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 den Back-End-Diensten und -Ressourcen, von denen Ihre Anwendungen abhängen, logische Namen zuordnen können. Sie hilft Ihren Anwendungen auch dabei, Ressourcen mithilfe eines der AWS SDKs, RESTful-API-Aufrufe 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— In den Verfahren in diesem Handbuch wird erklärt, wie Sie AWS Management Console mit dem Aufgaben ausführen können.
- AWS SDKs — Wenn Sie eine Programmiersprache verwenden, die ein SDK für AWS bereitstellt, können Sie ein SDK für den Zugriff AWS Cloud Map verwenden. SDKs vereinfachen die Authentifizierung, lassen sich leicht in die Entwicklungsumgebung integrieren und bieten einen einfachen 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 for Windows 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-Clientunterstützung — AWS Cloud Map Ab dem 22. Juni 2023 werden in allen neuen Regionen alle Befehle, an die `IPv6` Clients gesendet werden, an einen neuen Dualstack-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:

- AWS Identity and Access Management in 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 Serviceregistrierung 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 -Preisgestaltung.

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
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

Zur Verwendung eingerichtet AWS Cloud Map

Die Übersicht und die Verfahren in diesem Abschnitt 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 SDKs zu AWS
- Richten Sie das oder ein AWS Command Line InterfaceAWS 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/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und 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 jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

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 SDKs zu AWS

Um die API, die AWS CLI AWS Tools for Windows PowerShell, oder die AWS SDKs verwenden zu können, 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 programmatischen Zugriff, wenn sie mit AWS außerhalb des interagieren möchten. AWS Management Console 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 die AWS CLI, AWS SDKs oder APIs zu signieren. AWS	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zu AWS IAM Identity Center verwendenden im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs, Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch für AWS SDKs und Tools.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an	Folgen Sie den Anweisungen unter Verwenden temporäre r Anmeldeinformationen mit

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
	die AWS CLI, AWS SDKs oder APIs zu signieren. AWS	AWS Ressourcen im IAM-Benutzerhandbuch.
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu finden Sie unter Authentifizierung mithilfe von IAM-Benutzeranmeldedaten im Benutzerhandbuch. AWS CLIAWS Command Line Interface • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch für AWS SDKs und Tools. • Informationen zu AWS APIs finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch.

Richten Sie das oder ein AWS Command Line InterfaceAWS Tools for Windows PowerShell

Das AWS Command Line Interface (AWS CLI) ist ein einheitliches Tool für die Verwaltung von AWS Diensten. Informationen zur Installation und Konfiguration von finden Sie unter Getting Setup with the AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch. 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 for Windows 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

Dieses 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**
>
> Für die Zwecke dieses Tutorials dienen die Ressourcendetails, wie Domainnamen und IP-Adressen, nur zu Simulationszwecken. Sie können nicht über das Internet gelöst werden.

Voraussetzungen

Die folgenden Voraussetzungen müssen erfüllt sein, um dieses Tutorial erfolgreich abzuschließen.

- 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. Weitere Informationen zu diesem `dig` Befehl finden Sie unter dig — DNS lookup utility.

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 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 `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 der 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 `192.168.2.1` Sie für die IPv4-Adresse an.

   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 zurück zur Namespace-Detailseite zu navigieren.
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 Route 53-Konsole unter https://console.aws.amazon.com/route53/ an AWS Management Console und öffnen Sie sie.
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 Dienstinstanz ab.

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

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

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

5. Fragen Sie mit dem AWS CLI die 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.

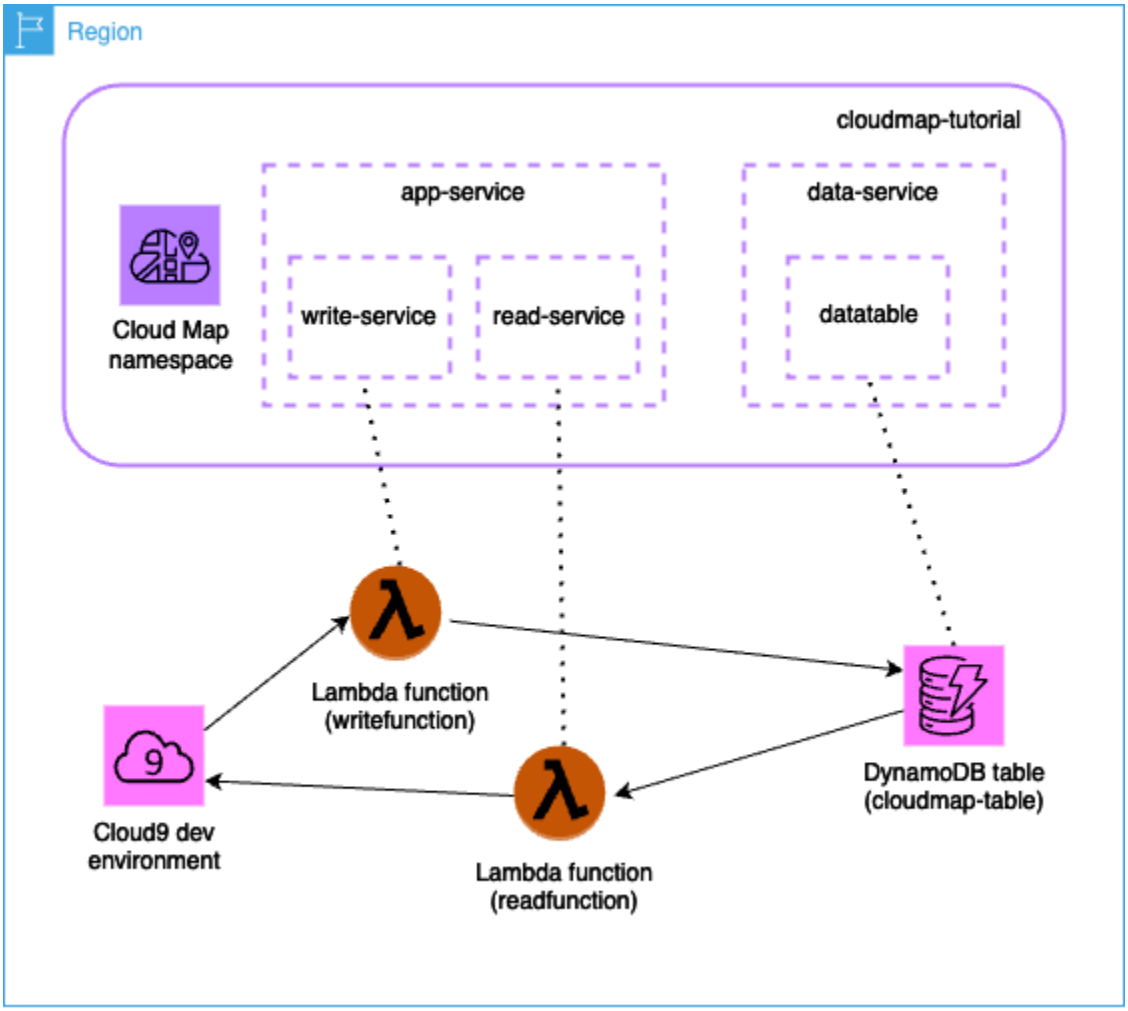
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Cloud Map Konsole unter https://console.aws.amazon.com/cloudmap/.
2. Wählen Sie aus der Liste der Namespaces den **`cloudmap-tutorial.com`** Namespace aus und klicken Sie auf Details anzeigen.
3. Wählen Sie auf der Seite mit 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 benutzerdefinierten Attributen verwenden

Dieses Tutorial zeigt, wie Sie AWS Cloud Map Service Discovery mit benutzerdefinierten Attributen verwenden können, die über die API auffindbar sind. AWS Cloud Map Dieses Tutorial führt Sie durch die Erstellung einer Client-Anwendung in einer AWS Cloud9 Umgebung, die zwei Lambda-Funktionen verwendet, 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 der Client-Anwendung und den Lambda-Funktionen verwendet AWS Cloud Map benutzerdefinierte Attribute, um die Ressourcen zu ermitteln, die für die Ausführung des Jobs benötigt werden.

Das folgende Diagramm zeigt die allgemeine Architektur, die in diesem Tutorial verwendet wird.



> **Important**
>
> Sie werden während des Workshops 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. In diesem Tutorial werden die in diesem Namespace erstellten Ressourcen mit AWS Cloud Map API-Aufrufen unter Verwendung benutzerdefinierter Attribute auffindbar sein. In einem späteren Schritt erfahren Sie mehr darüber.

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 `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: DynamoDB-Tabelle erstellen

In diesem Schritt erstellen Sie eine DynamoDB-Tabelle, die zum Speichern und Abrufen von Daten für die später in diesem Tutorial erstellte Beispielanwendung verwendet wird.

Informationen zum Erstellen einer DynamoDB finden Sie unter Schritt 1: Tabelle erstellen 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 den AWS Cloud Map Datendienst

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 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 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 die folgenden Schlüssel-Wert-Paare an.
      - Schlüssel =**`name`**, Wert = `datatable`
      - Schlüssel =`tablename`, Wert = `cloudmap`
   d. Stellen Sie sicher, dass die Attribute mit der Abbildung unten übereinstimmen, und wählen Sie Dienstinstanz registrieren aus.

Schritt 4: Erstellen Sie eine AWS Lambda Ausführungsrolle

In diesem Schritt erstellen Sie eine IAM-Rolle, die von der AWS Lambda Funktion, die wir im nächsten Schritt erstellen, verwendet wird. Sie können die Rolle benennen `cloudmap-role` und die Rechtegrenze weglassen, da diese IAM-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
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 widerspiegelt, 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.

```
import json
import boto3
import random

def lambda_handler(event, context):

    serviceclient = boto3.client('servicediscovery')

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

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

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

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

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

Schritt 6: Erstellen Sie den App-Dienst AWS Cloud Map

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 =**name**, Wert = `writeservice`
- Schlüssel =`function`, Wert = `writefunction`

d. Stellen Sie sicher, dass die Attribute mit der Abbildung unten übereinstimmen, und wählen Sie Dienstinstanz registrieren aus.

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 widerspiegelt, und stellen Sie dann die Funktion bereit.

```
import json
import boto3

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

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

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

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

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

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

Schritt 8: Erstellen Sie eine 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.

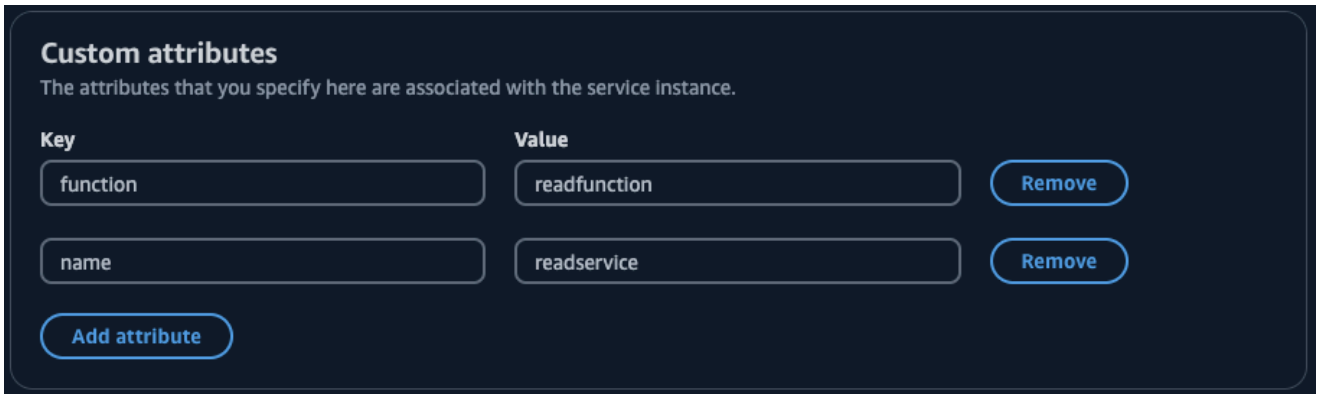
a. Wählen Sie als Instanztyp die Option Identifizierungsinformationen für eine andere Ressource aus.

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 =**`name`**, Wert = `readservice`
- Schlüssel =`function`, Wert = `readfunction`

d. Stellen Sie sicher, dass die Attribute mit der Abbildung unten übereinstimmen, und wählen Sie Dienstinstanz registrieren aus.



Schritt 9: Erstellen Sie eine Entwicklungsumgebung

AWS Cloud9 ist eine integrierte Entwicklungsumgebung (IDE), die von verwaltet wird AWS. Die AWS Cloud9 IDE stellt die Software und Tools bereit, die für die dynamische Programmierung benötigt werden. In diesem Schritt erstellen wir eine AWS Cloud9 Umgebung und konfigurieren sie, mit der Sie mit AWS SDK for Python (Boto3) der API programmieren werden. AWS

Informationen zum Erstellen einer AWS Cloud9 Umgebung finden Sie unter Erstellen einer EC2-Umgebung im AWS Cloud9 Benutzerhandbuch. Anhand der folgenden Tabelle können Sie ermitteln, welche Optionen angegeben oder ausgewählt werden müssen.

Option	Wert	
Name	Cloudmap-Tutorial	
Umgebungstyp	Neue EC2-Instanz	

Option	Wert
Instance-Typ	t2.micro
Plattform	Ubuntu Server 22.04 LTS

Lassen Sie die restlichen Standardauswahlen unverändert. Erstellen Sie die Umgebung und öffnen Sie sie dann in AWS Cloud9. Dies bietet Ihnen eine Bash-Shell, mit der Sie arbeiten können.

> **Important**
>
> Wenn Sie Probleme beim Öffnen Ihrer AWS Cloud9 Umgebung haben, finden Sie AWS Cloud9 weitere Informationen unter Problembehandlung: Umgebung kann nicht geöffnet werden im AWS Cloud9 Benutzerhandbuch.

Führen Sie mithilfe der Bash-Shell die folgenden Befehle aus, um die Umgebung zu konfigurieren.

1. Aktualisieren Sie die Umgebung.

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

2. Stellen Sie sicher, dass `python3` es installiert ist.

   ```
   python3 --version
   ```

3. Installieren Sie das Boto3-Paket in der Umgebung.

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

Schritt 10: Erstellen Sie einen Frontend-Client

Mithilfe der im vorherigen Schritt erstellten AWS Cloud9 Entwicklungsumgebung erstellen Sie einen Frontend-Client, der Code verwendet, der die von Ihnen konfigurierten Dienste erkennt AWS Cloud Map und diese Dienste aufruft.

1. Wählen Sie in der AWS Cloud9 Umgebung im Menü Datei die Option Neue Datei aus. Dadurch wird eine Datei mit dem Namen erstellt`Untitled1`.

2. Kopieren Sie den folgenden Code und fügen Sie ihn in die `Untitled1` Datei ein. Dieser Code erkennt die Lambda-Funktion zum Schreiben von Daten, indem er `name=writeservice` im Dienst nach dem benutzerdefinierten Attribut sucht. `app-service` Es wird der Name der Lambda-Funktion zurückgegeben, die für das Schreiben von Daten in die DynamoDB-Tabelle verantwortlich ist. Dann wird die Lambda-Funktion aufgerufen und eine Beispielnutzlast übergeben.

```
import boto3

serviceclient = boto3.client('servicediscovery')

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

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

lambdaclient = boto3.client('lambda')

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

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

3. Wählen Sie im Menü Datei die Option Speichern unter... und speichern Sie die Datei unter`writeclient.py`.
4. Verwenden Sie in der Bash-Shell in Ihrer AWS Cloud9 Umgebung den folgenden Befehl, um den Python-Code auszuführen.

```
python3 writeclient.py
```

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

```
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}}"}'
```

> **Note**
>
> Wenn es sich bei der Ausgabe um eine Fehlermeldung handelt, die besagt, dass für die Aufgabe ein Timeout aufgetreten ist, aktualisieren Sie den `writefunction` Timeout-Wert der Lambda-Funktion. Weitere Informationen finden Sie unter Configure Lambda function timeout im AWS Lambda Developer Guide.

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

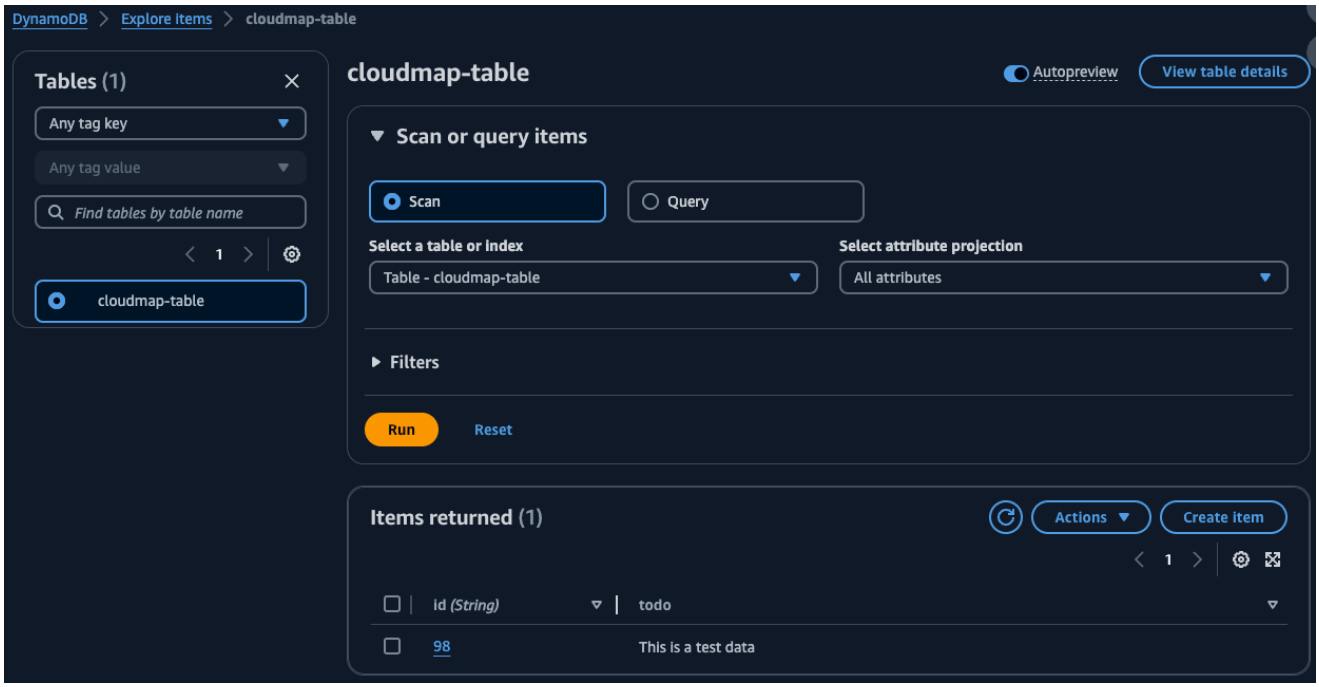
   a. Melden Sie sich bei der DynamoDB-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/dynamodb/.

   b. Wählen Sie im linken Navigationsbereich Tables (Tabellen) aus.

   c. Wählen Sie aus der Tabellenliste Ihre Cloudmap-Tabelle aus und wählen Sie im Menü Aktionen die Option Elemente durchsuchen aus.

   d. Notieren Sie sich im Abschnitt Zurückgegebene Artikel den numerischen Wert in der Spalte ID (Zeichenfolge).

      Im Folgenden wird ein Beispiel gezeigt, in dem sich der Wert id (String) befindet98.



   e. Wählen Sie in der AWS Cloud9 Umgebung im Menü Datei die Option Neue Datei, wodurch eine Datei mit dem Namen erstellt wird. `Untitled1`

f. Kopieren Sie den folgenden Code und fügen Sie ihn in die `Untitled1` Datei ein. Ersetzen Sie den `Payload` Wert durch den `id (String)` Wert aus Ihrer DynamoDB-Tabelle im vorherigen Schritt. Dieser Code liest aus der 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={ 'name': 'readservice' })

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

lambdaclient = boto3.client('lambda')

resp = lambdaclient.invoke(FunctionName=functionname,
 InvocationType='RequestResponse', Payload='"98"')

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

g. Wählen Sie im Menü Datei die Option Speichern unter... und speichern Sie die Datei unter`readclient.py`.

h. Verwenden Sie in der Bash-Shell in Ihrer AWS Cloud9 Umgebung den folgenden Befehl, um den Python-Code auszuführen.

```
python3 readclient.py
```

Die Ausgabe sollte in etwa folgendermaßen aussehen:

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

> **Note**
>
> Wenn es sich bei der Ausgabe um eine Fehlermeldung handelt, die besagt, dass für die Aufgabe ein Timeout aufgetreten ist, aktualisieren Sie den `readfunction` Timeout-Wert der Lambda-Funktion. Weitere Informationen finden Sie unter Configure Lambda function timeout im AWS Lambda Developer Guide.

Schritt 11: Bereinigen Sie die Ressourcen

Sobald 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 in diesem 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 8: (Optional) Ressourcen bereinigen im Amazon DynamoDB DynamoDB-Entwicklerhandbuch
Lambda-Funktionen und zugehörige IAM-Ausführungsrolle	Im Developer Guide finden Sie Ordnung AWS Lambda
AWS Cloud9 Umgebung	Löschen einer Umgebung AWS Cloud9 im AWS Cloud9 Benutzerhandbuch.

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 VPCs zugeordnet werden. Weitere Informationen finden Sie unter AssociateVPC WithHostedZone 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 von erstellte gehostete Zone AWS Cloud Map hat denselben*	• DiscoverInstances • CreatePrivateDnsNamespace

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
		Namen wie der Namespace und enthält DNS-Einträge mit Namen im Format Dienstname. 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	

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
		Route 53 auf die Anfrage 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 Format Dienstname. Namespace-Name.`* **Note** Der Namespace-Name muss in diesem Fall ein Domainnam	• DiscoverInstances • CreatePublicDnsNamespace

Namespace-Typ	Methode zur Erkennung von Instanzen	Funktionsweise	Zusätzliche Informationen
		e sein, den 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 als 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. z. B. `.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 VPCs und API-Aufrufen und öffentlichen DNS-Abfragen wählen, 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 VPCs wählen, 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 Wert Time to Live (TTL) 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 *`roten`* 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 (ersetzen Sie die *roten* 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 AWS Management Console und öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudmap/. AWS Cloud Map
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 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 einen Namespace erstellen und angeben, dass Sie Service-Instances entweder mithilfe von öffentlichen DNS-Abfragen oder DNS-Abfragen in VPCs erkennen möchten, AWS Cloud Map wird eine öffentliche oder private gehostete Zone von Amazon Route 53 erstellt. Wenn Sie den Namespace löschen, wird die entsprechende gehostete AWS Cloud Map Zone gelöscht.

Bevor Sie einen Namespace löschen, müssen Sie alle Dienstinstanzen deregistrieren und anschließend alle Dienste löschen, die im Namespace erstellt wurden. Weitere Informationen finden Sie unter Abmeldung einer Dienstinstanz AWS Cloud Map und Einen 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 https://console.aws.amazon.com/cloudmap/ AWS Cloud Map .
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 *roten* 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 *roten* 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 DiscoverInstancesAPI-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.

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

Themen

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 Integritätsprüfung dem Route 53-Datensatz zu, AWS Cloud Map der AWS Cloud Map erstellt wird, wenn Sie eine Instanz registrieren. Wenn Sie in der DNS-Konfiguration eines Dienstes `A` sowohl `AAAA` Eintragstypen als auch Eintragstypen angeben, AWS Cloud Map wird eine Integritätsprüfung erstellt, bei der die IPv4-Adresse verwendet wird, um den Zustand der Ressource zu überprüfen. 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 beispielsweise 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 bestätigt`HEALTHY`, 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 im 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 zurück, z. B. 2001:0 db 8:85 a 3:0000:0000:abcd: 0001:2345.

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 Instance Werte wie eine IPv4-Adresse, eine IPv6-Adresse oder beides angeben, AWS Cloud Map werden automatisch A - und/oder AAAA-Einträge erstellt, die denselben Namen haben wie der Wert von **`service-hostname`** im SRV-Datensatz.

Sie können Datensatztypen in den folgenden Kombinationen angeben:

- A
- AAAA
- A und AAAA
- CNAME
- SRV

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

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 Seite Namespace: ***`Namespace-Name`*** die Option Create service (Service 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 erfordert (z. B. HAProxy), geben Sie Folgendes für den Dienstnamen an:

- Beginnen Sie den Namen mit einem Unterstrich (_), zum Beispiel _exampleservice.
- *Beenden Sie den Namen mit.* _protocol, zum Beispiel. _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 den TTL-Wert (Time to Live) 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 *roten* 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 *roten* 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-
xxxxxxxxxx',
        'CreateDate': 1587081768.334,
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 60,
                    'Type': 'A',
                },
            ],
            'NamespaceId': 'ns-xxxxxxxxxx',
            'RoutingPolicy': 'MULTIVALUE',
        },
        'Id': 'srv-xxxxxxxxxx',
        'Name': 'service-name',
        'NamespaceId': 'ns-xxxxxxxxxx',
    },
    '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.

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 wird.
4. Wählen Sie auf der Seite Namespace: ***Namespace-Name den Dienst aus, den Sie bearbeiten*** möchten, und klicken Sie auf Details anzeigen.
5. Wählen Sie auf der Seite Service: ***Dienstname*** 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 Seite Service: ***Dienstname*** 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. I
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 ***roten*** 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 *roten* 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 einsehen 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 *`roten`* Wert 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 `Boto3` hier Anweisungen zur Installation, Konfiguration und Verwendung.
2. Importieren `Boto3` und `servicediscovery` als 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-xxxxxxxxxxxxxxx',
            'Name': 'myservice',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Einen 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 Dienst registrierten Instanzen deregistriert haben, führen Sie das folgende Verfahren aus, um den Dienst 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 Seite Namespace: ***Namespace-Name*** die Option für den Service aus, den Sie löschen möchten.
5. Wählen Sie Löschen aus.
6. Bestätigen Sie, dass Sie den Service löschen möchten.

AWS CLI

- Löschen Sie einen Dienst mit dem `delete-service` Befehl (ersetzen Sie den *roten* 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 *roten* 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:

- Amazon EC2-Instances
- Amazon-DynamoDB-Tabellen
- Amazon-S3-Buckets
- Amazon-Simple-Queue-Service-(Amazon-SQS)-Warteschlangen
- APIs, die auf Amazon API Gateway bereitgestellt werden

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 RegisterInstancein 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 RegisterInstancein der AWS Cloud Map API-Referenz.

Gehen Sie folgendermaßen vor, um eine Instanz 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 Seite Namespace: ***`Namespace-Name`*** den Service aus, den Sie verwenden möchten.
5. Wählen Sie auf der Seite Service: ***`Service-Name`*** die Option Register service instance (Service-Instance 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 Amazon EC2 EC2-Instance-ID oder andere identifizierende Informationen für eine Ressource angeben möchten, die keine IP-Adresse hat.

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

7. Geben Sie als Service-Instance-ID einen Bezeichner an, der einer vorhandenen Dienstinstanz zugeordnet ist. Dieses Feld ist nur erforderlich, wenn Sie die Werte einer vorhandenen Instanz aktualisieren möchten, indem Sie sie erneut registrieren.
8. Führen Sie je nach Wahl des Instanztyps die folgenden Schritte aus.

Instance-Typ	Schritte
IP-Adresse	a. Geben Sie unter Standardattribute für IPv4-Adresse gegebenenfalls eine IPv4-Adresse an, über die Ihre Anwendung auf die Ressource zugreifen kann, die dieser Dienstinstanz zugeordnet ist. b. Geben Sie für die IPv6-Adresse gegebenenfalls eine IPv6-IP-Adresse an, über die Ihre Anwendungen 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 zuzugreifen, die dieser Dienstins

Instance-Typ	Schritte	
	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.	
EC2-Instance	a. Wählen Sie für EC2-Instance-ID die ID der Amazon EC2 EC2-Insta nce aus, die Sie als AWS Cloud Map Service-I nstance 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 Amazon EC2 EC2-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 Zustandsprüfungen sind jedem der neuen oder aktualisierten Datensätze zugeordnet.

  Registrieren Sie eine Dienstinstanz mit dem `register-instance` Befehl (ersetzen Sie die *`roten`* 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 Zustandsprüfungen sind jedem der neuen oder aktualisierten Datensätze zugeordnet.

Registrieren Sie eine Dienstinstanz mit `register_instance()` (ersetzen Sie die *roten* 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 *roten* 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. Dienstinstanzen auflisten mit `list_instances()` (ersetzen Sie den *roten* Wert durch Ihren 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 Seite Namespace: ***Namespace-Name*** den Service aus, den Sie für die Registrierung der Service-Instance verwendet haben.
5. Wählen Sie auf der Seite Service: ***Service-Name*** den Namen der Service-Instance aus, den Sie aktualisieren möchten.
6. Wählen Sie im Abschnitt Custom attributes (Benutzerdefinierte Attribute) die Option Edit (Bearbeiten) aus.
7. Fügen Sie auf der Seite ***Service-Instance bearbeiten: Instance-Name*** 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 Seite Namespace: ***Namespace-Name*** den Dienst aus, mit dem Sie die Dienstinstanz registriert haben.

5. Wählen Sie auf der Seite Service: ***Dienstname*** 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 *roten* 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 Ihren Service verwenden.

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

3. Deregistrieren Sie eine Dienstinstanz mit `deregister-instance()` (ersetzen Sie die *roten* 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.

Diese 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

AWS Identity and Access Management in AWS Cloud Map

Um Aktionen an AWS Cloud Map Ressourcen durchzuführen, wie z. B. die Registrierung einer Domain oder die Aktualisierung eines Eintrags, müssen Sie AWS Identity and Access Management

(IAM) authentifizieren, dass Sie ein zugelassener AWS Benutzer sind. Wenn Sie die AWS Cloud Map Konsole verwenden, authentifizieren Sie Ihre Identität, indem Sie Ihren AWS Benutzernamen und ein Passwort angeben. Wenn Sie AWS Cloud Map programmgesteuert zugreifen, authentifiziert Ihre Anwendung Ihre Identität für Sie, indem sie Zugriffsschlüssel verwendet oder Anfragen signiert.

Nachdem Sie Ihre Identität authentifiziert haben, kontrolliert IAM Ihren Zugriff auf, AWS indem es überprüft, ob Sie berechtigt sind, Aktionen durchzuführen und auf Ressourcen zuzugreifen. Wenn Sie ein Kontoadministrator sind, können Sie mithilfe von IAM den Zugriff anderer Benutzer auf die mit Ihrem Konto verknüpften Ressourcen steuern.

In diesem Kapitel wird erklärt, wie Sie IAM verwenden und wie Sie Ihre Ressourcen AWS Cloud Map schützen können.

Topics

Authentifizierung

Sie können auf eine der folgenden Arten zugreifen AWS :

- Root-Benutzer des AWS-Kontos— Wenn Sie zum ersten Mal ein AWS Konto erstellen, beginnen Sie mit einer einzigen Anmeldeidentität, die vollständigen Zugriff auf alle AWS Dienste und Ressourcen im Konto hat. Diese Identität wird als Root-Benutzer des AWS-Kontos bezeichnet. Um darauf zuzugreifen, müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, das zur Erstellung des Kontos verwendet wurde. Wenn Sie ein Konto erstellen AWS-Konto, beginnen Sie mit einer einzigen 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 und verwenden Sie diese, 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.
- IAM-Benutzer — Ein IAM-Benutzer ist eine Identität in Ihrem AWS Konto, die über bestimmte benutzerdefinierte Berechtigungen verfügt (z. B. Berechtigungen zum Erstellen eines HTTP-Namespace in). AWS Cloud MapSie können Ihre IAM-Anmeldeinformationen verwenden, um AWS

Webseiten wie The, Re:Post oder das AWS Management ConsoleCenter zu schützen.AWSAWS Support

Außer Anmeldeinformationen können Sie Zugriffsschlüssel für jeden Benutzer erstellen. Sie können diese Schlüssel verwenden, wenn Sie programmgesteuert auf AWS Dienste zugreifen, entweder über eines der verschiedenen SDKs oder mithilfe von. AWS Command Line Interface Das SDK und die CLI-Tools verwenden die Zugriffsschlüssel, um Ihre Anfrage verschlüsselt zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie die Anfrage selbst signieren. AWS Cloud Map unterstützt Signature Version 4, ein Protokoll zur Authentifizierung eingehender API-Anfragen. Weitere Informationen zur Authentifizierung von Anfragen finden Sie im Benutzerhandbuch unter Signieren von AWS API-Anfragen.AWS Identity and Access Management

- IAM-Rolle – Eine IAM-Rolle ist eine IAM-Identität, die Sie in Ihrem Konto mit bestimmten Berechtigungen erstellen können. Eine IAM-Rolle ähnelt einem IAM-Benutzer insofern, als es sich um eine AWS Identität mit Berechtigungsrichtlinien handelt, die festlegen, was die Identität tun kann und was nicht. AWS Eine Rolle ist jedoch nicht einer einzigen Person zugeordnet, sondern kann von allen Personen angenommen werden, die diese Rolle benötigen. Einer Rolle sind außerdem keine standardmäßigen, langfristigen Anmeldeinformationen (Passwörter oder Zugriffsschlüssel) zugeordnet. Wenn Sie eine Rolle annehmen, erhalten Sie stattdessen temporäre Anmeldeinformationen für Ihre Rollensitzung. IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

  - Föderierter Benutzerzugriff — Anstatt einen IAM-Benutzer zu erstellen, können Sie vorhandene Benutzeridentitäten aus AWS Directory Service Ihrem Unternehmensbenutzerverzeichnis oder einem Web-Identitätsanbieter verwenden. Diese werden als Verbundbenutzer bezeichnet. AWS weist einem Verbundbenutzer eine Rolle zu, wenn der Zugriff über einen Identitätsanbieter angefordert wird. Weitere Informationen zu Verbundbenutzern finden Sie unter Verbundbenutzer und Rollen im IAM-Leitfaden.
  - AWS Dienstzugriff — Sie können eine IAM-Rolle in Ihrem Konto verwenden, um einem AWS Dienst Berechtigungen für den Zugriff auf die Ressourcen Ihres Kontos zu erteilen. Sie können beispielsweise eine Rolle erstellen, mit der Amazon Redshift in Ihrem Namen auf einen Amazon S3-Bucket zugreifen und die im Bucket gespeicherten Daten in einen Amazon Redshift-Cluster laden kann. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen für einen AWS Dienst.
  - Auf Amazon EC2 ausgeführte Anwendungen — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer Amazon EC2 EC2-Instance ausgeführt werden und API-Anfragen stellen AWS . Dies ist dem Speichern von

Zugriffsschlüsseln innerhalb der Amazon EC2 EC2-Instance vorzuziehen. Um einer Amazon EC2 EC2-Instance eine AWS Rolle zuzuweisen und sie für alle ihre Anwendungen verfügbar zu machen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht Programmen, die auf der Amazon-EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen zu erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Zugriffskontrolle

Um AWS Cloud Map Ressourcen zu erstellen, zu aktualisieren, zu löschen oder aufzulisten, benötigen Sie Berechtigungen, um die Aktion auszuführen, und Sie benötigen die Erlaubnis, auf die entsprechenden Ressourcen zuzugreifen. Darüber hinaus benötigen Sie gültige Zugriffsschlüssel, um die Aktion programmgesteuert ausführen zu können.

In den folgenden Abschnitten wird beschrieben, wie Sie Berechtigungen für verwalten AWS Cloud Map. Wir empfehlen Ihnen, zunächst die Übersicht zu lesen.

- Verwaltung der Zugriffsberechtigungen für Ihre AWS Cloud Map Ressourcen
- Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für AWS Cloud Map
- AWS Cloud Map Referenz zu API-Berechtigungen

Verwaltung der Zugriffsberechtigungen für Ihre AWS Cloud Map Ressourcen

Jede AWS Ressource gehört einem AWS Konto, und die Berechtigungen zum Erstellen oder Zugreifen auf eine Ressource werden durch Berechtigungsrichtlinien geregelt.

> **Note**
>
> Ein Kontoadministrator (oder Administratorbenutzer) ist ein Benutzer mit Administratorrechten. Weitere Informationen über Administratoren finden Sie unter Bewährte Methoden für IAM im IAM Benutzerhandbuch.

Wenn Sie Berechtigungen erteilen, entscheiden Sie, wer die Berechtigungen erhält, für welche Ressourcen Berechtigungen vergeben werden und welche Aktionen für die Benutzer zulässig sind.

ARNs für Ressourcen AWS Cloud Map

Sie können Berechtigungen auf Ressourcenebene für Namespaces und Services für ausgewählte Operationen gewähren oder verweigern. Weitere Informationen finden Sie unter AWS Cloud Map Referenz zu API-Berechtigungen.

Grundlegendes zum Eigentum an Ressourcen

Ein AWS Konto besitzt die Ressourcen, die in dem Konto erstellt wurden, unabhängig davon, wer die Ressourcen erstellt hat. Insbesondere ist der Ressourcenbesitzer das AWS Konto der Prinzipalentität (d. h. das Root-Benutzerkonto, ein IAM-Benutzer oder eine IAM-Rolle), das die Anfrage zur Ressourcenerstellung authentifiziert.

Die Funktionsweise wird anhand der folgenden Beispiele deutlich:

- Wenn Sie die Root-Benutzerkonto-Anmeldeinformationen Ihres AWS Kontos verwenden, um einen HTTP-Namespace zu erstellen, ist Ihr AWS Konto der Eigentümer der Ressource.
- Wenn Sie in Ihrem AWS Konto einen IAM-Benutzer erstellen und diesem Benutzer Berechtigungen zum Erstellen eines HTTP-Namespace gewähren, kann der Benutzer einen HTTP-Namespace erstellen. Ihr AWS Konto, zu dem der Benutzer gehört, besitzt jedoch die HTTP-Namespace-Ressource.
- Wenn Sie in Ihrem AWS Konto eine IAM-Rolle mit Berechtigungen zum Erstellen eines HTTP-Namespace erstellen, kann jeder, der die Rolle übernehmen kann, einen HTTP-Namespace erstellen. Ihr AWS Konto, zu dem die Rolle gehört, besitzt die HTTP-Namespace-Ressource.

Verwalten des Zugriffs auf Ressourcen

Eine Berechtigungsrichtlinie gibt an, wer Zugriff auf welche Objekte hat. In diesem Abschnitt werden die Optionen zum Erstellen von Berechtigungsrichtlinien für AWS Cloud Map erläutert. Allgemeine Informationen über die Syntax und Beschreibungen von IAM-Richtlinien finden Sie in der IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

Richtlinien, die mit einer IAM-Identität verknüpft sind, werden als identitätsbasierte Richtlinien (IAM-Richtlinien) bezeichnet, und Richtlinien, die mit einer Ressource verknüpft sind, werden als ressourcenbasierte Richtlinien bezeichnet. AWS Cloud Map unterstützt nur identitätsbasierte Richtlinien (IAM-Richtlinien).

Themen

- Identitätsbasierte Richtlinien (IAM-Richtlinien)
- Ressourcenbasierte Richtlinien

Identitätsbasierte Richtlinien (IAM-Richtlinien)

Richtlinien können IAM-Identitäten angefügt werden. Sie können z. B. Folgendes tun:

- Ordnen Sie einem Benutzer oder einer Gruppe in Ihrem Konto eine Berechtigungsrichtlinie zu — Ein Kontoadministrator kann eine einem bestimmten Benutzer zugeordnete Berechtigungsrichtlinie verwenden, um diesem Benutzer Berechtigungen zum Erstellen von AWS Cloud Map Ressourcen zu erteilen.
- Einer Rolle eine Berechtigungsrichtlinie zuordnen (kontoübergreifende Berechtigungen gewähren) — Sie können einem Benutzer, der mit einem anderen AWS Konto erstellt wurde, die Berechtigung zur Ausführung von AWS Cloud Map Aktionen erteilen. Dazu ordnen Sie einer IAM-Rolle eine Berechtigungsrichtlinie zu und erlauben dann dem Benutzer in dem anderen Konto, die Rolle einzunehmen. Im folgenden Beispiel wird erklärt, wie dieser Vorgang für zwei AWS -Konten, Konto A und Konto B, funktioniert:

  1. Der Administrator von Konto A erstellt eine IAM-Rolle und weist ihr eine Berechtigungsrichtlinie zu, die Berechtigungen zum Erstellen von oder für den Zugriff auf Ressourcen erteilt, die Konto A gehören.
  2. Der Administrator von Konto A weist der Rolle eine Vertrauensrichtlinie zu. Die Vertrauensrichtlinie identifiziert Konto B als Prinzipal, der die Rolle einnehmen darf.
  3. Anschließend kann der Administrator von Konto B Berechtigungen für Benutzer oder Gruppen in Konto B zuweisen, sodass diese die Rolle einnehmen können. Auf diese Weise können Benutzer in Konto B Ressourcen in Konto A erstellen bzw. darauf zugreifen.

  Weitere Informationen zur Vorgehensweise beim Delegieren von Berechtigungen an Benutzer aus einem anderen AWS -Konto finden Sie unter Access Management (Zugriffsverwaltung) im IAM-Benutzerhandbuch.

Mit der folgenden Beispielrichtlinie kann ein Benutzer die CreatePublicDnsNamespaceAktion ausführen, um einen öffentlichen DNS-Namespace für ein beliebiges AWS Konto zu erstellen. Die Amazon Route 53-Berechtigungen sind erforderlich, da beim Erstellen eines öffentlichen DNS-Namespace AWS Cloud Map auch eine von Route 53 gehostete Zone erstellt wird:

```
{
```

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

Wenn Sie möchten, dass die Richtlinie stattdessen für private DNS-Namespaces gilt, müssen Sie Berechtigungen zur Verwendung der Aktion erteilen. AWS Cloud Map CreatePrivateDnsNamespace Darüber hinaus erteilen Sie die Erlaubnis, dieselben Route 53-Aktionen wie im vorherigen Beispiel zu verwenden, da eine private, gehostete Route 53-Zone AWS Cloud Map erstellt wird. Sie erteilen außerdem die Erlaubnis, zwei Amazon EC2 EC2-Aktionen zu verwenden, `DescribeVpcs` und`DescribeRegions`:

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

```
        }
    ]
}
```

Weitere Informationen zum Anhängen von Richtlinien an Identitäten für AWS Cloud Map finden Sie unter. Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für AWS Cloud Map Weitere Informationen zu Benutzern, Gruppen, Rollen und Berechtigungen finden Sie im Thema Identitäten (Benutzer, Gruppen und Rollen) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Andere Services, z. B. Amazon S3, unterstützen auch die Zuordnung von Berechtigungsrichtlinien zu Ressourcen. Sie können beispielsweise eine Richtlinie an einen S3-Bucket anhängen, um die Zugriffsberechtigungen für diesen Bucket zu verwalten. AWS Cloud Map unterstützt das Anhängen von Richtlinien an Ressourcen nicht.

Angeben der Richtlinienelemente: Ressourcen, Aktionen, Effekte und Prinzipale

AWS Cloud Map umfasst API-Aktionen (siehe AWS Cloud Map API-Referenz), die Sie für jede AWS Cloud Map Ressource verwenden können (sieheARNs für Ressourcen AWS Cloud Map). Sie können Benutzern oder verbundenen Benutzern Berechtigungen zur Durchführung beliebiger oder aller dieser Aktionen erteilen. Beachten Sie, dass einige API-Aktionen, z. B. das Erstellen eines öffentlichen DNS-Namespace, Berechtigungen zur Durchführung mehrerer Aktionen erfordern.

Grundlegende Richtlinienelemente:

- Ressource – Sie verwenden einen Amazon-Ressourcennamen (ARN), um die Ressource, für welche die Richtlinie gilt, zu identifizieren. Weitere Informationen finden Sie unter ARNs für Ressourcen AWS Cloud Map.
- Aktion – Mit Aktionsschlüsselwörtern geben Sie die Ressourcenaktionen an, die Sie zulassen oder verweigern möchten. Je nach Angabe `Effect` ermöglicht oder verweigert die `servicediscovery:CreateHttpNamespace` Berechtigung einem Benutzer beispielsweise die Möglichkeit, die AWS Cloud Map CreateHttpNamespaceAktion auszuführen.
- Effekt – Dies ist die von Ihnen festgelegte Auswirkung (entweder Zugriffserlaubnis oder Zugriffsverweigerung), wenn ein Benutzer versucht, die jeweilige Aktion für die angegebene Ressource durchzuführen. Wenn Sie den Zugriff auf eine Aktion nicht ausdrücklich gestatten, wird er implizit verweigert. Sie können den Zugriff auf eine Ressource auch explizit verweigern. So können Sie sicherstellen, dass Benutzer nicht darauf zugreifen können, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird.

- Prinzipal – In identitätsbasierten Richtlinien (IAM-Richtlinien) ist der Benutzer, dem die Richtlinie zugewiesen ist, automatisch der Prinzipal. In ressourcenbasierten Richtlinien müssen Sie den Benutzer, das Konto, den Service oder die sonstige Entität angeben, die die Berechtigungen erhalten soll (gilt nur für ressourcenbasierte Richtlinien). AWS Cloud Map bietet keine Unterstützung für ressourcenbasierte Richtlinien.

Weitere Informationen über die Syntax und Beschreibungen von IAM-Richtlinien finden Sie in der IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

Eine Liste der AWS Cloud Map API-Aktionen und der Ressourcen, für die sie gelten, finden Sie unterAWS Cloud Map Referenz zu API-Berechtigungen.

Bedingungen in einer IAM-Richtlinie angeben

Beim Erteilen von Berechtigungen können Sie mithilfe der IAM-Richtliniensyntax die Bedingungen angeben, unter denen die Richtlinie wirksam werden soll. Beispielsweise kann festgelegt werden, dass eine Richtlinie erst ab einem bestimmten Datum angewendet wird, oder dass eine Richtlinie nur für einen bestimmten Namespace gilt.

Um Bedingungen auszudrücken, verwenden Sie vordefinierte Bedingungsschlüssel. AWS Cloud Map definiert seinen eigenen Satz von Bedingungsschlüsseln und unterstützt auch die Verwendung einiger globaler Bedingungsschlüssel. Weitere Informationen finden Sie unter den folgenden Themen:

- Hinweise zu AWS Cloud Map Bedingungsschlüsseln finden Sie unterAWS Cloud Map Referenz zu API-Berechtigungen.
- Informationen zu AWS globalen Bedingungsschlüsseln finden Sie unter AWS Global Condition Context Keys im IAM-Benutzerhandbuch.
- Informationen zur Angabe von Bedingungen in einer Richtliniensprache finden Sie unter IAM JSON Policy Elements: Condition im IAM-Benutzerhandbuch.

Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für AWS Cloud Map

Dieses Thema enthält Beispiele für identitätsbasierte Richtlinien, die zeigen, wie ein Kontoadministrator Berechtigungsrichtlinien an IAM-Identitäten (Benutzer, Gruppen und Rollen) anhängen und so Berechtigungen zur Ausführung von Aktionen an Ressourcen gewähren kann. AWS Cloud Map

> **Important**
>
> Wir empfehlen Ihnen, zunächst die einführenden Themen zu lesen, in denen die grundlegenden Konzepte und Optionen für die Verwaltung des Zugriffs auf Ihre Ressourcen erläutert werden. AWS Cloud Map Weitere Informationen finden Sie unter Verwaltung der Zugriffsberechtigungen für Ihre AWS Cloud Map Ressourcen.

Das folgende Beispiel zeigt eine Berechtigungsrichtlinie, die einem Benutzer die Berechtigung zum Registrieren, Abmelden und Registrieren von Service-Instances gewährt. Der Abschnitt `Sid` (die Anweisungs-ID) ist optional:

```
{
   "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

Eine Liste der Aktionen mit den anzugebenden ARNs, mit denen ihnen Berechtigungen erteilt oder entzogen werden, finden Sie unter AWS Cloud Map Referenz zu API-Berechtigungen.

Erforderliche Berechtigungen für die Verwendung der AWS Cloud Map -Konsole

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

```
{
   "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.

Zum Erstellen eines AWS Cloud Map Dienstes sind Berechtigungen erforderlich

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.

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

AWS 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 für alle AWS Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete 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: AWSCloudMapDiscoverInstanceAccess

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

Die Berechtigungen für diese Richtlinie finden Sie AWSCloudMapDiscoverInstanceAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSCloudMapReadOnlyAccess

Sie können `AWSCloudMapReadOnlyAccess` an Ihre IAM-Entitäten anhängen. Gewährt nur Lesezugriff auf alle AWS Cloud Map Aktionen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie AWSCloudMapReadOnlyAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSCloudMapRegisterInstanceAccess

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 in der Referenz zu verwalteten Richtlinien. AWSCloudMapRegisterInstanceAccessAWS

AWS verwaltete Richtlinie: AWSCloudMapFullAccess

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

Die Berechtigungen für diese Richtlinie finden Sie AWSCloudMapFullAccessin der Referenz zu AWS verwalteten Richtlinien.

AWS Cloud Map Aktualisierungen AWS verwalteter 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 AWS Cloud Map Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite 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

Beispiele für vom Kunden verwaltete Richtlinien für AWS Cloud Map Aktionen

Sie können Ihre eigenen benutzerdefinierten IAM-Richtlinien erstellen, um Berechtigungen für AWS Cloud Map Aktionen zu gewähren. Die benutzerdefinierten Richtlinien können Sie dann den IAM-Benutzern oder -Gruppen zuweisen, welche die angegebenen Berechtigungen benötigen. Diese Richtlinien funktionieren, wenn Sie die AWS Cloud Map API, die AWS SDKs oder die AWS CLI verwenden. Die folgenden Beispiele zeigen Berechtigungen für einige häufige Anwendungsfälle. Informationen zur Richtlinie, die einem Benutzer vollen Zugriff gewährt AWS Cloud Map, finden Sie unterErforderliche Berechtigungen für die Verwendung der AWS Cloud Map -Konsole.

Beispiele

- Beispiel 1: Lesezugriff auf alle AWS Cloud Map Ressourcen zulassen
- Beispiel 2: Erstellen aller Arten von Namespaces erlauben

Beispiel 1: Lesezugriff auf alle AWS Cloud Map Ressourcen zulassen

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

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

Beispiel 2: Erstellen aller Arten von Namespaces erlauben

Die folgende Berechtigungsrichtlinie erlaubt Benutzern, alle Arten von Namespaces zu erstellen:

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

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen IAM-Benutzer im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

AWS Cloud Map Referenz zu API-Berechtigungen

Wenn Sie eine Berechtigungsrichtlinie einrichten Zugriffskontrolle und schreiben, die Sie an eine IAM-Identität anhängen können (identitätsbasierte Richtlinien), können Sie die folgende Liste als Referenz verwenden. Die Listen enthalten 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`

DeregisterInstance

Erforderliche Berechtigungen (API-Aktion):

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

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`

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`
- `route53:ChangeResourceRecordSets`
- `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`
- `route53:ChangeResourceRecordSets`

AWS Cloud Map Referenz zu den Bedingungsschlüsseln

AWS Cloud Map definiert die folgenden Bedingungsschlüssel, die im `Condition` Element einer IAM-Richtlinie für bestimmte AWS Cloud Map Aktionen verwendet werden können. Diese Schlüssel können Sie verwenden, um die Bedingungen zu verfeinern, unter denen die Richtlinienanweisung angewendet wird. Einzelheiten dazu, welche AWS Cloud Map Aktionen diese Bedingungsschlüssel akzeptieren, finden Sie unter Aktionen definiert von AWS Cloud Map. Weitere Informationen zu Bedingungsschlüsseln im Allgemeinen finden Sie unterBedingungen in einer IAM-Richtlinie angeben.

`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

Überprüfung der Einhaltung von Vorschriften für AWS Cloud Map

Die Sicherheit und Einhaltung von AWS Cloud Map werden von externen Prüfern im Rahmen mehrerer AWS Compliance-Programme bewertet, darunter Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), ISO und FIPS.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter AWS Services im Umfang nach Compliance-Programmen. Allgemeine Informationen finden Sie unter AWS -Compliance-Programme.

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

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

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von Umgebungen beschrieben, auf denen auf Sicherheit und Compliance ausgerichtete Basisumgebungen eingerichtet werden. AWS
- Whitepaper „Architecting for HIPAA Security and Compliance“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen erstellen AWS können.
- AWS Ressourcen zur Einhaltung von Vorschriften — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- AWS Config— Mit diesem AWS Service wird bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus und hilft Ihnen AWS , die Einhaltung der Sicherheitsstandards und bewährten Verfahren der Sicherheitsbranche zu überprüfen.

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, mit denen der Zustand von Ressourcen in bestimmten Regionen überprüft wird, z. B. Amazon EC2 EC2-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 AWS PrivateLink damit eine private Verbindung zwischen Ihrer VPC und AWS Cloud Map herstellen. 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. Die AWS SDKs AWS CLI und stellen dem Service-Endpunkt verschiedene Host-Präfixe voran, wenn Sie jede API-Operation 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. z. B. `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 Fehler 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

Protokollieren von AWS Cloud Map API-Aufrufen 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.
- Ob die Anfrage im Namen eines IAM Identity Center-Benutzers gestellt wurde.

- 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 einen Pfad mit einer oder mehreren Regionen erstellen, indem Sie den verwenden. AWS CLI Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten in Ihrem Konto AWS-Regionen erfassen. Wenn du einen Trail mit nur einer Region erstellst, kannst du dir nur die Ereignisse ansehen, die in den Trails protokolliert wurden. AWS-Region Weitere Informationen zu Trails finden Sie unter Einen Trail für Sie erstellen AWS-Konto und Einen Trail für eine Organisation erstellen 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 der 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 angeben würden, wenn Sie erweiterte Event-Selektoren mithilfe der AWS CLI APIs oder konfigurieren würden. CloudTrail In der CloudTrail Spalte „Protokollierte Daten-APIs“ werden die API-Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten-APIs, bei denen angemeldet wurde CloudTrail
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances • DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • DiscoverInstancesRevision

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

Das folgende Beispiel zeigt, wie erweiterte Event-Selektoren 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	Fügen Sie SD hinzu ResourceTag
Löschen Sie ein oder mehrere Tags.	UntagResource	untag-resource	SD entfernen ResourceTag

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Listet Tags für eine Ressource auf	ListTagsForResource	list-tags-for-resource	Holen Sie sich SD ResourceTag

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-SD HttpNamespace
Erstellen eines privaten Namespace auf DNS-Basis	CreatePrivateDnsNamespace	create-private-dns-namespace	Neu-SD PrivateDnsNamespace

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Erstellen eines öffentlichen Namespace auf DNS-Basis	CreatePublicDnsNamespace	create-public-dns-namespace	Neu-SD PublicDns Namespace
Einen Service erstellen	CreateService	create-service	New-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 für den Betrieb 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 mit dem anzuzeigen 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. DiscoverInstancesAPI-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 Wiederholungen im Referenzhandbuch für AWS SDKs und Tools.

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
Tutorials hinzugefügt	Zwei Tutorials mit häufigen Anwendungsfällen 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	`AWSCloudMapDiscoverInstanceAccess` `AWSCloudMapRegisterInstanceAccess` , und `AWSCloudMapReadOnlyAccess` die Richtlinien wurden aktualisiert.	20. September 2023
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-Support	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 DiscoverInstances API-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 der APIs und hinzugefügt. AWS CLI	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.