region. Die Drosselung trägt dazu bei, die Leistung des Dienstes zu verbessern und eine faire Nutzung für alle Kunden zu gewährleisten. AWS Cloud Map Durch die Drosselung wird sichergestellt, dass API-Aufrufe die maximal zulässigen AWS Cloud Map DiscoverInstancesDiscoverInstancesAPI-Anforderungsquoten nicht überschreiten. DiscoverInstances API-Aufrufe, die aus einer der folgenden Quellen stammen, unterliegen den Anforderungsquoten:

- Eine Drittanbieteranwendung
- Ein Befehlszeilentool
- Die AWS Cloud Map Konsole

Wenn Sie ein API-Drosselungskontingent überschreiten, erhalten Sie den `RequestLimitExceeded` Fehlercode. Weitere Informationen finden Sie unter the section called “Anforderungsratenbegrenzung”.

Wie wird die Drosselung angewendet

AWS Cloud Map verwendet den Token-Bucket-Algorithmus, um die API-Drosselung zu implementieren. Mit diesem Algorithmus verfügt Ihr Konto über einen Bucket, der eine bestimmte Anzahl von Token enthält. Die Anzahl der Token im Bucket entspricht Ihrer Drosselungsquote zu einer bestimmten Sekunde. Es gibt einen Bucket für eine einzelne Region, und dieser gilt für alle Endpunkte in der Region.

Anforderungsratenbegrenzung

Durch die Drosselung wird die Anzahl der DiscoverInstancesAPI-Anfragen begrenzt, die Sie stellen können. Jede Anfrage entfernt ein Token aus dem Bucket. Die Bucket-Größe für den DiscoverInstancesAPI-Vorgang beträgt beispielsweise 2.000 Token, sodass Sie in einer Sekunde bis zu 2.000 DiscoverInstancesAnfragen stellen können. Wenn Sie 2.000 Anfragen in einer Sekunde überschreiten, werden Sie gedrosselt und die verbleibenden Anfragen innerhalb dieser Sekunde schlagen fehl.

Buckets werden automatisch mit einer festgelegten Geschwindigkeit wieder aufgefüllt. Wenn der Bucket nicht voll ausgelastet ist, wird jede Sekunde eine festgelegte Anzahl von Tokens hinzugefügt,

bis der Bucket seine Kapazität erreicht hat. Wenn der Bucket beim Eintreffen der Nachfüll-Token voll ausgelastet ist, werden diese Token verworfen. Die Bucket-Größe für den DiscoverInstancesAPI-Vorgang beträgt 2.000 Token, und die Nachfüllrate beträgt 1.000 Token pro Sekunde. Wenn Sie 2.000 DiscoverInstancesAPI-Anfragen in einer Sekunde stellen, wird der Bucket sofort auf null (0) Token reduziert. Der Bucket wird dann jede Sekunde mit bis zu 1.000 Token aufgefüllt, bis er seine maximale Kapazität von 2.000 Token erreicht hat.

Sie können Tokens verwenden, wenn sie dem Bucket hinzugefügt werden. Sie müssen nicht warten, bis der Bucket seine maximale Kapazität erreicht hat, bevor Sie API-Anfragen stellen. Wenn Sie den Bucket leeren, indem Sie 2.000 DiscoverInstancesAPI-Anfragen in einer Sekunde stellen, können Sie danach immer noch bis zu 1.000 DiscoverInstancesAPI-Anfragen pro Sekunde stellen, solange Sie dies benötigen. Das bedeutet, dass Sie die Nachfüll-Token sofort verwenden können, sobald sie Ihrem Bucket hinzugefügt werden. Der Bucket beginnt erst dann, sich bis zur maximalen Kapazität aufzufüllen, wenn Sie pro Sekunde weniger API-Anfragen stellen als die Nachfüllrate.

Wiederholversuche oder Stapelverarbeitung

Wenn eine API-Anfrage fehlschlägt, muss Ihre Anwendung die Anfrage möglicherweise erneut versuchen. Verwenden Sie ein angemessenes Schlafintervall zwischen aufeinanderfolgenden Anfragen, um die Anzahl der API-Anfragen zu reduzieren. Um die besten Ergebnisse zu erzielen, verwenden Sie ein zunehmendes oder variables Energiesparintervall.

Berechnen des Energiesparintervalls

Wenn Sie eine API-Anforderung abrufen oder wiederholen müssen, empfehlen wir die Verwendung eines exponentiellen Backoff-Algorithmus zum Berechnen des Energiesparintervalls zwischen API-Aufrufen. Indem Sie bei aufeinanderfolgenden Fehlerantworten immer längere Wartezeiten zwischen Wiederholungsversuchen verwenden, können Sie die Anzahl der fehlgeschlagenen Anfragen reduzieren. Weitere Informationen und Implementierungsbeispiele für diesen Algorithmus finden Sie unter Verhalten bei Wiederholungsversuchen im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch.

Anpassung der API-Drosselungsquoten

Sie können eine Erhöhung der API-Drosselungsquoten für Ihr Konto beantragen. AWS Um eine Kontingentanpassung anzufordern, kontaktieren Sie das AWS -Support -Center.

Dokumentenverlauf für AWS Cloud Map

In der folgenden Tabelle werden die wichtigsten Updates und neuen Funktionen des AWS Cloud Map Developer Guide beschrieben. Wir aktualisieren die Dokumentation regelmäßig, um das Feedback, das Sie uns senden, einzuarbeiten.

Änderung	Beschreibung	Datum
AWS Cloud Map Dienstattribute	Sie können jetzt Attribute auf Serviceebene angeben, um zu vermeiden, dass Attribute in allen Instanzen, die für einen Service registriert sind, dupliziert werden. Sie können diese Attribute für die komplexe Weiterleitung des Datenverkehrs, die Festlegung von Timeout- und Wiederholungswerten sowie für die Koordination zwischen Diensten und externen Integrationen verwenden.	13. Dezember 2024
Tutorials hinzugefügt	Zwei Tutorials mit häufigen Anwendungsfällen für die Verwendung AWS Cloud Map wurden hinzugefügt.	27. März 2024
CloudTrail Die Integrationsdokumentation wurde aktualisiert	Die Dokumentation, die die AWS Cloud Map Integration mit CloudTrail der Protokollierung von API-Aktivitäten beschreibt, wurde aktualisiert.	20. März 2024
Verwaltete Richtlinienaktualisierungen	`AWSCloudMapDiscoverInstance Access AWSCloudM`	20. September 2023

	`apRegisterInstanceAccess` , und `AWSCloudMapReadOnlyAccess` die Richtlinien wurden aktualisiert.	
Cloud Map und AWS PrivateLink	Sie können jetzt eine verwenden AWS PrivateLink , um eine private Verbindung zwischen Ihrer VPC und AWS Cloud Map herzustellen.	15. September 2023
Aktualisierung der verwalteten Richtlinien	`AWSCloudMapDiscoverInstanceAccess` Die Richtlinie wurde aktualisiert.	15. August 2023
AWS SDK für Python	Python-Befehlszeilenbeispiele hinzugefügt.	13. September 2022
IPv6 Unterstützung	API-Endpunkte sind jetzt `IPv6` nur in Netzwerken verfügbar.	28. Januar 2022
Erkennung von Dienstinstanzen	AWS Cloud Map Unterstützung für die Erstellung von Diensten in einem Namespace hinzugefügt, der DNS-Abfragen unterstützt, die nur mithilfe der DiscoverInstancesAPI-Operation und nicht mithilfe von DNS-Abfragen auffindbar sind.	24. März 2021
Ressourcen-Markierung	AWS Cloud Map Unterstützung für das Hinzufügen von Metadaten-Tags zu Ihren Namespaces und Diensten mithilfe von hinzugefügt. AWS Management Console	8. Februar 2021

Ressourcen-Markierung	AWS Cloud Map Unterstützung für das Hinzufügen von Metadaten-Tags zu Ihren Namespaces und Diensten mithilfe von und hinzugefügt. AWS CLI APIs	22. Juni 2020
Erste Veröffentlichung	Dies ist die erste Version des AWS Cloud Map Developer Guide.	28. November 2018

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.