Guía para desarrolladores

AWS Cloud Map

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Cloud Map: Guía para desarrolladores

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es AWS Cloud Map? ... 1
Componentes de AWS Cloud Map ... 1
Accediendo AWS Cloud Map ... 2
AWS Identity and Access Management ... 4
AWS Cloud Map Precios ... 4
AWS Cloud Map y conformidad con AWS la nube ... 5
Introducción ... 6
Configuración ... 6
Inscríbase en AWS ... 7
Acceda a la API, AWS CLIAWS Tools for Windows PowerShell, o al AWS SDKs ... 9
Configure el AWS Command Line Interface o AWS Tools for Windows PowerShell ... 10
Descarga un AWS SDK ... 11
Úselo AWS Cloud Map con consultas de DNS y llamadas a la API ... 11
Requisitos previos ... 11
Paso 1: Crea un espacio de nombres ... 12
Paso 2: Crear los servicios ... 13
Paso 3: Crear las instancias de servicio ... 14
Paso 4: Descubra las instancias de servicio ... 14
Paso 5: Eliminar ... 16
Utilice la detección de AWS Cloud Map servicios con consultas de DNS y llamadas a la API mediante AWS CLI ... 17
... 17
Requisitos previos ... 17
Crea un AWS Cloud Map espacio de nombres ... 17
Crea los servicios AWS Cloud Map ... 18
Registre las instancias AWS Cloud Map de servicio ... 20
Descubra las instancias AWS Cloud Map de servicio ... 21
Limpie los recursos ... 23
AWS Cloud Map Úselo con atributos personalizados ... 24
Requisitos previos ... 24
Paso 1: Crea un espacio de nombres ... 24
Paso 2: Crear una tabla de DynamoDB ... 25
Paso 3: Crear el servicio de datos ... 25
Paso 4: Crear un rol de ejecución ... 26

Paso 5: Crear la función Lambda para escribir datos ... 26
Paso 6: Crear el servicio de aplicaciones ... 28
Paso 7: Crear la función Lambda para leer los datos ... 29
Paso 8: Crear una instancia de servicio ... 30
Paso 9: Crear y ejecutar aplicaciones cliente ... 30
Paso 10: Limpiar ... 33
Utilice la detección de AWS Cloud Map servicios con atributos personalizados mediante el AWS CLI ... 34
... 34
Requisitos previos ... 34
Cree un AWS Cloud Map espacio de nombres ... 34
Creación de una tabla de DynamoDB ... 35
Cree un servicio AWS Cloud Map de datos y registre la tabla de DynamoDB ... 36
Crear un rol de IAM para las funciones de Lambda ... 36
Cree la función Lambda para escribir datos ... 38
Cree un servicio de AWS Cloud Map aplicaciones y registre la función de escritura Lambda ... 40
Cree la función Lambda para leer datos ... 41
Registrar la función de lectura Lambda como una instancia de servicio ... 43
Cree y ejecute aplicaciones cliente ... 43
Eliminar recursos ... 46
Espacios de nombres ... 48
Creación de un espacio de nombres ... 48
Opciones de descubrimiento de instancias ... 49
Procedimiento ... 53
Pasos a seguir a continuación ... 56
Listar espacios de nombres ... 57
Eliminación de un espacio de nombres ... 59
Servicios ... 61
Configuración de la comprobación de estado ... 62
Controles de estado de Route 53 ... 62
Comprobaciones de estado personalizadas ... 63
Configuración del DNS ... 64
Política de direccionamiento ... 64
Tipo de registro ... 65
Crear un servicio ... 67

Pasos a seguir a continuación ... 72
Actualización de un servicio ... 73
Listar servicios en un espacio de nombres ... 75
Eliminación de un servicio ... 77
Instancias de servicio ... 79
Registrar una instancia de servicio ... 79
Listado de instancias de servicio ... 85
Actualización de una instancia de servicio ... 87
Actualización de los atributos personalizados de una instancia de servicio ... 87
Anular el registro de una instancia de servicio ... 88
Seguridad ... 90
Identity and Access Management ... 90
Público ... 91
Autenticación con identidades ... 92
Administración de acceso mediante políticas ... 95
¿Cómo AWS Cloud Map funciona con IAM ... 98
Ejemplos de políticas basadas en identidades ... 106
AWS políticas gestionadas ... 113
AWS Cloud Map Referencia de permisos de API ... 115
Solución de problemas ... 119
Validación de la conformidad ... 121
Resiliencia ... 122
Seguridad de infraestructuras ... 122
AWS PrivateLink ... 123
Monitorización ... 126
Registra las llamadas a la AWS Cloud Map API mediante AWS CloudTrail ... 126
Eventos de datos ... 128
Eventos de administración ... 129
Ejemplos de evento ... 130
Etiquetado de recursos ... 134
Cómo se etiquetan los recursos ... 134
Restricciones ... 135
Actualizar las etiquetas de los recursos AWS Cloud Map ... 136
Service Quotas ... 138
Administrar sus cuotas de servicio ... 139
Gestione la limitación de las DiscoverInstances solicitudes de API ... 140

Cómo se aplica la limitación 141
Ajuste de las cuotas de limitación de las API 142
Historial de documentos 143
..... cxlvi

¿Qué es AWS Cloud Map?

AWS Cloud Map es una solución totalmente gestionada que puede utilizar para asignar nombres lógicos a los servicios y recursos de backend de los que dependen sus aplicaciones. También ayuda a sus aplicaciones a descubrir recursos mediante una de las llamadas a la AWS SDKs RESTful API o las consultas de DNS. AWS Cloud Map solo sirve recursos en buen estado, que pueden ser tablas de Amazon DynamoDB (DynamoDB), colas de Amazon Simple Queue Service (Amazon SQS), cualquier servicio de aplicaciones de nivel superior creado con instancias de Amazon Elastic Compute Cloud (Amazon) EC2 o tareas de Amazon Elastic Container Service (Amazon ECS), etc.

Componentes de AWS Cloud Map

Espacio de nombres

Para empezar, primero debe crear un espacio de AWS Cloud Map nombres que funcione como una forma de agrupar los servicios de una aplicación. Un espacio de nombres identifica el nombre que desea usar para ubicar los recursos y también especifica cómo desea ubicar los recursos: mediante llamadas a la AWS Cloud Map DiscoverInstancesAPI, consultas de DNS en una VPC o consultas de DNS públicas. En la mayoría de los casos, un espacio de nombres contiene todos los servicios de una aplicación, por ejemplo, una aplicación de facturación. Para obtener más información, consulte AWS Cloud Map espacios de nombres.

Servicio

Tras crear un espacio de nombres, se crea un AWS Cloud Map servicio para cada tipo de recurso que se desee utilizar para localizar los puntos finales. AWS Cloud Map Por ejemplo, puede crear servicios para servidores web y servidores de bases de datos.

Un servicio es una plantilla que se AWS Cloud Map utiliza cuando la aplicación agrega otro recurso, como otro servidor web. Si, al crear el espacio de nombres, eligió localizar los recursos mediante DNS, un servicio contiene información sobre los tipos de registros que desea utilizar para localizar el servidor web. Un servicio también indica si desea comprobar el estado del recurso y si desea utilizar las comprobaciones de estado de Amazon Route 53 o un comprobador de estado de terceros. Para obtener más información, consulte AWS Cloud Map servicios.

Instancia de servicio

Cuando la aplicación agrega un recurso, puede llamar a la acción de la AWS Cloud Map RegisterInstanceAPI en el código, lo que crea una instancia de AWS Cloud Map servicio en

un servicio. La instancia de servicio contiene información sobre cómo la aplicación puede localizar el recurso, ya sea mediante el DNS o mediante la acción de la AWS Cloud Map DiscoverInstancesAPI.

Cuando la aplicación necesita conectarse a un recurso, llama DiscoverInstanceso utiliza consultas de DNS públicas o privadas especificando el espacio de nombres y el servicio asociados al recurso. AWS Cloud Map devuelve información sobre cómo localizar uno o más recursos. Si especificó la comprobación de estado al crear el servicio, solo AWS Cloud Map devuelve las instancias en buen estado. Para obtener más información, consulte AWS Cloud Map instancias de servicio.

Accediendo AWS Cloud Map

Puede acceder de AWS Cloud Map las siguientes maneras:

- AWS Management Console— Los procedimientos de esta guía explican cómo utilizarlos AWS Management Console para realizar tareas.
- AWS SDKs— Si utilizas un lenguaje de programación que AWS proporciona un SDK para, puedes usar un SDK para acceder AWS Cloud Map. SDKs simplifique la autenticación, integre fácilmente su entorno de desarrollo y proporcione acceso a AWS Cloud Map los comandos. Para obtener más información, consulte Herramientas para Amazon Web Services.
- AWS Command Line Interface— Para obtener más información, consulte Cómo empezar a usarlo AWS CLI en la Guía del AWS Command Line Interface usuario.
- AWS Tools for Windows PowerShell— Para obtener más información, consulte Primeros pasos con el AWS Tools for Windows PowerShell en la Guía del Herramientas de AWS para PowerShell usuario.
- AWS Cloud Map API: si utilizas un lenguaje de programación para el que no hay un SDK disponible, consulta la referencia de la AWS Cloud Map API para obtener información sobre las acciones de la API y sobre cómo realizar solicitudes a la API.

> **Note**
>
> IPv6 Client Support: a partir del 22 de junio de 2023, en todas las regiones nuevas, todos los comandos que se envíen AWS Cloud Map desde `IPv6` los clientes se enrutarán a un nuevo punto final de doble pila (). `servicediscovery.<region>.api.aws` AWS Cloud Map `IPv6`-solo se puede acceder a las redes tanto para los terminales antiguos

(**`servicediscovery.<region>.amazonaws.com`**) como para los de doble pila en las siguientes regiones, que se publicaron antes del 22 de junio de 2023:

- EE. UU. Este (Ohio) us-east-2
- EE. UU. Este (Norte de Virginia) us-east-1
- EE. UU. Oeste (Norte de California) us-west-1
- EE. UU. Oeste (Oregón) us-west-2
- África (Ciudad del Cabo) (af-south-1)
- Asia-Pacífico (Hong Kong) ap-east-1
- Asia-Pacífico (Hyderabad): ap-south-2
- Asia-Pacífico (Yakarta) (ap-southeast-3)
- Región Asia Pacífico (Melbourne) (ap-southeast-4)
- Asia-Pacífico (Mumbai) ap-south-1
- Asia-Pacífico (Osaka) ap-northeast-3
- Asia-Pacífico (Seúl) ap-northeast-2
- Asia-Pacífico (Singapur) ap-southeast-1
- Asia-Pacífico (Sídney) ap-southeast-2
- Asia-Pacífico (Tokio) ap-northeast-1
- Canadá (Central) ca-central-1
- UE (Fráncfort) eu-central-1
- UE (Irlanda) eu-west-1
- UE (Londres) eu-west-2
- Europa (Milán) (eu-south-1)
- UE (París) eu-west-3
- Europa (España): eu-south-2
- UE (Estocolmo) eu-north-1
- Europa (Zúrich): eu-central-2
- Medio Oriente (Baréin) (me-south-1)
- Medio Oriente (EAU): me-central-1
- América del Sur (São Paulo) sa-east-1
- AWS GovCloud (EE. UU. Este) — -1 us-gov-east

- AWS GovCloud (EEUU-Oeste) — -1 us-gov-west

AWS Identity and Access Management

AWS Cloud Map se integra con AWS Identity and Access Management (IAM), un servicio que su organización puede utilizar para realizar las siguientes acciones:

- Cree usuarios y grupos en la cuenta de su organización AWS
- Comparta los recursos de su AWS cuenta entre los usuarios de la cuenta de manera eficiente
- Asignar credenciales de seguridad exclusivas a los usuarios
- Controlar de manera detallada el acceso de los usuarios a los servicios y recursos

Por ejemplo, puedes usar IAM with AWS Cloud Map para controlar qué usuarios de tu AWS cuenta pueden crear un nuevo espacio de nombres o registrar instancias.

Para obtener información general sobre IAM, consulte los siguientes recursos:

- Identity and Access Management para AWS Cloud Map
- AWS Identity and Access Management
- Guía del usuario de IAM

AWS Cloud Map Precios

AWS Cloud Map los precios se basan en los recursos que se registran en el registro de servicios y en las llamadas a la API que se realizan para descubrirlos. Aquí no AWS Cloud Map hay pagos por adelantado y solo pagas por lo que utilizas.

Si lo desea, puede habilitar la detección basada en DNS para los recursos con direcciones IP. También puede habilitar la comprobación de estado de los recursos mediante las comprobaciones de estado de Amazon Route 53, independientemente de que la detección de las instancias se realice mediante llamadas a la API o consultas de DNS. Incurrirá en gastos adicionales en relación con el uso de las comprobaciones de estado y de DNS de Route 53.

Para más información, consulte Precios de AWS Cloud Map.

AWS Cloud Map y conformidad con AWS la nube

Para obtener información sobre AWS Cloud Map el cumplimiento de diversas normas de cumplimiento de la seguridad y normas de auditoría, consulte las páginas siguientes:

- AWS Conformidad con la nube
- AWS Servicios incluidos en el ámbito de aplicación del programa de conformidad

Empezar con AWS Cloud Map

Las siguientes guías muestran cómo configurar el uso AWS Cloud Map y la realización de tareas comunes mediante espacios de AWS Cloud Map nombres.

Descripción general de la guía	Más información
¿Cómo registrarse AWS y prepararse para usarla AWS Cloud Map	Configurar para usar AWS Cloud Map
Uso de consultas de DNS y llamadas a la API para descubrir los servicios de backend.	Aprenda a utilizar la detección AWS Cloud Map de servicios con consultas de DNS y llamadas a la API
Uso de consultas de DNS y llamadas a la API para descubrir servicios de backend mediante. AWS CLI	Aprenda a utilizar la detección AWS Cloud Map de servicios con consultas de DNS y llamadas a la API mediante AWS CLI
Crear una aplicación de muestra y usar atributos personalizados en el código para descubrir recursos.	Aprenda a utilizar la detección AWS Cloud Map de servicios con atributos personalizados
Crear una aplicación de muestra y usar atributos personalizados en el código para descubrir recursos mediante AWS CLI.	Aprenda a utilizar la detección AWS Cloud Map de servicios con atributos personalizados mediante el AWS CLI

Configurar para usar AWS Cloud Map

La descripción general y los procedimientos de las siguientes secciones están pensados para ayudarle a empezar a usarlo AWS y a prepararlo para empezar a usarlo AWS Cloud Map.

Temas

- Inscríbase en AWS
- Acceda a la API, AWS CLIAWS Tools for Windows PowerShell, o al AWS SDKs
- Configure el AWS Command Line Interface o AWS Tools for Windows PowerShell
- Descarga un AWS SDK

Inscríbase en AWS

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Acceda a la API, AWS CLIAWS Tools for Windows PowerShell, o al AWS SDKs

Para usar la API, la AWS CLI AWS Tools for Windows PowerShell, o la AWS SDKs, debe crear claves de acceso. Estas claves constan de un ID de clave de acceso y una clave de acceso secreta, que se utilizan para firmar mediante programación las solicitudes que realiza a AWS.

Los usuarios necesitan acceso programático si quieren interactuar con personas AWS ajenas a. AWS Management Console La forma de conceder el acceso programático depende del tipo de usuario que acceda. AWS

Para conceder acceso programático a los usuarios, elija una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal (Usuarios administrados en el IAM Identity Center)	Usa credenciales temporales para firmar las solicitudes programáticas dirigidas al AWS CLI AWS SDKs, o. AWS APIs	Siga las instrucciones de la interfaz que desea utilizar: • Para ello AWS CLI, consulte Configuración del AWS CLI uso AWS IAM Identity Center en la Guía del AWS Command Line Interface usuario. • Para AWS SDKs ver las herramientas y AWS APIs, consulte la autenticación del Centro de Identidad de IAM en la Guía de referencia de herramientas AWS SDKs y herramientas.
IAM	Utilice credenciales temporales para firmar las solicitudes programáticas dirigidas al	Siga las instrucciones de Uso de credenciales temporales con AWS recursos de la Guía del usuario de IAM.

¿Qué usuario necesita acceso programático?	Para	Mediante
	AWS CLI AWS SDKs, o. AWS APIs	
IAM	(No recomendado) Utilice credenciales de larga duración para firmar las solicitudes programáticas dirigidas al AWS CLI AWS SDKs, o. AWS APIs	Siga las instrucciones de la interfaz que desea utilizar: • Para ello AWS CLI, consulte Autenticación con credencia les de usuario de IAM en la Guía del AWS Command Line Interface usuario. • Para obtener AWS SDKs información sobre las herramientas, consulte Autenticarse con credencia les de larga duración en la Guía de referencia de herramientas AWS SDKs y herramientas. • Para ello AWS APIs, consulte Administrar las claves de acceso para los usuarios de IAM en la Guía del usuario de IAM.

Configure el AWS Command Line Interface o AWS Tools for Windows PowerShell

El AWS Command Line Interface (AWS CLI) es una herramienta unificada para administrar AWS los servicios. Para obtener información sobre cómo instalar y configurar el AWS CLI, consulte Instalación o actualización a la última versión del AWS CLI en la Guía del AWS Command Line Interface usuario.

Si tiene experiencia con Windows PowerShell, es posible que prefiera usarlo AWS Tools for Windows PowerShell. Para obtener más información, consulte Configuración de AWS Tools for Windows PowerShell en la Guía del usuario de Herramientas de AWS para PowerShell .

Descarga un AWS SDK

Si utilizas un lenguaje de programación que AWS proporciona un SDK para, te recomendamos que utilices un SDK en lugar de la AWS Cloud Map API. El uso de un SDK tiene varias ventajas. SDKs simplifica la autenticación, se integra fácilmente en el entorno de desarrollo y proporciona acceso a AWS Cloud Map los comandos. Para obtener más información, consulte Herramientas para Amazon Web Services.

Aprenda a utilizar la detección AWS Cloud Map de servicios con consultas de DNS y llamadas a la API

El siguiente tutorial simula una arquitectura de microservicios con dos servicios de backend. El primer servicio se podrá detectar mediante una consulta de DNS. El segundo servicio solo se podrá detectar mediante la AWS Cloud Map API.

> **Note**
>
> Los detalles del recurso, como los nombres de dominio y las direcciones IP, son únicamente para fines de simulación. No se pueden resolver a través de Internet.

Para obtener una end-to-end AWS CLI versión de este tutorial, consulteAprenda a utilizar la detección AWS Cloud Map de servicios con consultas de DNS y llamadas a la API mediante AWS CLI.

Requisitos previos

Se deben cumplir los siguientes requisitos previos para completar el tutorial correctamente.

- Antes de comenzar, complete los pasos de Configurar para usar AWS Cloud Map.
- Si aún no lo ha instalado AWS Command Line Interface, siga los pasos que se indican en Instalar o actualizar la última versión del AWS CLI para instalarlo.

El tutorial requiere un intérprete de comandos o un terminal de línea de comando para ejecutar los comandos. En Linux y macOS, use su administrador de intérprete de comandos y paquetes preferido.

> **Note**
>
> En Windows, algunos comandos de la CLI de Bash que se utilizan habitualmente con Lambda (por ejemplo, `zip`) no son compatibles con los terminales integrados del sistema operativo. Para obtener una versión de Ubuntu y Bash integrada con Windows, instale el subsistema de Windows para Linux.

- El tutorial requiere un entorno local con el comando de utilidad de búsqueda de `dig` DNS.

Paso 1: Crear un AWS Cloud Map espacio de nombres

En este paso, crearás un espacio de nombres público AWS Cloud Map . AWS Cloud Map crea una zona alojada de Route 53 en su nombre con el mismo nombre. Esto le permite descubrir las instancias de servicio creadas en este espacio de nombres mediante registros DNS públicos o mediante llamadas a la AWS Cloud Map API.

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en. https://console.aws.amazon.com/cloudmap/
2. Elija Crear espacio de nombres.
3. Para el nombre del espacio de nombres, especifique. `cloudmap-tutorial.com`

   > **Note**
   >
   > Si vas a usarlo en producción, asegúrate de especificar el nombre de un dominio del que seas propietario o al que tengas acceso. Sin embargo, para los fines de este tutorial, no es necesario que se esté utilizando un dominio real.

4. (Opcional) En la descripción del espacio de nombres, especifique una descripción del uso que desee darle al espacio de nombres.
5. Para la detección de instancias, selecciona las llamadas a la API y las consultas de DNS públicas.
6. Deje el resto de los valores predeterminados y elija Crear espacio de nombres.

Paso 2: Crea los servicios AWS Cloud Map

En este paso, se crean dos servicios. El primer servicio se podrá detectar mediante llamadas a DNS y API públicas. El segundo servicio se podrá detectar únicamente mediante llamadas a la API.

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/.
2. En el panel de navegación izquierdo, selecciona Espacios de nombres para ver una lista de los espacios de nombres que has creado.
3. En la lista de espacios de nombres, seleccione el espacio de nombres y elija Ver detalles. **`cloudmap-tutorial.com`**
4. En la sección Servicios, elija Crear servicio y haga lo siguiente para crear el primer servicio.
   a. En Nombre del servicio, escriba `public-service`. El nombre del servicio se aplicará a los registros DNS que AWS Cloud Map cree. El formato que se utiliza es*`<service-name>`*`.`*`<namespace-name>`*.
   b. Para la configuración de detección de servicios, seleccione API y DNS.
   c. En la sección de configuración de DNS, en Política de enrutamiento, seleccione Enrutamiento de respuesta de valores múltiples.

      > **Note**
      > La consola lo traducirá a MULTIVALUE después de seleccionarlo. Para obtener más información sobre las opciones de enrutamiento disponibles, consulte Elegir una política de enrutamiento en la Guía para desarrolladores de Route 53.

   d. Deje el resto de los valores predeterminados y elija Crear servicio para volver a la página de detalles del espacio de nombres.
5. En la sección Servicios, selecciona Crear servicio y haz lo siguiente para crear el segundo servicio.
   a. En Nombre del servicio, escriba `backend-service`.
   b. Para la configuración de detección de servicios, seleccione solo API.
   c. Deje el resto de los valores predeterminados y elija Crear servicio.

Paso 3: Registrar las instancias AWS Cloud Map de servicio

En este paso, crearás dos instancias de servicio, una para cada servicio de nuestro espacio de nombres.

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en. https://console.aws.amazon.com/cloudmap/
2. En la lista de espacios de nombres, seleccione el espacio de nombres que creó en el paso 1 y elija Ver detalles.
3. En la página de detalles del espacio de nombres, en la lista de servicios, seleccione el **`public-service`** servicio y elija Ver detalles.
4. En la sección Instancias de servicio, elija Registrar instancia de servicio y haga lo siguiente para crear la primera instancia de servicio.
   a. En ID de instancia de servicio, especifique`first`.
   b. Para IPv4 la dirección, especifique`192.168.2.1`.
   c. Deje el resto de los valores predeterminados y elija Registrar instancia de servicio.
5. Con la ruta de navegación situada en la parte superior de la página, selecciona cloudmap-tutorial.com para volver a la página de detalles del espacio de nombres.
6. En la página de detalles del espacio de nombres, en la lista de servicios, selecciona el servicio de backend y selecciona Ver detalles.
7. En la sección Instancias de servicio, selecciona Registrar instancia de servicio y haz lo siguiente para crear la segunda instancia de servicio.
   a. En el ID de instancia de servicio, especifique si `second` desea indicar que se trata de la segunda instancia de servicio.
   b. En Tipo de instancia, seleccione Información de identificación de otro recurso.
   c. En el caso de los atributos personalizados, añada un par clave-valor con `service-name` como clave y `backend` como valor.
   d. Elija Register service instance (Registrar instancia de servicio).

Paso 4: Descubra las instancias de servicio AWS Cloud Map

Ahora que se han creado el AWS Cloud Map espacio de nombres, los servicios y las instancias de servicio, puede comprobar que todo funciona detectando las instancias. Usa el `dig` comando

para verificar la configuración del DNS público y la AWS Cloud Map API para verificar el servicio de backend. Para obtener más información sobre el `dig` comando, consulta dig: utilidad de búsqueda de DNS.

1. Inicie sesión en la consola de Route 53 AWS Management Console y ábrala en https://console.aws.amazon.com/route53/.
2. En el panel de navegación izquierdo, elija Hosted zones (Zonas alojadas).
3. Seleccione la zona alojada en cloudmap-tutorial.com. Esto muestra los detalles de la zona alojada en un panel independiente. Tome nota de los servidores de nombres asociados a su zona alojada, ya que los utilizaremos en el siguiente paso.
4. Con el comando dig y uno de los servidores de nombres de Route 53 de la zona alojada, consulte los registros DNS de la instancia de servicio.

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

   El `ANSWER SECTION` resultado debe mostrar la IPv4 dirección que asoció a su `public-service` servicio.

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

5. Con el AWS CLI, consulte los atributos de las segundas instancias de servicio.

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

   El resultado muestra los atributos que asoció al servicio como pares clave-valor.

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

```
    ],
    "InstancesRevision": 71462688285136850
}
```

Paso 5: Limpiar los recursos

Una vez que haya completado el tutorial, puede eliminar los recursos. AWS Cloud Map requiere que los limpie en orden inverso, primero las instancias de servicio, después los servicios y, por último, el espacio de nombres. AWS Cloud Map limpiará los recursos de Route 53 en tu nombre cuando sigas estos pasos.

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/.
2. En la lista de espacios de nombres, seleccione el espacio de **`cloudmap-tutorial.com`** nombres y elija Ver detalles.
3. En la página de detalles del espacio de nombres, en la lista de servicios, seleccione el **`public-service`** servicio y elija Ver detalles.
4. En la sección Instancias de servicio, seleccione la **`first`** instancia y elija Anular registro.
5. Con la ruta de navegación situada en la parte superior de la página, selecciona cloudmap-tutorial.com para volver a la página de detalles del espacio de nombres.
6. En la página de detalles del espacio de nombres, en la lista de servicios, selecciona el servicio de servicio público y selecciona Eliminar.
7. Repita los pasos 3 a 6 para. `backend-service`
8. En el panel de navegación de la izquierda, selecciona Namespaces.
9. Seleccione el espacio de **`cloudmap-tutorial.com`** nombres y elija Eliminar.

> **Note**
>
> Aunque AWS Cloud Map limpia los recursos de Route 53 por ti, puedes ir a la consola de Route 53 para comprobar que se ha eliminado la zona `cloudmap-tutorial.com` alojada.

Aprenda a utilizar la detección AWS Cloud Map de servicios con consultas de DNS y llamadas a la API mediante AWS CLI

En este tutorial se muestra cómo utilizar la detección de AWS Cloud Map servicios mediante la AWS Command Line Interface (CLI). Creará una arquitectura de microservicios con dos servicios de backend: uno que se puede detectar mediante consultas de DNS y otro que se puede detectar únicamente mediante la API. AWS Cloud Map

Para ver un tutorial que incluye los pasos de la AWS Cloud Map consola, consulte. Aprenda a utilizar la detección AWS Cloud Map de servicios con consultas de DNS y llamadas a la API

Requisitos previos

Se deben cumplir los siguientes requisitos previos para completar el tutorial correctamente.

- Antes de comenzar, complete los pasos de Configurar para usar AWS Cloud Map.
- Si aún no lo ha instalado AWS Command Line Interface, siga los pasos que se indican en Instalar o actualizar la última versión del AWS CLI para instalarlo.

  El tutorial requiere un intérprete de comandos o un terminal de línea de comando para ejecutar los comandos. En Linux y macOS, use su administrador de intérprete de comandos y paquetes preferido.

  > **Note**
  >
  > En Windows, algunos comandos de la CLI de Bash que se utilizan habitualmente con Lambda (por ejemplo, `zip`) no son compatibles con los terminales integrados del sistema operativo. Para obtener una versión de Ubuntu y Bash integrada con Windows, instale el subsistema de Windows para Linux.

- El tutorial requiere un entorno local con el comando de utilidad de búsqueda de `dig` DNS.

Crea un AWS Cloud Map espacio de nombres

En primer lugar, crearás un espacio de nombres público AWS Cloud Map . AWS Cloud Map creará una zona alojada en Route 53 con el mismo nombre, lo que permitirá la detección de servicios mediante registros de DNS y llamadas a la API.

1. Cree el espacio de nombres DNS público:

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

   El comando devuelve un identificador de operación que puedes usar para comprobar el estado de la creación del espacio de nombres:

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

2. Compruebe el estado de la operación para confirmar que el espacio de nombres se creó correctamente:

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

3. Una vez que la operación se haya realizado correctamente, obtenga el ID del espacio de nombres:

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

   Este comando devuelve el ID del espacio de nombres, que necesitarás para los pasos siguientes:

```
ns-abcd1234xmplefgh
```

Crea los servicios AWS Cloud Map

Ahora, crea dos servicios dentro de tu espacio de nombres. El primer servicio se podrá detectar mediante llamadas al DNS y a la API, mientras que el segundo solo se podrá detectar mediante llamadas a la API.

1. Cree el primer servicio con la detección de DNS habilitada:

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

El comando devuelve detalles sobre el servicio creado:

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

2. Cree el segundo servicio con una detección exclusiva mediante API:

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

El comando devuelve detalles sobre el servicio creado:

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

Registre las instancias AWS Cloud Map de servicio

A continuación, registre las instancias de servicio para cada uno de sus servicios. Estas instancias representan los recursos reales que se descubrirán.

1. Registre la primera instancia con una IPv4 dirección para la detección de DNS:

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

   El comando devuelve un identificador de operación:

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

2. Comprueba el estado de la operación para confirmar que la instancia se registró correctamente:

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

3. Registra la segunda instancia con atributos personalizados para la detección de la API:

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

El comando devuelve un identificador de operación:

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

4. Comprueba el estado de la operación para confirmar que la instancia se registró correctamente:

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

Descubra las instancias AWS Cloud Map de servicio

Ahora que ha creado y registrado sus instancias de servicio, puede comprobar que todo funciona detectándolas mediante consultas de DNS y la AWS Cloud Map API.

1. Primero, obtén el ID de la zona hospedada de Route 53:

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

Esto devuelve el ID de la zona alojada:

```
/hostedzone/Z1234ABCDXMPLEFGH
```

2. Obtenga los servidores de nombres de su zona alojada:

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

```
    --output text
```

Esto devuelve uno de los servidores de nombres:

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

3. Usa el `dig` comando para consultar los registros DNS de tu servicio público:

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

El resultado debe mostrar la IPv4 dirección que asociaste a tu servicio:

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

4. Usa el AWS CLI para descubrir la instancia del servicio de backend:

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

El resultado muestra los atributos que ha asociado al servicio:

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

Limpie los recursos

Una vez que hayas completado el tutorial, limpia los recursos para evitar incurrir en cargos. AWS Cloud Map requiere que los limpies en orden inverso: primero las instancias de servicio, después los servicios y, por último, el espacio de nombres.

1. Anule el registro de la primera instancia de servicio:

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

2. Anule el registro de la segunda instancia de servicio:

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

3. Elimine el servicio público:

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

4. Elimine el servicio de backend:

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

5. Elimine el espacio de nombres de :

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

6. Compruebe que se elimine la zona alojada de Route 53:

```
aws route53 list-hosted-zones-by-name \
```

```
    --dns-name cloudmap-tutorial.com
```

Aprenda a utilizar la detección AWS Cloud Map de servicios con atributos personalizados

El siguiente tutorial muestra cómo utilizar la detección de AWS Cloud Map servicios con atributos personalizados que se pueden detectar mediante la AWS Cloud Map API. El tutorial explica cómo crear y ejecutar aplicaciones cliente mediante AWS CloudShell. Las aplicaciones utilizan dos funciones de Lambda para escribir datos en una tabla de DynamoDB y, a continuación, leerlos de la tabla. Las funciones de Lambda y la tabla de DynamoDB se registran como instancias de servicio. AWS Cloud Map El código de las aplicaciones cliente y las funciones Lambda utiliza atributos AWS Cloud Map personalizados para descubrir los recursos necesarios para realizar el trabajo.

Para obtener una versión AWS CLI basada de este tutorial, consulteAprenda a utilizar la detección AWS Cloud Map de servicios con atributos personalizados mediante el AWS CLI.

> **Important**
>
> Creará AWS recursos durante el taller, lo que tendrá un coste en su AWS cuenta. Se recomienda limpiar los recursos tan pronto como termine el taller para minimizar el costo.

Requisitos previos

Antes de comenzar, complete los pasos de Configurar para usar AWS Cloud Map.

Paso 1: Crea un AWS Cloud Map espacio de nombres

En este paso, crearás un AWS Cloud Map espacio de nombres. Un espacio de nombres es una construcción que se utiliza para agrupar los servicios de una aplicación. Al crear el espacio de nombres, se especifica cómo se podrán detectar los recursos. Los recursos creados en el espacio de nombres creado en este paso se podrán detectar mediante llamadas a la API mediante atributos personalizados. AWS Cloud Map

1. Inicie sesión en AWS Management Console y abra la consola en AWS Cloud Map . https://console.aws.amazon.com/cloudmap/
2. Elija Crear espacio de nombres.

3. Para el nombre del espacio de nombres, especifique. `cloudmap-tutorial`
4. (Opcional) En la descripción del espacio de nombres, especifique una descripción del uso que va a dar al espacio de nombres.
5. Para la detección de instancias, selecciona Llamadas a la API.
6. Deje el resto de los valores predeterminados y elija Crear espacio de nombres.

Paso 2: Crear una tabla de DynamoDB

En este paso, creará una tabla de DynamoDB. La tabla se utiliza para almacenar y recuperar datos para la aplicación de ejemplo que se va a crear en los pasos siguientes.

Para obtener información sobre cómo crear un DynamoDB, consulte el paso 1: Crear una tabla en DynamoDB en la Guía para desarrolladores de DynamoDB y utilice la siguiente tabla para determinar qué opciones especificar.

Opción	Valor
Nombre de la tabla	mapa de nubes
Clave de partición	id

Mantenga los valores predeterminados para el resto de la configuración y cree la tabla.

Paso 3: Crear un servicio de AWS Cloud Map datos y registrar la tabla de DynamoDB como instancia

En este paso, se crea un AWS Cloud Map servicio y, a continuación, se registra la tabla de DynamoDB creada en el último paso como instancia de servicio.

1. Abra la consola en AWS Cloud Map https://console.aws.amazon.com/cloudmap/
2. En la lista de espacios de nombres, seleccione el espacio de **`cloudmap-tutorial`** nombres y elija Ver detalles.
3. En la sección Servicios, selecciona Crear servicio y haz lo siguiente.
   a. En Nombre del servicio, escriba `data-service`.
   b. Deje el resto de los valores predeterminados y elija Crear servicio.

4. En la sección Servicios, selecciona el `data-service` servicio y elige Ver detalles.
5. En la sección Instancias de servicio, selecciona Registrar instancia de servicio.
6. En la página Registrar una instancia de servicio, haga lo siguiente.
   a. En Tipo de instancia, seleccione Información de identificación para otro recurso.
   b. En ID de instancia de servicio, especifique`data-instance`.
   c. En la sección Atributos personalizados, especifique el siguiente par clave-valor: clave =`tablename`, valor =. `cloudmap`

Paso 4: Crear un AWS Lambda rol de ejecución

En este paso, se crea un rol de IAM que utilizará la AWS Lambda función del paso siguiente. Puede asignar un nombre a la función de IAM `cloudmap-tutorial-role` y omitir el límite de los permisos, ya que la función solo se utiliza en este tutorial y, posteriormente, puede eliminarla.

Para crear el rol de servicio para Lambda (consola de IAM)

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en. https://console.aws.amazon.com/iam/
2. En el panel de navegación de la consola de IAM, seleccione Roles y, a continuación, elija Crear rol.
3. En Tipo de entidad de confianza, elija Servicio de AWS.
4. Para Servicio o caso de uso, elija Lambda y, a continuación, elija el caso de uso de Lambda.
5. Elija Siguiente.
6. Busque y seleccione la casilla situada junto a la **`PowerUserAccess`** política y, a continuación, seleccione Siguiente.
7. Elija Siguiente.
8. En Nombre del rol, especifique`cloudmap-tutorial-role`.
9. Revise el rol y, a continuación, elija Crear rol.

Paso 5: Crear la función Lambda para escribir datos

En este paso, creará una función Lambda creada desde cero que escriba datos en la tabla de DynamoDB mediante la AWS Cloud Map API para consultar el servicio que ha creado. AWS Cloud Map

Para obtener información sobre la creación de una función Lambda, consulte Crear una función Lambda con la consola en la Guía para AWS Lambda desarrolladores y utilice la siguiente tabla para determinar qué opciones especificar o elegir.

Opción	Valor
Nombre de la función	función de escritura
Tiempo de ejecución	Python 3.12
Arquitectura	x86_64
Permisos	Usa un rol existente
Rol existente	cloudmap-tutorial-role

Tras crear la función, actualice el código de ejemplo para que refleje el siguiente código de Python y, a continuación, implemente la función. Tenga en cuenta que está especificando el atributo `datatable` personalizado que ha asociado a la instancia de AWS Cloud Map servicio que ha creado para la tabla de DynamoDB. La función genera una clave que es un número aleatorio entre 1 y 100 y la asocia a un valor que se pasa a la función cuando se llama a ella.

```
import json
import boto3
import random

def lambda_handler(event, context):

    serviceclient = boto3.client('servicediscovery')

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

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

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)
```

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

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

Tras implementar la función, para evitar errores de tiempo de espera, actualice el tiempo de espera de la función a 5 segundos. Para obtener más información, consulte Configurar el tiempo de espera de una función Lambda en la AWS Lambda Guía para desarrolladores.

Paso 6: Crear un servicio de AWS Cloud Map aplicaciones y registrar la función de escritura de Lambda como una instancia

En este paso, se crea un AWS Cloud Map servicio y, a continuación, se registra la función de escritura de Lambda como instancia de servicio.

1. Abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación de la izquierda, selecciona Namespaces.
3. En la lista de espacios de nombres, seleccione el espacio de **`cloudmap-tutorial`** nombres y elija Ver detalles.
4. En la sección Servicios, selecciona Crear servicio y haz lo siguiente.
   a. En Nombre del servicio, escriba `app-service`.
   b. Deje el resto de los valores predeterminados y elija Crear servicio.
5. En la sección Servicios, selecciona el `app-service` servicio y elige Ver detalles.
6. En la sección Instancias de servicio, selecciona Registrar instancia de servicio.
7. En la página Registrar una instancia de servicio, haga lo siguiente.
   a. En Tipo de instancia, seleccione Información de identificación para otro recurso.
   b. En ID de instancia de servicio, especifique`write-instance`.
   c. En la sección Atributos personalizados, especifique los siguientes pares clave-valor.
      - clave =**`action`**, valor = `write`
      - clave =`functionname`, valor = `writefunction`

Paso 7: Crear la función Lambda para leer los datos

En este paso, creará una función Lambda creada desde cero que escriba datos en la tabla de DynamoDB que ha creado.

Para obtener información sobre la creación de una función Lambda, consulte Crear una función Lambda con la consola en la Guía para AWS Lambda desarrolladores y utilice la siguiente tabla para determinar qué opciones especificar o elegir.

Opción	Valor
Nombre de la función	función de lectura
Tiempo de ejecución	Python 3.12
Arquitectura	x86_64
Permisos	Usa un rol existente
Rol existente	cloudmap-tutorial-role

Tras crear la función, actualice el código de ejemplo para que refleje el siguiente código de Python y, a continuación, implemente la función. La función escanea la tabla y devuelve todos los elementos.

```
import json
import boto3

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

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

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

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

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

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

Después de implementar la función, para evitar errores de tiempo de espera, actualice el tiempo de espera de la función a 5 segundos. Para obtener más información, consulte Configurar el tiempo de espera de una función Lambda en la AWS Lambda Guía para desarrolladores.

Paso 8: Registrar la función de lectura Lambda como una AWS Cloud Map instancia de servicio

En este paso, registrará la función de lectura Lambda como una instancia de servicio en el `app-service` servicio que creó anteriormente.

1. Abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación de la izquierda, selecciona Namespaces.
3. En la lista de espacios de nombres, seleccione el espacio de **`cloudmap-tutorial`** nombres y elija Ver detalles.
4. En la sección Servicios, selecciona el **`app-service`** servicio y elige Ver detalles.
5. En la sección Instancias de servicio, selecciona Registrar instancia de servicio.
6. En la página Registrar una instancia de servicio, haga lo siguiente.
   a. En Tipo de instancia, seleccione Información de identificación para otro recurso.
   b. En ID de instancia de servicio, especifique`read-instance`.
   c. En la sección Atributos personalizados, especifique los siguientes pares clave-valor.
      - clave =**`action`**, valor = `read`
      - clave =`functionname`, valor = `readfunction`

Paso 9: Crear y ejecutar clientes de lectura y escritura en AWS CloudShell

Puede crear y ejecutar aplicaciones cliente AWS CloudShell que utilicen código para descubrir los servicios que ha configurado AWS Cloud Map y realizar llamadas a estos servicios.

1. Abra la AWS CloudShell consola en https://console.aws.amazon.com/cloudshell/
2. Utilice el siguiente comando para crear un archivo llamado`writefunction.py`.

```
vim writeclient.py
```

3. En el `writeclient.py` archivo, entre en el modo de inserción pulsando el `i` botón. A continuación, copia y pega el siguiente código. Este código descubre la función Lambda para escribir datos buscando el atributo personalizado `name=writeservice` en el `app-service` servicio. Se devuelve el nombre de la función Lambda responsable de escribir los datos en la tabla de DynamoDB. A continuación, se invoca la función Lambda y se pasa una carga útil de muestra que se escribe en la tabla como un valor.

```
import boto3

serviceclient = boto3.client('servicediscovery')

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

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

lambdaclient = boto3.client('lambda')

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

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

4. Pulse la tecla escape `:wq`, escriba y pulse la tecla enter para guardar el archivo y salir.
5. Usa el siguiente comando para ejecutar el código de Python.

```
python3 writeclient.py
```

El resultado debe ser una `200` respuesta similar a la siguiente.

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

6. Para comprobar que la escritura se realizó correctamente en el paso anterior, cree un cliente de lectura.

   a. Use el siguiente comando para crear un archivo llamado`readfunction.py`.

   ```
   vim readclient.py
   ```

   b. En el `readclient.py` archivo, pulse el `i` botón para entrar en el modo de inserción. A continuación, copia y pega el siguiente código. Este código escanea la tabla y devolverá el valor que escribiste en la tabla en el paso anterior.

   ```
   import boto3

   serviceclient = boto3.client('servicediscovery')

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

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

   lambdaclient = boto3.client('lambda')

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

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

   c. Pulse la tecla escape`:wq`, escriba y pulse la tecla enter para guardar el archivo y salir.

   d. Usa el siguiente comando para ejecutar el código de Python.

   ```
   python3 readclient.py
   ```

   El resultado debe tener un aspecto similar al siguiente, con una lista del valor escrito en la tabla mediante la ejecución `writefunction.py` y la clave aleatoria generada en la función de escritura de Lambda.

   ```
     b'{"statusCode": 200, "body": "{\\"Items\\": [{\\"id\\": \\"45\
   \", \\"todo\\": \\"This is a test data\\"}], \\"Count\\": 1, \
   \"ScannedCount\\": 1, \\"ResponseMetadata\\": {\\"RequestId\\": \
   \"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"HTTPStatusCode\
   \": 200, \\"HTTPHeaders\\": {\\"server\\": \\"Server\\", \\"date\\": \\"Thu, 25
   ```

```
 Jul 2024 20:43:33 GMT\\", \\"content-type\\": \\"application/x-amz-json-1.0\
\", \\"content-length\\": \\"91\\", \\"connection\\": \\"keep-alive\\", \\"x-
amzn-requestid\\": \\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\",
 \\"x-amz-crc32\\": \\"1163081893\\"}, \\"RetryAttempts\\": 0}}"}'
```

Paso 10: Limpiar los recursos

Una vez que haya completado el tutorial, elimine los recursos para evitar incurrir en cargos adicionales. AWS Cloud Map requiere que los limpie en orden inverso, primero las instancias de servicio, después los servicios y, por último, el espacio de nombres. En los siguientes pasos, se explica cómo limpiar los AWS Cloud Map recursos utilizados en el tutorial.

Para eliminar los AWS Cloud Map recursos

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/.
2. En la lista de espacios de nombres, seleccione el espacio de **`cloudmap-tutorial`** nombres y elija Ver detalles.
3. En la página de detalles del espacio de nombres, en la lista de servicios, seleccione el **`data-service`** servicio y elija Ver detalles.
4. En la sección Instancias de servicio, seleccione la **`data-instance`** instancia y elija Anular registro.
5. Con la ruta de navegación situada en la parte superior de la página, selecciona cloudmap-tutorial.com para volver a la página de detalles del espacio de nombres.
6. En la página de detalles del espacio de nombres, en la lista de servicios, selecciona el servicio de datos y selecciona Eliminar.
7. Repita los pasos 3 a 6 para el `app-service` servicio y las `write-instance` instancias de servicio. `read-instance`
8. En el panel de navegación de la izquierda, selecciona Namespaces.
9. Seleccione el espacio de **`cloudmap-tutorial`** nombres y elija Eliminar.

En la siguiente tabla se enumeran los procedimientos que puede seguir para eliminar los demás recursos utilizados en el tutorial.

Recurso	Pasos	
Tabla de DynamoDB	Paso 6: (opcional) Elimine la tabla de DynamoDB para limpiar los recursos de la Guía para desarrolladores de Amazon DynamoDB	
Funciones Lambda y función de ejecución de IAM asociada	Limpie en la guía para desarrolladores AWS Lambda	

Aprenda a utilizar la detección AWS Cloud Map de servicios con atributos personalizados mediante el AWS CLI

En este tutorial se muestra cómo utilizar la detección AWS Cloud Map de servicios con atributos personalizados. Creará una aplicación de microservicios que se utilizará AWS Cloud Map para descubrir recursos de forma dinámica mediante atributos personalizados. La aplicación consta de dos funciones de Lambda que escriben y leen datos en una tabla de DynamoDB, con todos los recursos registrados. AWS Cloud Map

Para obtener una AWS Management Console versión del tutorial, consulte. Aprenda a utilizar la detección AWS Cloud Map de servicios con atributos personalizados

Requisitos previos

Antes de comenzar este tutorial, complete los pasos que se indican enConfigurar para usar AWS Cloud Map.

Cree un AWS Cloud Map espacio de nombres

Un espacio de nombres es una construcción que se utiliza para agrupar los servicios de una aplicación. En este paso, crearás un espacio de nombres que permita detectar los recursos mediante llamadas a la API. AWS Cloud Map

1. Ejecuta el siguiente comando para crear un espacio de nombres HTTP:

```
aws servicediscovery create-http-namespace \
```

```
  --name cloudmap-tutorial \
  --creator-request-id cloudmap-tutorial-request
```

El comando devuelve un identificador de operación. Puede comprobar el estado de la operación con el siguiente comando:

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

2. Una vez creado el espacio de nombres, puedes recuperar su ID para usarlo en los siguientes comandos:

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

3. Guarde el ID del espacio de nombres en una variable para su uso posterior:

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

Creación de una tabla de DynamoDB

A continuación, cree una tabla de DynamoDB que almacene los datos de la aplicación:

1. Ejecute el siguiente comando para crear la tabla:

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

2. Espere a que la tabla se active antes de continuar:

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

Este comando espera hasta que la tabla esté completamente creada y lista para usarse.

Cree un servicio AWS Cloud Map de datos y registre la tabla de DynamoDB

Ahora, cree un servicio en su espacio de nombres para representar los recursos de almacenamiento de datos:

1. Ejecute el siguiente comando para crear un AWS Cloud Map servicio para los recursos de almacenamiento de datos:

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

2. Obtenga el ID de servicio del servicio de datos:

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

3. Registre la tabla de DynamoDB como una instancia de servicio con un atributo personalizado que especifique el nombre de la tabla:

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

   El atributo personalizado `tablename=cloudmap` permite a otros servicios descubrir el nombre de la tabla de DynamoDB de forma dinámica.

Crear un rol de IAM para las funciones de Lambda

Cree una función de IAM que las funciones de Lambda utilizarán para acceder a los AWS recursos:

1. Cree el documento de política de confianza para la función de IAM:

```
cat > lambda-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

2. Ejecute el siguiente comando para crear el rol de IAM mediante la política de confianza:

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

3. Cree un archivo para una política de IAM personalizada con permisos de privilegios mínimos:

```
cat > cloudmap-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "servicediscovery:DiscoverInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
```

```
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/cloudmap"
    }
  ]
}
EOF
```

4. Cree y adjunte la política a la función de IAM:

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

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

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

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

Cree la función Lambda para escribir datos

Para crear una función Lambda que escriba datos en la tabla de DynamoDB, siga estos pasos:

1. Cree el archivo Python para la función de escritura:

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

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

        response = serviceclient.discover_instances(
```

```
            NamespaceName='cloudmap-tutorial',
            ServiceName='data-service')

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

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

        dynamodbclient = boto3.resource('dynamodb')

        table = dynamodbclient.Table(tablename)

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

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

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

Esta función se utiliza AWS Cloud Map para descubrir el nombre de la tabla de DynamoDB a partir del atributo personalizado y, a continuación, escribe los datos en la tabla.

2. Package e implemente la función Lambda:

```
zip writefunction.zip writefunction.py

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

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

3. Actualice el tiempo de espera de la función para evitar errores de tiempo de espera:

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

Cree un servicio de AWS Cloud Map aplicaciones y registre la función de escritura Lambda

Para crear otro servicio en el espacio de nombres que represente las funciones de la aplicación, siga estos pasos:

1. Cree un servicio para las funciones de la aplicación:

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

2. Obtenga el ID de servicio del servicio de la aplicación:

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

3. Registre la función de escritura de Lambda como una instancia de servicio con atributos personalizados:

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

Los atributos `action=write` personalizados `functionname=writefunction` permiten a los clientes descubrir esta función en función de su propósito.

Cree la función Lambda para leer datos

Para crear una función Lambda que lea datos de la tabla de DynamoDB, siga estos pasos:

1. Cree el archivo Python para la función de lectura:

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

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

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

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

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

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

        table = dynamodbclient.Table(tablename)

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

        # For production, you would implement pagination like this:
        # items = []
        # while 'LastEvaluatedKey' in response:
        #     items.extend(response['Items'])
        #     response = table.scan(
        #         Select='ALL_ATTRIBUTES',
        #         ExclusiveStartKey=response['LastEvaluatedKey']
        #     )
        # items.extend(response['Items'])

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

Esta función también se utiliza AWS Cloud Map para descubrir el nombre de la tabla de DynamoDB y, a continuación, lee los datos de la tabla. Incluye comentarios sobre la gestión de errores y la paginación.

2. Package e implemente la función Lambda:

```
zip readfunction.zip readfunction.py

aws lambda create-function \
  --function-name readfunction \
  --runtime python3.12 \
```

```
  --role $ROLE_ARN \
  --handler readfunction.lambda_handler \
  --zip-file fileb://readfunction.zip \
  --architectures x86_64
```

3. Actualice el tiempo de espera de la función:

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

Registrar la función de lectura Lambda como una instancia de servicio

Para registrar la función de lectura Lambda como otra instancia de servicio en el servicio de aplicaciones, siga este paso:

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

Los atributos `action=read` personalizados `functionname=readfunction` permiten a los clientes descubrir esta función en función de su propósito.

Cree y ejecute aplicaciones cliente

Para crear una aplicación cliente de Python que utilice AWS Cloud Map para descubrir e invocar la función de escritura, siga estos pasos:

1. Cree un archivo Python para la aplicación cliente de escritura:

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

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

    print("Discovering write function...")
    response = serviceclient.discover_instances(
```

```
        NamespaceName='cloudmap-tutorial',
        ServiceName='app-service',
        QueryParameters={ 'action': 'write' }
    )

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

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

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

    lambdaclient = boto3.client('lambda')

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

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

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

Este cliente usa la `QueryParameters` opción para buscar instancias de servicio con el `action=write` atributo.

2. Cree un archivo Python para la aplicación cliente de lectura:

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

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

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

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

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

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

    lambdaclient = boto3.client('lambda')

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

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

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

3. Ejecute el cliente de escritura para añadir datos a la tabla de DynamoDB:

```
python3 writeclient.py
```

El resultado debería mostrar una respuesta correcta con el código de estado HTTP 200.

4. Ejecute el cliente de lectura para recuperar los datos de la tabla de DynamoDB:

```
python3 readclient.py
```

El resultado debe mostrar los datos que se escribieron en la tabla, incluidos el identificador generado aleatoriamente y el valor «Se trata de un dato de prueba».

Eliminar recursos

Cuando termines con el tutorial, limpia los recursos para evitar incurrir en cargos adicionales.

1. En primer lugar, ejecute el siguiente comando para anular el registro de las instancias de servicio:

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

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

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

2. Ejecute el siguiente comando para eliminar los servicios:

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

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

3. Ejecute el siguiente comando para eliminar el espacio de nombres:

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

4. Ejecute el siguiente comando para eliminar las funciones de Lambda:

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

5. Ejecute el siguiente comando para eliminar la función y la política de IAM:

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

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

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

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

6. Ejecute el siguiente comando para eliminar la tabla de DynamoDB:

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

7. Ejecute el siguiente comando para limpiar los archivos temporales:

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

AWS Cloud Map espacios de nombres

Un espacio de nombres es una entidad lógica AWS Cloud Map que se utiliza para agrupar los servicios de una aplicación con un nombre y un nivel de visibilidad comunes. Al crear un espacio de nombres, se especifica lo siguiente:

- Un nombre que desee que utilice la aplicación para detectar instancias.
- El método mediante el cual se AWS Cloud Map pueden descubrir las instancias de servicio en las que se registra. Puede decidir si sus recursos deben descubrirse públicamente a través de Internet, de forma privada en una nube privada virtual (VPC) específica o solo mediante llamadas a la API.

Los siguientes son conceptos generales sobre los espacios de nombres.

- Los espacios de nombres son específicos del lugar en el Región de AWS que se crearon. Para usarlos AWS Cloud Map en varias regiones, tendrás que crear espacios de nombres en cada región.
- Si crea un espacio de nombres para permitir, por ejemplo, la detección mediante consultas de DNS en una VPC, crea AWS Cloud Map automáticamente una zona alojada de Route 53 privada. Esta zona alojada se puede asociar a varias. VPCs Para obtener más información, consulte Associate VPCWith HostedZone in the Amazon Route 53 API Reference.

Temas

- Crear un espacio de AWS Cloud Map nombres para agrupar los servicios de aplicaciones
- Listar espacios de AWS Cloud Map nombres
- Eliminar un AWS Cloud Map espacio de nombres

Crear un espacio de AWS Cloud Map nombres para agrupar los servicios de aplicaciones

Puede crear un espacio de nombres para agrupar los servicios de su aplicación con un nombre descriptivo que permita descubrir los recursos de la aplicación mediante llamadas a la API o consultas de DNS.

Opciones de descubrimiento de instancias

En la siguiente tabla, se resumen las diferentes opciones de detección de instancias AWS Cloud Map y el tipo de espacio de nombres correspondiente que puede crear, en función de los servicios y la configuración de la aplicación.

Tipo de espacio de nombres	Método de descubrimiento de instancias	Funcionamiento	Información adicional
HTTP	Llamadas a la API	Los recursos de tu aplicación pueden descubrir otros recursos llamando únicamente a la `DiscoverInstances` API.	• DiscoverInstances • CreateHttpNamespace
DNS privado	Llamadas a la API y consultas de DNS en una VPC	Los recursos de su aplicación pueden detectar otros recursos llamando a la `DiscoverInstances` API y consultando los servidores de nombres de la zona alojada privada de Route 53 que se crea automáticamente. AWS Cloud Map La zona alojada creada por AWS Cloud Map tiene el mismo nombre que el espacio de nombres y contiene registros	• DiscoverInstances • CreatePrivateDnsNamespace

Tipo de espacio de nombres	Método de descubrimiento de instancias	Funcionamiento	Información adicional
		DNS con nombres en ese formato. *service-name namespace-name* . **Note** El solucionador de Route 53 resuelve las consultas de DNS que se originan en la VPC utilizando los registros de la zona alojada privada. Si la zona alojada privada no incluye ningún registro que coincida con el nombre de dominio en una consulta de DNS, Route 53 responde a la consulta con `NXDOMAIN`	

Tipo de espacio de nombres	Método de descubrimiento de instancias	Funcionamiento	Información adicional
		(dominio no existente).	

Tipo de espacio de nombres	Método de descubrimiento de instancias	Funcionamiento	Información adicional
DNS público	Llamadas a la API y consultas públicas de DNS	Los recursos de su aplicación pueden descubrir otros recursos llamando a la `DiscoverInstances` API y consultando los servidores de nombres de la zona alojada pública de Route 53 que AWS Cloud Map se crea automáticamente. La zona alojada pública tiene el mismo nombre que el espacio de nombres y contiene registros DNS con nombres en ese formato. *`service-name namespace-name`* . **Note** En este caso, el nombre del espacio de nombres debe ser un nombre de dominio	• DiscoverInstances • CreatePublicDnsNamespace

Tipo de espacio de nombres	Método de descubrimiento de instancias	Funcionamiento	Información adicional
		que hayas registrado.	

Procedimiento

Puedes seguir estos pasos para crear un espacio de nombres con AWS CLI AWS Management Console, o el SDK para Python.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en. https://console.aws.amazon.com/cloudmap/
2. Elija Create namespace (Crear espacio de nombres).
3. En el campo Nombre del espacio de nombres, introduce un nombre que se utilizará para detectar instancias.

   > **Note**
   >
   > - Los espacios de nombres configurados para consultas de DNS públicas deben terminar en un dominio de nivel superior. Por ejemplo, `.com`.
   > - Puede especificar un nombre de dominio internacionalizado (IDN) si convierte primero el nombre a Punycode. Para obtener información sobre convertidores online, busque en Internet “convertidor de punycode”.
   >
   > También puede convertir un nombre de dominio internacionalizado a Punycode al crear espacios de nombres mediante programación. Por ejemplo, si utiliza Java, puede convertir un valor Unicode a Punycode mediante el método `toASCII` de la biblioteca de IDN de java.net.

4. (Opcional) En la descripción del espacio de nombres, introduzca la información sobre el espacio de nombres que estará visible en la página de espacios de nombres y en la sección Información del espacio de nombres. Puede utilizar esta información para identificar fácilmente un espacio de nombres.

5. Para la detección de instancias, puedes elegir entre llamadas a API, llamadas a API y consultas de DNS VPCs, o llamadas a API y consultas de DNS públicas para crear un espacio de nombres HTTP, DNS privado o DNS público, respectivamente. Para obtener más información, consulte Opciones de descubrimiento de instancias.

   En función de lo que selecciones, sigue estos pasos.

   - Si eliges las llamadas a la API y las consultas de DNS en VPCs, para la VPC, elige una nube privada virtual (VPC) a la que quieras asociar el espacio de nombres.
   - Si eliges llamadas a la API y consultas de DNS en VPCs o llamadas a la API y consultas de DNS públicas, en el caso del TTL, especifica un valor numérico en segundos. El valor del tiempo de vida (TTL) determina durante cuánto tiempo los solucionadores de DNS almacenan en caché la información del registro DNS de inicio de autoridad (SOA) de la zona hospedada de Route 53 creada con su espacio de nombres. Para obtener más información sobre TTL, consulte TTL (segundos) en la Guía para desarrolladores de Amazon Route 53.
6. (Opcional) En Etiquetas, elija Añadir etiquetas y, a continuación, especifique una clave y un valor para etiquetar el espacio de nombres. Puede especificar una o varias etiquetas para agregarlas a su espacio de nombres. Las etiquetas te permiten categorizar tus AWS recursos para que puedas administrarlos más fácilmente. Para obtener más información, consulte Etiquetar sus recursos AWS Cloud Map.
7. Elija Create namespace (Crear espacio de nombres). Puede ver el estado de la operación utilizando ListOperations. Para obtener más información, consulte ListOperationsla referencia de la AWS Cloud Map API

AWS CLI

- Crea un espacio de nombres con el comando correspondiente al tipo de descubrimiento de instancias que prefieras (reemplaza *red* los valores por los tuyos).
  - Cree un espacio de nombres de HTTP usando `create-http-namespace`. Las instancias de servicio registradas con un espacio de nombres de HTTP pueden detectarse mediante una solicitud `DiscoverInstances` pero no con DNS.

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

- Cree un espacio de nombre privado basado en DNS, que podrá verse solo dentro de una Amazon VPC especificada. Puede descubrir las instancias que se registraron con un espacio de nombres de DNS privado mediante una solicitud `DiscoverInstances` o mediante DNS.

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

- Cree un espacio de nombres público basado en DNS que se pueda ver en Internet con `create-public-dns-namespace`. Puede descubrir las instancias que se registraron con un espacio de nombres DNS público mediante una solicitud `DiscoverInstances` o mediante DNS.

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

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Crea un espacio de nombres con el comando correspondiente al tipo de descubrimiento de instancias que prefieras (reemplaza *red* los valores por los tuyos):

   - Cree un espacio de nombres de HTTP usando `create_http_namespace()`. Las instancias de servicio registradas con un espacio de nombres de HTTP pueden detectarse usando `discover_instances()` pero no con DNS.

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

   - Cree un espacio de nombre privado basado en DNS, que podrá verse solo dentro de una Amazon VPC especificada. Puede descubrir las instancias que se registraron con un

espacio de nombres DNS privado mediante una solicitud `discover_instances()` o mediante DNS.

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

- Cree un espacio de nombres público basado en DNS que se pueda ver en Internet con `create_public_dns_namespace()`. Puede descubrir las instancias que se registraron con un espacio de nombres DNS público mediante una solicitud `discover_instances()` o mediante DNS.

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

- Salida de respuesta de ejemplo

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

Pasos a seguir a continuación

Tras crear un espacio de nombres, puede crear servicios en el espacio de nombres para agrupar los recursos de la aplicación que, de forma colectiva, sirvan para un propósito concreto en la aplicación. Un servicio actúa como plantilla para registrar los recursos de la aplicación como instancias. Para obtener más información sobre la creación AWS Cloud Map de servicios, consulte Creación de un AWS Cloud Map servicio para un componente de la aplicación.

Listar espacios de AWS Cloud Map nombres

Tras crear los espacios de nombres, puedes ver una lista de los espacios de nombres que has creado siguiendo estos pasos.

AWS Management Console

1. Inicia sesión en AWS Management Console y abre la consola en. AWS Cloud Map https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación, selecciona Espacios de nombres para ver una lista de los espacios de nombres. Puede ordenar los espacios de nombres por nombre, descripción, modo de detección de instancias o ID del espacio de nombres. También puedes introducir un nombre o un ID de espacio de nombres en el campo de búsqueda para localizar y ver un espacio de nombres específico.

AWS CLI

- Enumere los espacios de nombres con el comando `list-namespaces`.

```
aws servicediscovery list-namespaces
```

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Enumere los espacios de nombres con `list_namespaces()`.

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

Salida de respuesta de ejemplo

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

Eliminar un AWS Cloud Map espacio de nombres

Cuando termines de usar un espacio de nombres, puedes eliminarlo. Al eliminar un espacio de nombres, ya no podrá utilizarlo para registrar o detectar instancias de servicio.

> **Note**
>
> Al crear un espacio de nombres, si especifica que desea detectar instancias de servicio mediante consultas de DNS públicas o consultas de DNS VPCs, AWS Cloud Map crea una zona alojada pública o privada de Amazon Route 53. Al eliminar el espacio de nombres, AWS Cloud Map elimina la zona alojada correspondiente.

Antes de eliminar un espacio de nombres, debes anular el registro de todas las instancias de servicio y, a continuación, eliminar todos los servicios que se crearon en el espacio de nombres. Para obtener más información, consulte Anular el registro de una instancia de servicio AWS Cloud Map y Eliminar un AWS Cloud Map servicio.

Después de anular el registro de las instancias y eliminar los servicios que se crearon en un espacio de nombres, sigue estos pasos para eliminar el espacio de nombres.

AWS Management Console

1. Inicia sesión en y abre la consola en. AWS Management Console AWS Cloud Map https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).

3. Seleccione el espacio de nombres que desee eliminar y, a continuación, elija Eliminar.
4. Confirma que deseas eliminar el servicio; para ello, vuelve a seleccionar Eliminar.

AWS CLI

- Elimine un espacio de nombres con el `delete-namespace` comando (sustituya el *red* valor por el suyo propio). Si el espacio de nombres aún contiene uno o más servicios, se producirá un error en la solicitud.

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

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Elimine un espacio de nombres por `delete_namespace()` (sustituya el *red* valor por el suyo propio). Si el espacio de nombres aún contiene uno o más servicios, se producirá un error en la solicitud.

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

Salida de respuesta de ejemplo

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

AWS Cloud Map servicios

Un AWS Cloud Map servicio es una plantilla para registrar instancias de servicio que consta del nombre del servicio y la configuración de DNS, si corresponde, del servicio. También puede configurar una comprobación de estado para determinar el estado de las instancias del servicio y filtrar los recursos en mal estado. Un servicio puede representar un componente de la aplicación. Por ejemplo, puede crear un servicio para los recursos que gestionan los pagos en su aplicación y otro para los recursos que gestionan los usuarios.

Un servicio le permite localizar los recursos de una aplicación recuperando uno o más puntos finales que se pueden utilizar para conectarse al recurso. La ubicación de los recursos se realiza mediante consultas de DNS o la acción de la AWS Cloud Map `DiscoverInstances`API, en función de cómo hayas configurado el espacio de nombres. Puedes usar la AWS Cloud Map consola para analizar la detección de instancias a nivel de servicio.

También puedes especificar metadatos personalizados como atributos a nivel de servicio mediante la `UpdateServiceAttributes` API. Puedes configurar los atributos del servicio para evitar la duplicación de los atributos en las instancias y modificarlos sin necesidad de realizar ningún cambio en los atributos de la instancia. La información que puedes especificar como atributos a nivel de servicio incluye, entre otros, lo siguiente:

- Ponderación de los puntos finales a la hora de desplazar el tráfico durante las implementaciones progresivas.
- Preferencias de servicio, como los tiempos de espera de las API y las políticas de reintentos sugeridas.

Para obtener más información, consulta la referencia UpdateServiceAttributesde la AWS Cloud Map API.

En los temas siguientes se describen las configuraciones de DNS y de comprobación de estado de los servicios, e incluyen instrucciones para crear, enumerar, actualizar y eliminar un servicio.

Temas

- AWS Cloud Map configuración de la comprobación del estado del servicio
- AWS Cloud Map configuración de DNS del servicio
- Creación de un AWS Cloud Map servicio para un componente de la aplicación

- Actualización de un AWS Cloud Map servicio
- Listar AWS Cloud Map servicios en un espacio de nombres
- Eliminar un AWS Cloud Map servicio

AWS Cloud Map configuración de la comprobación del estado del servicio

Los controles de estado ayudan a determinar si las instancias de servicio están en buen estado o no. Si no configuras una comprobación de estado durante la creación del servicio, el tráfico se enrutará a las instancias de servicio independientemente del estado de las instancias. Al configurar una comprobación de estado, AWS Cloud Map devuelve los recursos en buen estado de forma predeterminada. Puedes usar el `HealthStatus` parámetro de la `DiscoverInstances` API para filtrar los recursos por estado y obtener una lista de los recursos en mal estado. También puedes usar la `GetInstancesHealthStatus` API para recuperar el estado de una instancia de servicio concreta.

Al crear un AWS Cloud Map servicio, puede configurar una comprobación de estado de Route 53 o una comprobación de estado personalizada de terceros.

Controles de estado de Route 53

Si especificas la configuración de una comprobación de estado de Amazon Route 53, AWS Cloud Map crea una comprobación de estado de Route 53 cada vez que registras una instancia y la borra cuando cancelas el registro de la instancia.

En el caso de los espacios de nombres DNS públicos, AWS Cloud Map asocia la comprobación de estado al registro de Route 53 que se AWS Cloud Map crea al registrar una instancia. Si especificas ambos `A` tipos de `AAAA` registro en la configuración de DNS de un servicio, AWS Cloud Map crea una comprobación de estado que utiliza la IPv4 dirección para comprobar el estado del recurso. Si el punto final especificado por la IPv4 dirección no está en buen estado, Route 53 considera que tanto los `A` registros como están en mal estado. `AAAA` Si especificas un tipo de `CNAME` registro en la configuración de DNS de un servicio, no podrás configurar una comprobación de estado de Route 53.

En el caso de los espacios de nombres para los que se utilizan llamadas a la API para detectar instancias, AWS Cloud Map crea una comprobación de estado de Route 53. Sin embargo, no hay

ningún registro de DNS AWS Cloud Map al que asociar la comprobación de estado. Para determinar si una comprobación de estado está en buen estado, puede configurar la supervisión mediante la consola de Route 53 o Amazon CloudWatch. Para obtener más información acerca de cómo utilizar la consola de Route 53, consulte Recibir notificaciones cuando se produzca un error en una comprobación de estado en la Guía para desarrolladores de Amazon Route 53. Para obtener más información sobre el uso CloudWatch, consulta PutMetricAlarmla referencia de la CloudWatch API de Amazon.

> **Note**
>
> - No puede configurar una comprobación de estado de Amazon Route 53 para un servicio creado en un espacio de nombres DNS privado.
> - Un comprobador de estado de Route 53 incluido en cada comprobación de estado Región de AWS envía una solicitud de comprobación de estado a un punto final cada 30 segundos. De media, su punto de enlace recibirá una solicitud de comprobación de estado alrededor de cada dos segundos. Sin embargo, los comprobadores de estado no se coordinan entre sí. Por lo tanto, a veces puede ver varias solicitudes en un segundo, seguidas de unos segundos sin comprobaciones de estado. Para ver una lista de las regiones de control de estado, consulte Regiones.

Para obtener más información acerca de los cargos por las comprobaciones de estado de Route 53, consulte Precios de Route 53.

Comprobaciones de estado personalizadas

Si configuras AWS Cloud Map usar un control de estado personalizado al registrar una instancia, debes usar un verificador de estado de terceros para evaluar el estado de tus recursos. Las comprobaciones de estado personalizadas son útiles en las circunstancias siguientes:

- No puede utilizar una comprobación de estado de Route 53 porque el recurso no está disponible en Internet. Por ejemplo, suponga que tiene una instancia que se encuentra en una VPC de Amazon. Puede usar una comprobación de estado personalizada para esta instancia. Sin embargo, para que la comprobación de estado funcione, su comprobador de estado también debe estar en la misma VPC que su instancia.
- Desea utilizar un comprobador de estado de terceros, independientemente de donde se encuentren sus recursos.

Cuando utilizas una comprobación de estado personalizada, AWS Cloud Map no comprueba directamente el estado de un recurso determinado. En su lugar, el comprobador de estado externo comprueba el estado del recurso y devuelve el estado de la aplicación. Luego, su solicitud deberá enviar una `UpdateInstanceCustomHealthStatus` solicitud que transmita este estado a. AWS Cloud Map Si el estado inicial transmitido es`UNHEALTHY`, y si no hay otro `UpdateInstanceCustomHealthStatus` en 30 segundos que transmita un estado de`HEALTHY`, se confirma que el recurso está en mal estado. AWS Cloud Map deja de enrutar el tráfico a ese recurso.

AWS Cloud Map configuración de DNS del servicio

Al crear un servicio en un espacio de nombres que admite la detección de instancias mediante consultas de DNS, AWS Cloud Map crea registros DNS de Route 53. Debe especificar una política de enrutamiento y un tipo de registro DNS de Route 53 que se apliquen a todos los registros DNS de Route 53 que AWS Cloud Map cree.

Política de direccionamiento

Una política de enrutamiento determina cómo responde Route 53 a las consultas de DNS que se utilizan para la detección de instancias de servicio. Las políticas de enrutamiento compatibles y la forma en que AWS Cloud Map se relacionan son las siguientes.

Direccionamiento ponderado

Route 53 devuelve el valor aplicable de una instancia de AWS Cloud Map servicio seleccionada al azar de entre las instancias que registró con el mismo AWS Cloud Map servicio. Todos los registros tienen el mismo valor de ponderación, por lo que no se puede dirigir más o menos tráfico a las instancias.

Por ejemplo, suponga que el servicio incluye configuraciones para un registro A y una comprobación de estado, y utiliza el servicio para registrar 10 instancias. Route 53 responde a las consultas de DNS con la dirección IP de una instancia seleccionada de forma aleatoria entre las instancias con estado correcto. Si ninguna de las instancias tiene un estado correcto, Route 53 responde a las consultas de DNS como si todas las instancias tuvieran un estado correcto.

Si no define ninguna comprobación de estado para el servicio, Route 53 entiende que todas las instancias tienen estado correcto y devuelve el valor aplicable de una instancia seleccionada al azar.

Para obtener más información, consulte Enrutamiento ponderado en la Guía para desarrolladores de Amazon Route 53.

Direccionamiento de respuesta con varios valores

Si define una comprobación de estado para el servicio y el resultado de la misma es correcto, Route 53 devuelve el valor aplicable para un máximo de ocho instancias.

Por ejemplo, supongamos que el servicio incluye configuraciones para un registro A y una comprobación de estado. Utilice el servicio para registrar 10 instancias. Route 53 responde a las consultas de DNS con direcciones IP para solo un máximo de ocho instancias en estado correcto. Si el número de instancias con estado correcto es inferior a ocho, Route 53 responde a todas las consultas de DNS con las direcciones IP de todas las instancias con estado correcto.

Si no define ninguna comprobación de estado para el servicio, Route 53 entiende que todas las instancias tiene estado correcto y devuelve los valores de hasta ocho instancias.

Para obtener más información, consulte Enrutamiento de respuesta con varios valores en la Guía para desarrolladores de Amazon Route 53.

Tipo de registro

Un tipo de registro DNS de Route 53 determina el tipo de valor que Route 53 devuelve en respuesta a las consultas de DNS que se utilizan para la detección de instancias de servicio. Los distintos tipos de registros DNS que puede especificar y los valores asociados que devuelve Route 53 en respuesta a las consultas son los siguientes.

A

Si especifica este tipo, Route 53 devuelve la dirección IP del recurso en un IPv4 formato, como 192.0.2.44.

AAAA

Si especifica este tipo, Route 53 devuelve la dirección IP del recurso en IPv6 formato, como 2001:0 db 8:85 a 3:0000:0000:abcd: 0001:2345.

CNAME

Si especifica este tipo, Route 53 devuelve el nombre de dominio del recurso (por ejemplo, www.example.com).

> **Note**
>
> - Para configurar un registro DNS CNAME, debe especificar la política de enrutamiento y enrutamiento ponderado.
> - Al configurar un registro DNS CNAME, no puede configurar una comprobación de estado de Route 53.

SRV

Si especifica este tipo, Route 53 devuelve el valor de un `SRV` registro. Estos son los valores que se utilizan para un registro de SRV:

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

Considere lo siguiente:

- Los valores de `priority` y `weight` están establecidos en 1 y no se pueden cambiar.
- Para`port`, AWS Cloud Map utiliza el valor que especifique para Port (AWS_INSTANCE_PORT) al registrar una instancia.
- El valor de `service-hostname` es una concatenación de los valores siguientes:
  - El valor que especificas para el ID de instancia de servicio (instanceID) al registrar una instancia
  - El nombre del servicio
  - El nombre del espacio de nombres

  Por ejemplo, supongamos que especificas test como ID de instancia cuando registras una instancia. El nombre del servicio es backend y el nombre del espacio de nombres es example.com. AWS Cloud Map asigna el siguiente valor al atributo `service-hostname` del registro SRV:

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

> **Note**
>
> Si especificas valores como una IPv4 dirección, una IPv6 dirección o ambos al registrar una instancia, crea AWS Cloud Map automáticamente registros A y/o AAAA que tienen el mismo nombre que el valor del `service-hostname` registro SRV.

Puede especificar tipos de registros en las siguientes combinaciones:

- A
- AAAA
- A y AAAA
- CNAME
- SRV

Si especifica los tipos de registro A y AAAA, puede especificar una dirección IPv4 IP, una dirección IPv6 IP o ambas al registrar una instancia.

Creación de un AWS Cloud Map servicio para un componente de la aplicación

Tras crear un espacio de nombres, puede crear servicios para representar los distintos componentes de la aplicación que se destinen a fines específicos. Por ejemplo, puede crear un servicio para los recursos de su aplicación que procese los pagos.

> **Note**
>
> No puedes crear varios servicios a los que se pueda acceder mediante consultas de DNS con nombres que solo difieran según las mayúsculas y minúsculas (como EXAMPLE y EXAMPLE). Si lo intenta, estos servicios tendrán el mismo nombre de DNS. Si utiliza un espacio de nombres al que solo se pueda acceder mediante llamadas a la API, puede crear servicios con nombres que solo se diferencien por las mayúsculas y las minúsculas.

Siga estos pasos para crear un servicio con AWS Management Console AWS CLI, y el SDK para Python.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/.
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).

3. En la página Namespaces (Espacios de nombres), elija el espacio de nombres al que desea añadir el servicio.
4. En la *namespace-name* página Namespace:, selecciona Crear servicio.
5. En Nombre del servicio, introduzca un nombre que describa las instancias que registra al utilizar este servicio. El valor se utiliza para detectar instancias de AWS Cloud Map servicio en las llamadas a la API o en las consultas de DNS.

> **Note**
>
> Si desea AWS Cloud Map crear un registro SRV al registrar una instancia y utiliza un sistema que requiere un formato SRV específico (por ejemplo HAProxy), especifique lo siguiente para el nombre del servicio:
>
> - Comience el nombre con un guion bajo (_), por ejemplo, _exampleservice.
> - Termine el nombre con ***._protocol***, por ejemplo. _tcp.
>
> Cuando registras una instancia, AWS Cloud Map crea un registro SRV y asigna un nombre concatenando el nombre del servicio y el nombre del espacio de nombres, por ejemplo:
> _servicioejemplo._tcp.ejemplo.com

6. (Opcional) En Descripción del servicio, ingresa una descripción para el servicio. La descripción que introduzca aquí aparece en la página de servicios y en la página de detalles de cada servicio.
7. Si el espacio de nombres admite consultas de DNS, en la configuración de detección de servicios, puede configurar la capacidad de detección a nivel de servicio. Elija entre permitir tanto las llamadas a la API como las consultas de DNS o solo las llamadas a la API para la detección de instancias en este servicio.

> **Note**
>
> Si eliges las llamadas a la API, no AWS Cloud Map se crearán registros SRV cuando registres una instancia.

Si eliges API y DNS, sigue estos pasos para configurar los registros de DNS. Puedes añadir o eliminar registros DNS.

1. Para la política de enrutamiento, seleccione la política de enrutamiento de Amazon Route 53 para los registros de DNS que se AWS Cloud Map crean al registrar las instancias. Puede seleccionar entre el enrutamiento ponderado y el enrutamiento de respuesta de valores múltiples. Para obtener más información, consulte Política de direccionamiento.

   > **Note**
   >
   > No puede usar la consola AWS Cloud Map para configurar la creación de un registro de alias de Route 53 al registrar una instancia. Si desea AWS Cloud Map crear registros de alias para un balanceador de cargas de Elastic Load Balancing al registrar instancias mediante programación, elija Enrutamiento ponderado para la política de enrutamiento.

2. En Tipo de registro, elija el tipo de registro DNS que determine qué devuelve Route 53 en respuesta a las consultas de DNS. AWS Cloud Map Para obtener más información, consulte Tipo de registro.
3. En el caso del TTL, especifique un valor numérico para definir el valor del tiempo de vida (TTL), en segundos, a nivel de servicio. El valor de TTL determina durante cuánto tiempo los solucionadores de DNS guardan en memoria caché la información para este registro antes de reenviar otra consulta de DNS a Amazon Route 53 para actualizar la configuración.

8. En Configuración de comprobación de estado, en las opciones de comprobación de estado, elija el tipo de comprobación de estado aplicable a las instancias de servicio. Puede elegir no configurar ninguna comprobación de estado o puede elegir entre una comprobación de estado de Route 53 o una comprobación de estado externa para sus instancias. Para obtener más información, consulte AWS Cloud Map configuración de la comprobación del estado del servicio.

   > **Note**
   >
   > Las comprobaciones de estado de Route 53 solo se pueden configurar para los servicios de los espacios de nombres DNS públicos.

Si elige las comprobaciones de estado de Route 53, proporcione la siguiente información.

1. En el campo Umbral de error, indique un número entre 1 y 10 que defina el número de comprobaciones de estado consecutivas de Route 53 que una instancia de servicio debe superar o no superar para que su estado de salud cambie.
2. Para el protocolo Health Check, seleccione el método que utilizará Route 53 para comprobar el estado de las instancias de servicio.
3. Si elige el protocolo de comprobación de estado HTTP o HTTPS, en Ruta de comprobación de estado, indique la ruta que desee que Amazon Route 53 solicite al realizar las comprobaciones de estado. La ruta puede ser cualquier valor, como el archivo `/docs/route53-health-check.html`. Cuando el estado del recurso es correcto, el valor devuelto es un código de estado HTTP de formato 2xx o 3xx. También puede incluir parámetros de cadena de consulta, como `/welcome.html?language=jp&login=y`. La consola de AWS Cloud Map añade automáticamente un carácter de barra inclinada (/).

Para obtener más información sobre los controles de estado de Route 53, consulte Cómo determina Amazon Route 53 si un chequeo de estado está en buen estado en la Guía para desarrolladores de Amazon Route 53.

9. (Opcional) En Etiquetas, elija Agregar etiquetas y, a continuación, especifique una clave y un valor para etiquetar su espacio de nombres. Puede especificar una o varias etiquetas para agregarlas a su espacio de nombres. Las etiquetas te permiten categorizar tus AWS recursos para que puedas administrarlos más fácilmente. Para obtener más información, consulte Etiquetar sus recursos AWS Cloud Map.
10. Elija Crear servicio.

AWS CLI

- Cree un servicio con el `create-service` comando. Sustituya los *red* valores por los suyos.

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

Salida:

```
{
        "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)

Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.

1. Importe `Boto3` y use `servicediscovery` como su servicio.

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

2. Crea un servicio con`create_service()`. Sustituya los *`red`* valores por los suyos propios. Para obtener más información, consulte create_service.

```
response = client.create_service(
    DnsConfig={
        'DnsRecords': [
            {
```

```
                'TTL': 60,
                'Type': 'A',
            },
        ],
        'NamespaceId': 'ns-xxxxxxxxxxx',
        'RoutingPolicy': 'MULTIVALUE',
    },
    Name='service-name',
    NamespaceId='ns-xxxxxxxxxxx',
)
```

Salida de respuesta de ejemplo

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

Pasos a seguir a continuación

Tras crear un servicio, puede registrar los recursos de la aplicación como instancias de servicio que contienen información sobre cómo la aplicación puede localizar el recurso. Para obtener más

información sobre el registro AWS Cloud Map de instancias de servicio, consulteRegistrar un recurso como instancia de servicio AWS Cloud Map.

Después de crear un servicio, también puede especificar metadatos personalizados, como las ponderaciones de los puntos finales, los tiempos de espera de las API y las políticas de reintentos, como atributos del servicio. Para obtener más información consulte ServiceAttributes y `UpdateServiceAttributes` en la Referencia de la API de AWS Cloud Map .

Actualización de un AWS Cloud Map servicio

Según la configuración del servicio, puede actualizar sus etiquetas, el umbral de error de las comprobaciones de estado de Route 53 y el tiempo de vida (TTL) de los solucionadores de DNS. Para actualizar un servicio, lleve a cabo el siguiente procedimiento.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en. https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).
3. En la página Espacios de nombres, elija el espacio de nombres en el que se crea el servicio.
4. En la ***namespace-name*** página Espacio de nombres:, selecciona el servicio que deseas editar y elige Ver detalles.
5. En la ***service-name*** página Servicio:, selecciona Editar.

   > **Note**
   >
   > No puedes usar el flujo de trabajo del botón Editar para editar los valores de los servicios que solo permiten las llamadas a la API, por ejemplo, la detección de instancias. Sin embargo, puedes añadir o eliminar etiquetas en la *service-name* página Servicio:.

6. En la página Editar servicio, en la sección Descripción del servicio, puede actualizar cualquier descripción previamente establecida para el servicio o añadir una descripción nueva. También puedes añadir etiquetas y actualizar el TTL para los solucionadores de DNS.
7. En la configuración de DNS, para TTL, puede especificar un período de tiempo actualizado, en segundos, que determine cuánto tiempo los solucionadores de DNS almacenan en caché

la información de este registro antes de que envíen otra consulta de DNS a Amazon Route 53 para obtener la configuración actualizada.

8. Si ha configurado las comprobaciones de estado de Route 53, en Umbral de error puede especificar un nuevo número entre 1 y 10 que defina la cantidad de comprobaciones de estado de Route 53 consecutivas que una instancia de servicio debe superar o no para que su estado de salud cambie.
9. Elija Actualizar servicio.

AWS CLI

- Actualice un servicio con el `update-service` comando (sustituya el *`red`* valor por el suyo propio).

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

Salida:

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

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Actualiza un servicio por `update_service()` (reemplaza el *`red`* valor por el tuyo).

```
response = client.update_service(
    Id='srv-xxxxxxxxxxx',
    Service={
```

```
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 300,
                    'Type': 'A',
                },
            ],
        },
        'Description': "new description",
    }
)
```

Salida de respuesta de ejemplo

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

Listar AWS Cloud Map servicios en un espacio de nombres

Para ver una lista de los servicios que ha creado en un espacio de nombres, realice el siguiente procedimiento.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en. https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).
3. Elija el espacio de nombres que contiene los servicios que desea enumerar. Puedes ver una lista de todos los servicios en Servicios e introducir el nombre o la ID del servicio en el campo de búsqueda para encontrar un servicio específico.

AWS CLI

- Enumere los servicios con el comando `list-services`. El siguiente comando muestra todos los servicios de un espacio de nombres utilizando el ID del espacio de nombres como filtro. Sustituya el valor *red* con sus valores propios.

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

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Enumere los servicios con `list_services()`.

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

Salida de respuesta de ejemplo

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

Eliminar un AWS Cloud Map servicio

Para poder eliminar un servicio, antes debe anular el registro de todas las instancias del servicio registradas con este. Para obtener más información, consulte Anular el registro de una instancia de servicio AWS Cloud Map.

Tras anular el registro de todas las instancias registradas mediante el servicio, lleve a cabo el siguiente procedimiento para eliminar el servicio.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en. https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).
3. Elija como opción el espacio de nombres que contiene el servicio que desea eliminar.
4. En la *namespace-name* página Espacio de nombres:, elige la opción del servicio que deseas eliminar.
5. Elija Eliminar.
6. Confirme que desea eliminar el servicio.

AWS CLI

- Elimine un servicio con el `delete-service` comando (sustituya el *red* valor por el suyo propio).

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

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.

2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Elimine un servicio por `delete_service()` (sustituya el *`red`* valor por el suyo propio).

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

Salida de respuesta de ejemplo

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

AWS Cloud Map instancias de servicio

Una instancia de servicio contiene información acerca de cómo localizar un recurso, como un servidor web, para una aplicación. Después de registrar las instancias, puede localizarlas mediante consultas de DNS o la acción de la AWS Cloud Map DiscoverInstancesAPI. Los recursos que puedes registrar incluyen, entre otros, los siguientes:

- EC2 Instancias de Amazon
- Tablas de Amazon DynamoDB
- Buckets de Amazon S3
- Colas de Amazon Simple Queue Service (Amazon SQS)
- APIs desplegado sobre Amazon API Gateway

Puede especificar valores de atributos para las instancias de servicios y los clientes pueden usar estos atributos para filtrar los recursos que AWS Cloud Map devuelven. Por ejemplo, una aplicación puede solicitar recursos que estén en una fase de implementación concreta, como BETA o PROD. También puede usar atributos para el control de versiones.

Los siguientes procedimientos describen cómo puede registrar los recursos de su aplicación como instancias de servicio, ver una lista de instancias registradas en un servicio, editar determinados parámetros de la instancia y anular el registro de una instancia.

Temas

- Registrar un recurso como instancia de servicio AWS Cloud Map
- Listar instancias AWS Cloud Map de servicio
- Actualización de una instancia AWS Cloud Map de servicio
- Anular el registro de una instancia de servicio AWS Cloud Map

Registrar un recurso como instancia de servicio AWS Cloud Map

Puede registrar los recursos de su aplicación como instancias en un AWS Cloud Map servicio. Por ejemplo, supongamos que ha creado un servicio `users` para todos los recursos de la aplicación que administran los datos de los usuarios. A continuación, puede registrar una tabla de DynamoDB que se utilice para almacenar los datos de los usuarios como instancia en este servicio.

> **Note**
>
> Las siguientes funciones no están disponibles en la AWS Cloud Map consola:
>
> - Al registrar una instancia de servicio con la consola, no puede crear un registro de alias que redirija el tráfico a un balanceador de carga de Elastic Load Balancing (ELB). Cuando registra una instancia, debe incluir el atributo `AWS_ALIAS_DNS_NAME`. Para obtener más información, consulta RegisterInstance en la AWS Cloud Map Referencia de la API de .
> - Si registra una instancia con un servicio que incluye una comprobación de estado personalizada, no puede especificar el estado inicial para dicha comprobación. De forma predeterminada, el estado inicial de las comprobaciones de estado personalizadas es Healthy (Buen estado). Si desea que el estado inicial sea Unhealthy (Mal estado), registre la instancia mediante programación e incluya el atributo `AWS_INIT_HEALTH_STATUS`. Para obtener más información, consulta RegisterInstance en la AWS Cloud Map Referencia de la API de .

Para registrar una instancia en un servicio, sigue estos pasos.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/.
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).
3. En la página Namespaces (Espacios de nombres), elija el espacio de nombres que contiene el servicio que desea utilizar como plantilla para registrar una instancia de servicio.
4. En la *`namespace-name`* página Namespace:, elige el servicio que quieres usar.
5. En la *`service-name`* página Servicio:, selecciona Registrar instancia de servicio.
6. En la página Registrar una instancia de servicio, elija un tipo de instancia. Según la configuración de descubrimiento de instancias del espacio de nombres, puedes elegir especificar una dirección IP, un ID de EC2 instancia de Amazon u otra información de identificación para un recurso que no tenga una dirección IP.

   > **Note**
   >
   > Puedes elegir la EC2 instancia solo en los espacios de nombres HTTP.

7. En el caso del ID de instancia de servicio, proporciona un identificador asociado a la instancia de servicio.

> **Note**
>
> Si quieres actualizar una instancia existente, proporciona el identificador asociado a la instancia que quieres actualizar. A continuación, sigue los pasos siguientes para actualizar los valores y volver a registrar la instancia.

8. En función del tipo de instancia que haya elegido, lleve a cabo los siguientes pasos.

> **Important**
>
> No puedes usar el `AWS_` prefijo (no distingue entre mayúsculas y minúsculas) en una clave cuando especificas un atributo personalizado.

Tipo de instancia	Pasos	
Dirección IP	a. En Atributos estándar, para la IPv4dirección, proporciona una IPv4 dirección, si la hay, desde la que la aplicació n pueda acceder al recurso asociado a esta instancia de servicio. b. Como IPv6 dirección , proporciona una dirección IPv6 IP, si la hay, desde la que tus aplicaciones puedan acceder al recurso asociado a esta instancia de servicio.	

Tipo de instancia	Pasos	
	c. En el caso de Port, especifique cualquier puerto que la aplicació n deba incluir para acceder al recurso asociado a esta instancia de servicio. El puerto es obligatorio cuando el servicio incluye un registro de SRV o un chequeo de estado de Amazon Route 53. d. (Opcional) En Atributos personalizados, especifiq ue los pares clave-val or que desee asociar al recurso.	
EC2 instancia	a. Por EC2 ejemplo, ID, selecciona el ID de la EC2 instancia de Amazon que quieres registrar como instancia de AWS Cloud Map servicio. b. (Opcional) En Atributos personalizados, especifiq ue los pares clave-val or que desee asociar al recurso.	

Tipo de instancia	Pasos	
Identificando la información de otro recurso	a. En Atributos estándar, si la configuración del servicio incluye un registro DNS CNAME, verás un campo CNAME. En CNAME, especifique el nombre de dominio que desea que Route 53 devuelva en respuesta a las consultas de DNS (por ejemplo,). `example.com` b. En Atributos personali zados, especifica cualquier información de identificación de un recurso que no sea una dirección IP o un ID de EC2 instancia de Amazon como par clave-valor. Por ejemplo, puede registrar una función Lambda especific ando una clave llamada `function` y proporcio nando el nombre de la función Lambda como valor. También puede especificar una clave llamada `name` y proporcionar un nombre que pueda usar para la detección de instancias mediante programación.	

9. Elija Register service instance (Registrar instancia de servicio).

AWS CLI

- Al enviar una solicitud `RegisterInstance`:
  - Para cada registro de DNS que defina en el servicio especificado por `ServiceId`, se crea o actualiza un registro en la zona alojada que esté asociada al espacio de nombres correspondiente.
  - Si el servicio incluye `HealthCheckConfig`, se crea una comprobación de estado en función de los ajustes de la configuración de la comprobación de estado.
  - Todas las comprobaciones de estado están asociadas a cada uno de los registros nuevos o actualizados.

  Registre una instancia de servicio con el `register-instance` comando (sustituya los *red* valores por los suyos propios).

```
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. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Al enviar una solicitud `RegisterInstance`:
   - Para cada registro de DNS que defina en el servicio especificado por `ServiceId`, se crea o actualiza un registro en la zona alojada que esté asociada al espacio de nombres correspondiente.

- Si el servicio incluye `HealthCheckConfig`, se crea una comprobación de estado en función de los ajustes de la configuración de la comprobación de estado.
- Todas las comprobaciones de estado están asociadas a cada uno de los registros nuevos o actualizados.

Registre una instancia de servicio con `register_instance()` (sustituya los *red* valores por los suyos propios).

```
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)
```

Salida de respuesta de ejemplo

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

Listar instancias AWS Cloud Map de servicio

Para ver una lista de las instancias de servicio que ha registrado con un servicio, realice el siguiente procedimiento.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/.
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).

3. Elija el espacio de nombres que contiene el servicio cuyas instancias desea enumerar.
4. Elija el nombre del servicio que ha utilizado para crear las instancias de servicio. Verás una lista de instancias en Instancias de servicio. Puedes introducir el ID de la instancia en el campo de búsqueda para ver una instancia específica.

AWS CLI

- Enumere las instancias de servicio con el `list-instances` comando (sustituya el *`red`* valor por el suyo propio).

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

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Enumere las instancias de servicio por `list_instances()` (sustituya el *`red`* valor por el suyo propio).

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

Salida de respuesta de ejemplo

```
{
    'Instances': [
        {
            'Attributes': {
                'AWS_INSTANCE_IPV4': '172.2.1.3',
                'AWS_INSTANCE_PORT': '808',
```

```
            },
            'Id': 'i-xxxxxxxxxxxxxxxxx',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

Actualización de una instancia AWS Cloud Map de servicio

Puede actualizar instancias de servicio de dos maneras, dependiendo de los valores que desee actualizar:

- Actualizar cualquier valor: si desea actualizar alguno de los valores que especificó para una instancia de servicio al registrarla, incluidos los atributos personalizados, debe volver a registrar la instancia de servicio y volver a especificar todos los valores. Siga los pasos que se indican a continuación Registrar un recurso como instancia de servicio AWS Cloud Map y especifique el ID de instancia de la instancia de servicio existente como ID de instancia de servicio.

  Como alternativa, puedes usar la `RegisterInstance`API. Puedes especificar el ID de la instancia y el servicio existentes mediante los `ServiceId` parámetros `InstanceId` y, además, volver a especificar otros valores.
- Actualizar solo atributos personalizados: si desea actualizar solo los atributos personalizados de una instancia de servicio, no es necesario volver a registrar la instancia. Solo puede actualizar esos valores. Consulte Actualización de los atributos personalizados de una instancia de servicio.

Actualización de los atributos personalizados de una instancia de servicio

Para actualizar solo los atributos personalizados de una instancia de servicio

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en https://console.aws.amazon.com/cloudmap/.
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).
3. En la página Namespaces (Espacios de nombres), elija el espacio de nombres que contiene el servicio que utilizó originalmente para registrar la instancia de servicio.

4. En la *namespace-name* página Namespace:, elija el servicio que utilizó para registrar la instancia de servicio.
5. En la *service-name* página Servicio:, elija el nombre de la instancia de servicio que desee actualizar.
6. En la sección Custom attributes (Atributos personalizados), elija Edit (Editar).
7. En la *instance-name* página Editar instancia de servicio: añada, elimine o actualice los atributos personalizados. Puede actualizar tanto las claves como los valores de los atributos.
8. Elija Update service instance (Modificar instancia de servicio).

Anular el registro de una instancia de servicio AWS Cloud Map

Para poder eliminar un servicio, antes debe anular el registro de todas las instancias del servicio registradas con este.

Para anular el registro de una instancia de servicio, lleve a cabo el siguiente procedimiento.

AWS Management Console

1. Inicie sesión en AWS Management Console y abra la AWS Cloud Map consola en. https://console.aws.amazon.com/cloudmap/
2. En el panel de navegación, seleccione Namespaces (Espacios de nombres).
3. Elija como opción el espacio de nombres que contiene la instancia de servicio cuyo registro desea anular.
4. En la *namespace-name* página Namespace:, selecciona el servicio que utilizaste para registrar la instancia de servicio.
5. En la *service-name* página Servicio:, elige la instancia de servicio que quieres anular del registro.
6. Elija Anular registro.
7. Confirme que desea anular el registro de la instancia de servicio.

AWS CLI

- Anule el registro de una instancia de servicio con el `deregister-instance` comando (sustituya *red* los valores por los suyos). Este comando elimina los registros DNS de

Amazon Route 53 y cualquier comprobación de estado que se haya AWS Cloud Map creado para la instancia especificada.

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

AWS SDK for Python (Boto3)

1. Si aún no tiene `Boto3` instalado, puede encontrar las instrucciones de instalación, configuración y uso `Boto3` aquí.
2. Importe `Boto3` y use `servicediscovery` como su servicio.

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

3. Anule el registro de una instancia de servicio por `deregister-instance()` (sustituya *red* los valores por los suyos propios). Este comando elimina los registros DNS de Amazon Route 53 y cualquier comprobación de estado que se haya AWS Cloud Map creado para la instancia especificada.

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

Salida de respuesta de ejemplo

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

Seguridad en AWS Cloud Map

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores independientes prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información sobre los programas de conformidad aplicables AWS Cloud Map, consulte los AWS servicios incluidos en el ámbito de aplicación por programa de conformidad.
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

La siguiente documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS Cloud Map. En los temas siguientes se muestra cómo configurarlo AWS Cloud Map para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus AWS Cloud Map recursos.

Temas

- Identity and Access Management para AWS Cloud Map
- Validación de conformidad para AWS Cloud Map
- Resiliencia en AWS Cloud Map
- Seguridad de la infraestructura en AWS Cloud Map

Identity and Access Management para AWS Cloud Map

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de

IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. AWS Cloud Map La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración de acceso mediante políticas
- ¿Cómo AWS Cloud Map funciona con IAM
- Ejemplos de políticas basadas en la identidad para AWS Cloud Map
- AWS políticas administradas para AWS Cloud Map
- AWS Cloud Map Referencia de permisos de API
- Solución de problemas AWS Cloud Map de identidad y acceso

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice. AWS Cloud Map

Usuario del servicio: si utiliza el AWS Cloud Map servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más AWS Cloud Map funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en AWS Cloud Map, consulte Solución de problemas AWS Cloud Map de identidad y acceso.

Administrador de servicios: si estás a cargo de AWS Cloud Map los recursos de tu empresa, probablemente tengas acceso total a ellos AWS Cloud Map. Su trabajo consiste en determinar a qué AWS Cloud Map funciones y recursos deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su gestionador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar la IAM AWS Cloud Map, consulte¿Cómo AWS Cloud Map funciona con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a AWS Cloud Map. Para ver ejemplos de

políticas AWS Cloud Map basadas en la identidad que puede utilizar en IAM, consulte. Ejemplos de políticas basadas en la identidad para AWS Cloud Map

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su gestionador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Autenticación multifactor AWS en IAM en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren

que inicie sesión como usuario raíz, consulta Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulta ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales

temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.

- Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en AWS ellas, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre

la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utiliza para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad

principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puedes establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): SCPs son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.

- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

¿Cómo AWS Cloud Map funciona con IAM

Antes de utilizar IAM para gestionar el acceso AWS Cloud Map, infórmese sobre las funciones de IAM disponibles para su uso. AWS Cloud Map

Característica de IAM	AWS Cloud Map soporte
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí

Característica de IAM	AWS Cloud Map soporte
Claves de condición de política (específicas del servicio)	Sí
ACLs	No
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí
Sesiones de acceso directo (FAS)	Sí
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo AWS Cloud Map funcionan otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en la identidad para AWS Cloud Map

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está asociada. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para AWS Cloud Map

Para ver ejemplos de políticas AWS Cloud Map basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Cloud Map

Políticas basadas en recursos dentro de AWS Cloud Map

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Acciones políticas para AWS Cloud Map

Compatibilidad con las acciones de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de AWS Cloud Map acciones, consulta las acciones definidas AWS Cloud Map en la Referencia de autorización del servicio.

Las acciones políticas AWS Cloud Map utilizan el siguiente prefijo antes de la acción:

```
servicediscovery
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de políticas AWS Cloud Map basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Cloud Map

Recursos de políticas para AWS Cloud Map

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de AWS Cloud Map recursos y sus tipos ARNs, consulte los recursos definidos AWS Cloud Map en la Referencia de autorización de servicios. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS Cloud Map.

Para ver ejemplos de políticas AWS Cloud Map basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Cloud Map

Claves de condición de la política para AWS Cloud Map

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de claves de AWS Cloud Map condición, consulte las claves de condición AWS Cloud Map en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por AWS Cloud Map.

AWS Cloud Map admite las siguientes claves de condición específicas del servicio que puede utilizar para proporcionar un filtrado detallado a sus políticas de IAM.

`servicediscovery:NamespaceArn`

Un filtro que le permite obtener los objetos especificando el nombre de recurso de Amazon (ARN) para el espacio de nombres relacionado.

`servicediscovery:NamespaceName`

Un filtro que le permite obtener objetos especificando el nombre del espacio de nombres relacionado.

`servicediscovery:ServiceArn`

Un filtro que le permite obtener los objetos especificando el nombre de recurso de Amazon (ARN) para los servicios relacionados.

`servicediscovery:ServiceName`

Un filtro que le permite obtener los objetos especificando el nombre del servicio relacionado.

Para ver ejemplos de políticas basadas en la identidad, consulte. AWS Cloud Map Ejemplos de políticas basadas en la identidad para AWS Cloud Map

ACLs in AWS Cloud Map

Soporta ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con AWS Cloud Map

Admite ABAC (etiquetas en las políticas): sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Utilizar credenciales temporales con AWS Cloud Map

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de roles, consulte Cambio de un usuario a un rol de IAM (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Sesiones de acceso directo para AWS Cloud Map

Admite sesiones de acceso directo (FAS): sí

Cuando utiliza un usuario o un rol de IAM para realizar acciones en AWSél, se le considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para AWS Cloud Map

Compatible con roles de servicio: No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de AWS Cloud Map . Edite las funciones de servicio solo cuando se AWS Cloud Map proporcionen instrucciones para hacerlo.

Funciones vinculadas al servicio para AWS Cloud Map

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en la identidad para AWS Cloud Map

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de AWS Cloud Map . Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos AWS Cloud Map, incluido el formato ARNs de cada uno de los tipos de recursos, consulte las claves de condición, recursos y acciones de AWS Cloud Map la Referencia de autorización de servicios.

Temas

- Prácticas recomendadas sobre las políticas
- Mediante la consola de AWS Cloud Map
- AWS Cloud Map ejemplo de acceso a la consola
- Permita que AWS Cloud Map los usuarios vean sus propios permisos
- Permita el acceso de lectura a todos los recursos AWS Cloud Map
- AWS Cloud Map ejemplo de instancia de servicio
- Ejemplo de creación AWS Cloud Map de servicio
- Ejemplo de creación AWS Cloud Map de espacios de nombres

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear AWS Cloud Map recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las

políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Mediante la consola de AWS Cloud Map

Para acceder a la AWS Cloud Map consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los AWS Cloud Map recursos de su cuenta

Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la AWS Cloud Map consola, adjunte también la política `ReadOnly` AWS gestionada AWS Cloud Map `ConsoleAccess` o la política gestionada a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

AWS Cloud Map ejemplo de acceso a la consola

Para conceder acceso total a la AWS Cloud Map consola, debes conceder los permisos de la siguiente política de permisos:

JSON

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:*",
            "route53:GetHostedZone",
            "route53:ListHostedZonesByName",
            "route53:CreateHostedZone",
            "route53:DeleteHostedZone",
            "route53:ChangeResourceRecordSets",
            "route53:CreateHealthCheck",
            "route53:GetHealthCheck",
            "route53:DeleteHealthCheck",
            "route53:UpdateHealthCheck",
            "ec2:DescribeInstances",
            "ec2:DescribeVpcs",
            "ec2:DescribeRegions"
         ],
         "Resource":"*"
```

```
        }
    ]
}
```

Aquí se explica por qué son necesarios los permisos:

`servicediscovery:*`

Permite realizar todas las AWS Cloud Map acciones.

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

Permite AWS Cloud Map administrar las zonas alojadas al crear y eliminar espacios de nombres DNS públicos y privados.

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

AWS Cloud Map Gestionamos las comprobaciones de estado cuando incluye las comprobaciones de estado de Amazon Route 53 al crear un servicio.

`ec2:DescribeVpcs` y **`ec2:DescribeRegions`**

Permita AWS Cloud Map administrar las zonas alojadas privadas.

Permita que AWS Cloud Map los usuarios vean sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

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

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

Permita el acceso de lectura a todos los recursos AWS Cloud Map

La siguiente política de permisos concede al usuario acceso de solo lectura a todos los recursos de AWS Cloud Map :

JSON

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

```
      }
   ]
}
```

AWS Cloud Map ejemplo de instancia de servicio

El siguiente ejemplo muestra una política de permisos que concede a un usuario permiso para registrar, anular el registro y descubrir instancias de servicio. El `Sid` o ID de instrucción es opcional:

JSON

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

La política concede permisos para las acciones que son necesarias con el fin de registrar y administrar las instancias de servicio. El permiso de Route 53 es necesario si utilizas espacios de nombres DNS públicos o privados, ya que AWS Cloud Map crea, actualiza y elimina los registros

y las comprobaciones de estado de Route 53 al registrar y anular el registro de las instancias. El carácter comodín (*) `Resource` permite el acceso a todas las AWS Cloud Map instancias y a los registros y comprobaciones de estado de Route 53 que son propiedad de la cuenta corriente. AWS

Ejemplo de creación AWS Cloud Map de servicio

Al añadir una política de permisos para permitir que una identidad de IAM cree un AWS Cloud Map servicio, debe especificar el nombre de recurso de Amazon (ARN) del espacio de nombres y del servicio en AWS Cloud Map el campo de recursos. El ARN incluye la región, el ID de cuenta y el ID del espacio de nombres. Como todavía no sabrás cuál es el identificador de servicio del servicio, te recomendamos que utilices un comodín. El siguiente es un ejemplo de fragmento de política.

JSON

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

Ejemplo de creación AWS Cloud Map de espacios de nombres

La siguiente política de permisos permite a los usuarios crear todo tipo de espacios de AWS Cloud Map nombres:

JSON

```
{
```

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

AWS políticas administradas para AWS Cloud Map

Una política AWS administrada es una política independiente creada y administrada por AWS. AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

AWS política gestionada: AWSCloud MapDiscoverInstanceAccess

Puede adjuntar `AWSCloudMapDiscoverInstanceAccess` a sus entidades de IAM. Proporciona acceso a la API AWS Cloud Map Discovery.

Para ver los permisos de esta política, consulte AWSCloudMapDiscoverInstanceAccess en la Referencia de la política administrada de AWS .

AWS política gestionada: AWSCloud MapReadOnlyAccess

Puede adjuntar `AWSCloudMapReadOnlyAccess` a sus entidades de IAM. Otorga acceso de solo lectura a todas las AWS Cloud Map acciones.

Para ver los permisos de esta política, consulte AWSCloudMapReadOnlyAccess en la Referencia de la política administrada de AWS .

AWS política gestionada: AWSCloud MapRegisterInstanceAccess

Puede adjuntar `AWSCloudMapRegisterInstanceAccess` a sus entidades de IAM. Concede acceso de solo lectura a espacios de nombres y servicios; además, concede permiso para registrar y anular el registro de instancias de servicio.

Para ver los permisos de esta política, consulte AWSCloudMapRegisterInstanceAccess en la Referencia de la política administrada de AWS .

AWS política gestionada: AWSCloud MapFullAccess

Puede adjuntar `AWSCloudMapFullAccess` a sus entidades de IAM. Proporciona acceso completo a todas las AWS Cloud Map acciones

Para ver los permisos de esta política, consulte AWSCloudMapFullAccess en la Referencia de la política administrada de AWS .

AWS Cloud Map actualizaciones de las políticas AWS gestionadas

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas AWS Cloud Map desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios, suscríbase a la fuente RSS de la página del historial del AWS Cloud Map documento.

Cambio	Descripción	Fecha
AWSCloudMapDiscoverInstanceAccess, AWSCloudMapRegisterInstanceAccess, AWSCloudMapReadOnlyAccess— Actualizaciones de las políticas existentes.	AWS Cloud Map actualizó estas políticas para proporcionar acceso a las nuevas operaciones de la AWS Cloud Map `DiscoverInstanceRevision` API.	15 de agosto de 2023

AWS Cloud Map Referencia de permisos de API

Cuando configuras el control de acceso y escribes una política de permisos que puedas adjuntar a una identidad de IAM (políticas basadas en la identidad), puedes usar la siguiente lista como referencia. La lista incluye cada acción de la AWS Cloud Map API y las acciones a las que debes conceder permisos de acceso. Las acciones se especifican en el `Action` campo de la política. Para obtener más información sobre el valor del recurso que debe especificar en el `Resource` campo o en la política de IAM, consulte las claves de condición, recursos y acciones de AWS Cloud Map la Referencia de autorización de servicios.

Puede utilizar claves de AWS Cloud Map condición específicas en sus políticas de IAM para algunas operaciones. Para obtener más información, consulte Condition keys for AWS Cloud Map en la Referencia de autorizaciones de servicio.

Para especificar una acción, utilice el prefijo `servicediscovery` seguido del nombre de acción de la API; por ejemplo, `servicediscovery:CreatePublicDnsNamespace` y `route53:CreateHostedZone`.

Permisos necesarios para las acciones de AWS Cloud Map

CreateHttpNamespace

Permisos necesarios (acción de API):

- `servicediscovery:CreateHttpNamespace`

CreatePrivateDnsNamespace

Permisos necesarios (acción de la API):

- `servicediscovery:CreatePrivateDnsNamespace`

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

CreatePublicDnsNamespace

Permisos necesarios (acción de la API):

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

CreateService

Permisos necesarios (acción de la API): `servicediscovery:CreateService`

DeleteNamespace

Permisos necesarios (acción de la API):

- `servicediscovery:DeleteNamespace`

DeleteService

Permisos necesarios (acción de la API): `servicediscovery:DeleteService`

DeleteServiceAttributes

Permisos necesarios (acción de la API): `servicediscovery:DeleteServiceAttributes`

DeregisterInstance

Permisos necesarios (acción de la API):

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

DiscoverInstances

Permisos necesarios (acción de la API): `servicediscovery:DiscoverInstances`

GetInstance

Permisos necesarios (acción de la API): `servicediscovery:GetInstance`

GetInstancesHealthStatus

Permisos necesarios (acción de la API): `servicediscovery:GetInstancesHealthStatus`

GetNamespace

Permisos necesarios (acción de la API): `servicediscovery:GetNamespace`

GetOperation

Permisos necesarios (acción de la API): `servicediscovery:GetOperation`

GetService

Permisos necesarios (acción de la API): `servicediscovery:GetService`

GetServiceAttributes

Permisos necesarios (acción de la API): `servicediscovery:GetServiceAttributes`

ListInstances

Permisos necesarios (acción de la API): `servicediscovery:ListInstances`

ListNamespaces

Permisos necesarios (acción de la API): `servicediscovery:ListNamespaces`

ListOperations

Permisos necesarios (acción de la API): `servicediscovery:ListOperations`

ListServices

Permisos necesarios (acción de la API): `servicediscovery:ListServices`

ListTagsForResource

Permisos necesarios (acción de la API): `servicediscovery:ListTagsForResource`

RegisterInstance

Permisos necesarios (acción de la API):

- `servicediscovery:RegisterInstance`
- `route53:GetHealthCheck`
- `route53:CreateHealthCheck`

- `route53:UpdateHealthCheck`
- `ec2:DescribeInstances`

TagResource

Permisos necesarios (acción de la API): `servicediscovery:TagResource`

UntagResource

Permisos necesarios (acción de la API): `servicediscovery:UntagResource`

UpdateHttpNamespace

Permisos necesarios (acción de la API): `servicediscovery:UpdateHttpNamespace`

UpdateInstanceCustomHealthStatus

Permisos necesarios (acción de la API): `servicediscovery:UpdateInstanceCustomHealthStatus`

UpdatePrivateDnsNamespace

Permisos necesarios (acción de la API):

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

UpdatePublicDnsNamespace

Permisos necesarios (acción de la API):

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

UpdateService

Permisos necesarios (acción de la API):

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

UpdateServiceAttributes

Permisos necesarios (acción de la API): `servicediscovery:UpdateServiceAttributes`

Solución de problemas AWS Cloud Map de identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar problemas comunes que pueden surgir al trabajar con un AWS Cloud Map IAM.

Temas

- No estoy autorizado a realizar ninguna acción en AWS Cloud Map
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis AWS Cloud Map recursos

No estoy autorizado a realizar ninguna acción en AWS Cloud Map

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `servicediscovery:`*`GetWidget`*.

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

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `servicediscovery:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a AWS Cloud Map.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS Cloud Map. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis AWS Cloud Map recursos

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que respaldan las políticas basadas en recursos o las listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si AWS Cloud Map es compatible con estas funciones, consulte. ¿Cómo AWS Cloud Map funciona con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Validación de conformidad para AWS Cloud Map

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la HIPAA.
- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos

de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.

- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en AWS Cloud Map

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

AWS Cloud Map es principalmente un servicio global. Sin embargo, puede usarlo AWS Cloud Map para crear comprobaciones de estado de Route 53 que comprueben el estado de los recursos en regiones específicas, como las EC2 instancias de Amazon y los balanceadores de carga de Elastic Load Balancing.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura AWS global.

Seguridad de la infraestructura en AWS Cloud Map

Como servicio gestionado, AWS Cloud Map está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a AWS Cloud Map través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puede mejorar la seguridad de su VPC configurándola AWS Cloud Map para usar un punto final de VPC de interfaz. Para obtener más información, consulte Acceso AWS Cloud Map mediante un punto final de interfaz (AWS PrivateLink).

Acceso AWS Cloud Map mediante un punto final de interfaz (AWS PrivateLink)

Puede usarlo AWS PrivateLink para crear una conexión privada entre su VPC y. AWS Cloud Map Puede acceder AWS Cloud Map como si estuviera en su VPC, sin el uso de una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o AWS Direct Connect una conexión. Las instancias de la VPC no necesitan direcciones IP públicas para acceder a AWS Cloud Map.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a AWS Cloud Map.

Para obtener más información, consulte Acceso a los Servicios de AWS a través de AWS PrivateLink en la Guía de AWS PrivateLink .

Consideraciones sobre AWS Cloud Map

Antes de configurar un punto final de interfaz para AWS Cloud Map, consulte las consideraciones de la guía.AWS PrivateLink

Si su Amazon VPC no tiene una puerta de enlace a Internet y sus tareas utilizan el controlador de registro para enviar la información de `awslogs` registro a CloudWatch Logs, debe crear un punto de enlace de VPC de interfaz para Logs. CloudWatch Para obtener más información, consulte Uso de CloudWatch registros con puntos de enlace de VPC de interfaz en la Guía del usuario de Amazon CloudWatch Logs.

Los puntos de enlace de VPC no admiten AWS solicitudes entre regiones. Asegúrese de crear su punto de conexión en la misma región en la que tiene previsto enviar llamadas a la API de AWS Cloud Map.

Los puntos de conexión de VPC solo admiten DNS proporcionadas por Amazon a través de Amazon Route 53. Si desea utilizar su propio DNS, puede utilizar el enrutamiento de DNS condicional. Para obtener más información, consulte Conjuntos de opciones de DHCP en la Guía del usuario de Amazon VPC.

El grupo de seguridad asociado al punto de conexión de VPC debe permitir las conexiones entrantes en el puerto 443 desde la subred privada de Amazon VPC.

Cree un punto final de interfaz para AWS Cloud Map

Puede crear un punto final de interfaz para AWS Cloud Map usar la consola de Amazon VPC o AWS Command Line Interface ()AWS CLI. Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

Cree un punto final de interfaz para AWS Cloud Map utilizar los siguientes nombres de servicio:

> **Note**
>
> La API `DiscoverInstances` no estará disponible en estos dos puntos de conexión.

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

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

Cree un punto final de interfaz para que el plano de AWS Cloud Map datos acceda a la `DiscoverInstances` API con los siguientes nombres de servicio:

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

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

> **Note**
>
> Deberá deshabilitar la inyección de prefijos de host cuando llame a `DiscoverInstances` con los nombres de DNS de VPCE regionales o de zona para los puntos de conexión del plano de datos. Al llamar a cada operación de API, anteponga al punto final del servicio

varios prefijos de host, lo que produce URL no válidas cuando se especifica un punto final de VPC. AWS CLI AWS SDKs

Si habilita DNS privado para el punto de conexión de interfaz, puede realizar solicitudes a la API para AWS Cloud Map usando su nombre de DNS predeterminado para la región. Por ejemplo, `servicediscovery.us-east-1.amazonaws.com`.

La AWS PrivateLink conexión VPCE se admite en todas las regiones en las que AWS Cloud Map sea compatible; sin embargo, el cliente debe comprobar qué zonas de disponibilidad admiten la VPCE antes de definir un punto final. Para saber qué zonas de disponibilidad son compatibles con los puntos finales de la interfaz de VPC en una región, utilice el describe-vpc-endpoint-services comando o utilice el. AWS Management Console Por ejemplo, los siguientes comandos devuelven las zonas de disponibilidad en las que puede implementar puntos de conexión de VPC de una interfaz AWS Cloud Map dentro de la región Este de EE. UU. (Ohio):

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

Monitorización AWS Cloud Map

La monitorización es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de sus soluciones de AWS . Debe recopilar los datos de supervisión de todas las partes de la AWS solución para poder depurar más fácilmente un error multipunto en caso de que se produzca. No obstante, antes de comenzar a monitorizar, debe crear un plan de monitorización que incluya respuestas a las siguientes preguntas:

- ¿Cuáles son los objetivos de la supervisión?
- ¿Qué recursos va a supervisar?
- ¿Con qué frecuencia va a supervisar estos recursos?
- ¿Qué herramientas de supervisión va a utilizar?
- ¿Quién se encargará de realizar las tareas de supervisión?
- ¿Quién debería recibir una notificación cuando surjan problemas?

Temas

- Registra las llamadas a la AWS Cloud Map API mediante AWS CloudTrail

Registra las llamadas a la AWS Cloud Map API mediante AWS CloudTrail

AWS Cloud Map está integrado con AWS CloudTrailun servicio que proporciona un registro de las acciones realizadas por un usuario, rol o un Servicio de AWS. CloudTrail captura todas las llamadas a la API AWS Cloud Map como eventos. Las llamadas capturadas incluyen llamadas desde la AWS Cloud Map consola y llamadas en código a las operaciones de la AWS Cloud Map API. Con la información recopilada por CloudTrail, puede determinar a qué solicitud se realizó AWS Cloud Map, la dirección IP desde la que se realizó la solicitud, cuándo se realizó y detalles adicionales.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario.
- Si la solicitud se realizó en nombre de un usuario de IAM Identity Center.

- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro Servicio de AWS.

CloudTrail está activa en tu cuenta Cuenta de AWS al crear la cuenta y automáticamente tienes acceso al historial de CloudTrail eventos. El historial de CloudTrail eventos proporciona un registro visible, consultable, descargable e inmutable de los últimos 90 días de eventos de gestión registrados en un. Región de AWSPara obtener más información, consulte Uso del historial de CloudTrail eventos en la Guía del usuario.AWS CloudTrail La visualización del historial de eventos no conlleva ningún CloudTrail cargo.

Para tener un registro continuo de los eventos de Cuenta de AWS los últimos 90 días, crea un almacén de datos de eventos de senderos o CloudTraillagos.

CloudTrail senderos

Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. Todos los senderos creados con él AWS Management Console son multirregionales. Puede crear un registro de seguimiento de una sola región o de varias regiones mediante la AWS CLI. Se recomienda crear un sendero multirregional, ya que puedes capturar toda la actividad de tu Regiones de AWS cuenta. Si crea un registro de seguimiento de una sola región, solo podrá ver los eventos registrados en la Región de AWS del registro de seguimiento. Para obtener más información acerca de los registros de seguimiento, consulte Creación de un registro de seguimiento para su Cuenta de AWS y Creación de un registro de seguimiento para una organización en la Guía del usuario de AWS CloudTrail .

Puede enviar una copia de sus eventos de administración en curso a su bucket de Amazon S3 sin coste alguno CloudTrail mediante la creación de una ruta; sin embargo, hay cargos por almacenamiento en Amazon S3. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios. Para obtener información acerca de los precios de Amazon S3, consulte Precios de Amazon S3.

CloudTrail Almacenes de datos de eventos en Lake

CloudTrail Lake le permite ejecutar consultas basadas en SQL en sus eventos. CloudTrail Lake convierte los eventos existentes en formato JSON basado en filas al formato Apache ORC. ORC es un formato de almacenamiento en columnas optimizado para una recuperación rápida de datos. Los eventos se agregan en almacenes de datos de eventos, que son recopilaciones inmutables de eventos en función de criterios que se seleccionan aplicando selectores de eventos

avanzados. Los selectores que se aplican a un almacén de datos de eventos controlan los eventos que perduran y están disponibles para la consulta. Para obtener más información sobre CloudTrail Lake, consulte Cómo trabajar con AWS CloudTrail Lake en la Guía del AWS CloudTrail usuario.

CloudTrail Los almacenes de datos y las consultas sobre eventos de Lake conllevan costes. Cuando crea un almacén de datos de eventos, debe elegir la opción de precios que desee utilizar para él. La opción de precios determina el costo de la incorporación y el almacenamiento de los eventos, así como el período de retención predeterminado y máximo del almacén de datos de eventos. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios.

AWS Cloud Map eventos de datos en CloudTrail

Los eventos de datos proporcionan información sobre las operaciones de recursos que se realizan en un recurso o dentro de él (por ejemplo, descubrir una instancia registrada en un espacio de nombres). Se denominan también operaciones del plano de datos. Los eventos de datos suelen ser actividades de gran volumen. De forma predeterminada, CloudTrail no registra los eventos de datos. El historial de CloudTrail eventos no registra los eventos de datos.

Se aplican cargos adicionales a los eventos de datos. Para obtener más información sobre CloudTrail los precios, consulta AWS CloudTrail Precios.

Puede registrar eventos de datos para los tipos de AWS Cloud Map recursos mediante la CloudTrail consola o las operaciones de la CloudTrail API. AWS CLI Para obtener más información sobre cómo registrar los eventos de datos, consulte Registro de eventos de datos con la AWS Management Console y Registro de eventos de datos con la AWS Command Line Interface en la Guía del usuario de AWS CloudTrail .

En la siguiente tabla se enumeran los tipos de AWS Cloud Map recursos para los que puede registrar eventos de datos. La columna Tipo de evento de datos (consola) muestra el valor que se puede elegir en la lista de tipos de eventos de datos de la CloudTrail consola. La columna de valores resources.type muestra el `resources.type` valor que se debe especificar al configurar los selectores de eventos avanzados mediante o. AWS CLI CloudTrail APIs La CloudTrail columna Datos APIs registrados muestra las llamadas a la API registradas CloudTrail para el tipo de recurso.

Tipo de evento de datos (consola)	resources.type value	Datos APIs registrados en CloudTrail
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances • DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • DiscoverInstancesRevision

Puede configurar selectores de eventos avanzados para filtrar según los campos `eventName`, `readOnly` y `resources.ARN` y así registrar solo los eventos que son importantes para usted. Para obtener más información sobre estos campos, consulte AdvancedFieldSelector en la Referencia de la API de AWS CloudTrail .

El siguiente ejemplo muestra cómo configurar selectores de eventos avanzados para registrar todos los eventos de AWS Cloud Map datos.

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

AWS Cloud Map eventos de administración en CloudTrail

Los eventos de administración proporcionan información sobre las operaciones de administración que se realizan en los recursos de su empresa Cuenta de AWS. Se denominan también operaciones del plano de control. De forma predeterminada, CloudTrail registra los eventos de administración.

AWS Cloud Map registra todas las operaciones del plano de AWS Cloud Map control como eventos de administración. Para obtener una lista de las operaciones del plano de AWS Cloud Map control en las que se AWS Cloud Map registra CloudTrail, consulte la referencia de la AWS Cloud Map API.

AWS Cloud Map ejemplos de eventos

Un evento representa una solicitud única de cualquier fuente e incluye información sobre la operación de API solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que los eventos no aparecen en ningún orden específico.

El siguiente ejemplo muestra un evento CloudTrail de administración que demuestra la `CreateHTTPNamespace` operación.


```
{
    "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"
}
```

El siguiente ejemplo muestra un evento CloudTrail de datos que demuestra la `DiscoverInstances` operación.

```
{
            "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"
```

```
        }
```

Para obtener información sobre el contenido de los CloudTrail registros, consulte el contenido de los CloudTrail registros en la Guía del AWS CloudTrail usuario.

Etiquetar sus recursos AWS Cloud Map

Una etiqueta es una etiqueta que se asigna a un AWS recurso. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario.

Las etiquetas te permiten clasificar tus AWS recursos, por ejemplo, por su propósito, propietario o entorno. Cuando tenga muchos recursos del mismo tipo, puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. Por ejemplo, puedes definir un conjunto de etiquetas para tus AWS Cloud Map servicios que te ayuden a realizar un seguimiento del propietario y del nivel de pila de cada servicio. Le recomendamos que diseñe un conjunto coherente de claves de etiqueta para cada tipo de recurso.

Además, las etiquetas no se asignan a los recursos automáticamente. Después de agregar una etiqueta, puede editar las claves y los valores de las etiquetas o eliminar etiquetas de un recurso en cualquier momento. Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.

Las etiquetas no tienen ningún significado semántico AWS Cloud Map y se interpretan estrictamente como una cadena de caracteres. Puede establecer el valor de una etiqueta como una cadena vacía, pero no puede asignarle un valor nulo. Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo.

Puede trabajar con etiquetas mediante la AWS Management Console AWS CLI, la y la AWS Cloud Map API.

Si utilizas AWS Identity and Access Management (IAM), puedes controlar qué usuarios de tu AWS cuenta tienen permiso para crear, editar o eliminar etiquetas.

Cómo se etiquetan los recursos

Puede etiquetar AWS Cloud Map espacios de nombres y servicios nuevos o existentes.

Si utilizas la AWS Cloud Map consola, puedes aplicar etiquetas a los recursos nuevos cuando se creen o a los recursos existentes en cualquier momento mediante la pestaña Etiquetas de la página de recursos correspondiente.

Si utilizas la AWS Cloud Map API, el SDK o un AWS SDK AWS CLI, puedes aplicar etiquetas a los nuevos recursos mediante el `tags` parámetro de la acción de API correspondiente o a los recursos

existentes mediante la acción de la TagResourceAPI. Para obtener más información, consulte TagResource.

Además, algunas acciones de creación de recursos le permiten especificar etiquetas para un recurso al crearlo. Si no se pueden aplicar etiquetas durante la creación del recurso, el proceso de creación de recursos falla. Esto garantiza que los recursos que pretendía etiquetar en el momento de su creación se creen con etiquetas específicas o no se creen en absoluto. Si etiqueta recursos en el momento de su creación, no es necesario ejecutar scripts de etiquetado personalizados después de la creación del recurso.

En la siguiente tabla se describen los AWS Cloud Map recursos que se pueden etiquetar y los recursos que se pueden etiquetar al crearlos.

Soporte de etiquetado para los recursos AWS Cloud Map

Recurso	Admite etiquetas	Admite la propagación de etiquetas	Admite el etiquetado en el momento de la creación (AWS Cloud Map API AWS CLI, AWS SDK)
AWS Cloud Map espacios de nombres	Sí	No. Las etiquetas del espacio de nombres no se propagan a ningún otro recurso asociado al espacio de nombres.	Sí
AWS Cloud Map servicios	Sí	No. Las etiquetas de servicio no se propagan a ningún otro recurso asociado al servicio.	Sí

Restricciones

Se aplican las siguientes restricciones básicas a las etiquetas:

- Número máximo de etiquetas para cada recurso: 50.
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.
- Longitud máxima de la clave: 128 caracteres Unicode en UTF-8
- Longitud máxima del valor: 256 caracteres Unicode en UTF-8
- Si su esquema de etiquetado se usa en varios AWS servicios y recursos, recuerde que otros servicios pueden tener restricciones en cuanto a los caracteres permitidos. Los caracteres permitidos generalmente son: letras, números y espacios representables en UTF-8, además de los siguientes caracteres: + - = . _ : / @.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No utilices `aws:AWS:`, ni ninguna combinación de mayúsculas o minúsculas, como prefijo para las claves o los valores, ya que está reservado para su uso. AWS Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas con este prefijo no se tienen en cuenta para el límite. tags-per-resource

Actualizar las etiquetas de los recursos AWS Cloud Map

Usa los siguientes AWS CLI comandos u operaciones de AWS Cloud Map API para agregar, actualizar, enumerar y eliminar las etiquetas de tus recursos.

Soporte de etiquetado para los recursos AWS Cloud Map

Tarea	Acción de la API	AWS CLI	AWS Tools for Windows PowerShell
Agregar o sobrescribir una o varias etiquetas.	TagResource	tag-resource	Agregar- Etiquetar SDResource
Eliminar una o varias etiquetas.	UntagResource	untag-resource	Eliminar SDResource etiqueta
Enumerar las etiquetas de un recurso	ListTagsForResource	list-tags-for-resource	Obtener- SDResource Etiqueta

Los siguientes ejemplos muestran cómo agregar o quitar etiquetas a los recursos mediante la AWS CLI.

Ejemplo 1: Etiquetar un recurso existente

El siguiente comando etiqueta un recurso existente.

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

Ejemplo 2: Eliminar la etiqueta de un recurso existente

El siguiente comando elimina una etiqueta de un recurso existente.

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

Ejemplo 3: enumerar etiquetas de un recurso

El siguiente comando enumera las etiquetas asociadas a un recurso existente.

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

Algunas acciones de creación de recursos le permiten especificar etiquetas al crear el recurso. Las siguientes acciones admiten etiquetado durante la creación.

Tarea	Acción de la API	AWS CLI	AWS Tools for Windows PowerShell
Crear un espacio de nombres de HTTP	CreateHttpNamespace	create-http-namespace	Nuevo: SDHttp espacio de nombres
Crear un espacio de nombres privado basado en DNS	CreatePrivateDnsNamespace	create-private-dns-namespace	Nuevo- SDPrivateDnsNamespace
Crear un espacio de nombres público basado en DNS	CreatePublicDnsNamespace	create-public-dns-namespace	Nuevo- SDPublicDnsNamespace
Crear un servicio	CreateService	create-service	Nuevo- SDService

AWS Cloud Map cuotas de servicio

AWS Cloud Map los recursos están sujetos a las siguientes cuotas de servicio a nivel de cuenta. Cada cuota de la lista se aplica a cada AWS región en la que se crean AWS Cloud Map los recursos.

Nombre	Valor predeterminado	Ajusta	Descripción
Atributos personalizados por instancia	Cada región admitida: 30	No	El número máximo de atributos personalizados que puede especificar al registrar una instancia.
DiscoverInstances tasa de ráfaga de operaciones por cuenta	Cada región admitida: 2000	Sí	La velocidad máxima de ráfaga para llamar a una DiscoverInstances operación desde una sola cuenta.
DiscoverInstances operación por cuenta (tasa constante)	Cada región admitida: 1000	Sí	La tasa máxima constante para realizar DiscoverInstances llamadas desde una sola cuenta.
DiscoverInstancesRevision tasa de operación por cuenta	Cada región admitida: 3000	Sí	La tarifa máxima para realizar DiscoverI nstancesRevision llamadas desde una sola cuenta.
Instancias por espacio de nombres	Cada región admitida: 2000	Sí	El número máximo de instancias de servicio que puede registrar con el mismo espacio de nombres.

Nombre	Valor predeterminado	Ajusta	Descripción
Instancias por servicio	Cada región admitida: 1000	No	El número máximo de instancias que puede registrar en una región con el mismo servicio.
Espacios de nombres por región	Cada región admitida: 50	Sí	El número máximo de espacios de nombres que puede crear por región.

* Cuando se crea un espacio de nombres, se crea automáticamente una zona alojada de Amazon Route 53. Esta zona alojada se descuenta de la cuota del número de zonas alojadas que puedes crear con una AWS cuenta. Para obtener más información, consulte Cuotas en zonas alojadas en la Guía para desarrolladores de Amazon Route 53.

** Aumentar las instancias de los espacios de nombres de DNS para AWS Cloud Map requiere un aumento del límite de registros por zona alojada de Route 53, lo que conlleva cargos adicionales.

Administrar tus cuotas AWS Cloud Map de servicio

AWS Cloud Map se ha integrado con Service Quotas, un AWS servicio que le permite ver y gestionar sus cuotas desde una ubicación central. Para obtener más información, consulte ¿Qué es Service Quotas? en la Guía del usuario de Service Quotas.

Service Quotas facilita la búsqueda del valor de sus cuotas de AWS Cloud Map servicio.

AWS Management Console

Para ver las cuotas AWS Cloud Map de servicio mediante el AWS Management Console

1. Abra la consola de Service Quotas en https://console.aws.amazon.com/servicequotas/.
2. En el panel de navegación, elija Servicios de AWS .
3. En la lista Servicios de AWS , busque y seleccione AWS Cloud Map.
4. En la lista de cuotas de servicio AWS Cloud Map, puede ver el nombre de la cuota de servicio, el valor aplicado (si está disponible), la cuota AWS predeterminada y si el valor de la cuota es ajustable.

Para ver información adicional sobre una cuota de servicio, como la descripción, elija el nombre de la cuota para que aparezcan los detalles de la cuota.

5. (Opcional) Para solicitar un aumento de cuota, seleccione la cuota que desee aumentar y elija Solicitar aumento a nivel de cuenta.

Para trabajar más con las cuotas de servicio, AWS Management Console consulte la Guía del usuario de Service Quotas.

AWS CLI

Para ver las cuotas AWS Cloud Map de servicio mediante el AWS CLI

Ejecute el siguiente comando para ver las AWS Cloud Map cuotas predeterminadas.

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

Ejecute el siguiente comando para ver AWS Cloud Map las cuotas aplicadas.

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

Para obtener más información sobre cómo trabajar con cuotas de servicio mediante el AWS CLI, consulte la Referencia de AWS CLI comandos de Service Quotas. Para solicitar un aumento de cuota, consulte el `request-service-quota-increase` comando en la Referencia de comandos de la AWS CLI.

Gestiona la limitación de las solicitudes de AWS Cloud Map DiscoverInstances API

AWS Cloud Map limita las solicitudes de DiscoverInstancesAPI para cada AWS cuenta por región. La limitación ayuda a mejorar el rendimiento del servicio y a garantizar un uso justo para todos los clientes. AWS Cloud Map La limitación garantiza que las llamadas a la AWS Cloud Map DiscoverInstancesAPI no superen las cuotas máximas de solicitudes de API permitidas

DiscoverInstances. DiscoverInstances Las llamadas a la API que se originan en cualquiera de las siguientes fuentes están sujetas a las cuotas de solicitudes:

- Una aplicación de terceros
- Una herramienta de línea de comandos
- ¿La AWS Cloud Map consola

Si supera una cuota de limitación de la API, aparece el código de error `RequestLimitExceeded`. Para obtener más información, consulte the section called “Limitación de velocidad de solicitudes”.

Cómo se aplica la limitación

AWS Cloud Map utiliza el algoritmo token bucket para implementar la regulación de la API. Con este algoritmo, su cuenta tiene un bucket que contiene un número específico de tokens. El número de tokens del bucket representa su cuota de limitación en un segundo determinado. Hay un bucket para una sola región, y este se aplica a todos los puntos de conexión de la región.

Limitación de velocidad de solicitudes

La limitación limita el número de solicitudes a la DiscoverInstancesAPI que puedes realizar. Cada solicitud elimina un token del bucket. Por ejemplo, el tamaño del depósito para la operación de la DiscoverInstancesAPI es de 2000 tokens, por lo que puedes realizar hasta 2000 DiscoverInstancessolicitudes en un segundo. Si superan las 2000 solicitudes en un segundo, estará limitado y las solicitudes restantes dentro de ese segundo fallarán.

Los buckets se recargan automáticamente a una tasa fija. Si el bucket no ha alcanzado su capacidad máxima, se vuelve a agregar un número determinado de tokens cada segundo hasta que el bucket alcance su capacidad máxima. Si el bucket ha alcanzado su capacidad máxima cuando llegan los tokens de recarga, estos tokens se descartan. El tamaño del depósito para la operación de la DiscoverInstancesAPI es de 2000 fichas y la tasa de recarga es de 1000 fichas por segundo. Si realizas 2000 solicitudes a la DiscoverInstancesAPI en un segundo, el depósito se reduce inmediatamente a cero (0) tokens. A continuación, el bucket se recarga con hasta 1000 tokens por segundo hasta alcanzar su capacidad máxima de 2000 tokens.

Puede usar los tokens a medida que se vayan agregando al bucket. No tiene que esperar a que el bucket esté al máximo de su capacidad para realizar solicitudes de la API. Si agotas el depósito realizando 2000 solicitudes de DiscoverInstancesAPI en un segundo, podrás seguir realizando hasta 1000 solicitudes de DiscoverInstancesAPI por segundo durante el tiempo que necesites. Esto

significa que puede utilizar inmediatamente los tokens de recarga a medida que se vayan agregando a su bucket. El bucket solo comienza a recargarse hasta su capacidad máxima cuando realice menos solicitudes de API por segundo que la tasa de recarga.

Reintentos o procesamiento por lotes

Si se produce un error en una solicitud de la API, es posible que la aplicación tenga que volver a intentarlo. Para reducir el número de solicitudes de la API, use un intervalo de suspensión entre solicitudes sucesivas adecuado. Para obtener resultados óptimos, utilice un intervalo de suspensión creciente o variable.

Cálculo del intervalo de suspensión

Cuando tenga que sondear o reintentar una solicitud de API, recomendamos que utilice un algoritmo de retardo exponencial para calcular el intervalo de suspensión entre las llamadas al API. Al utilizar tiempos de espera cada vez más largos entre reintentos para las respuestas a errores consecutivos, puedes reducir el número de solicitudes erróneas. Para obtener más información y ejemplos de implementación de este algoritmo, consulta Retry Behavior en la Guía de referencia de herramientas AWS SDKs y herramientas.

Ajuste de las cuotas de limitación de las API

Puedes solicitar un aumento de las cuotas de limitación de la API para tu cuenta. AWS Para solicitar un ajuste de cuota, póngase en contacto con AWS Support Center.

Historial de documentos para AWS Cloud Map

En la siguiente tabla se describen las principales actualizaciones y nuevas características de la Guía para desarrolladores de AWS Cloud Map . Actualizamos la documentación con frecuencia para dar respuesta a los comentarios que se nos envía.

Cambio	Descripción	Fecha
AWS Cloud Map atributos de servicio	Ahora puede especificar los atributos a nivel de servicio para evitar la duplicación de los atributos en las instancia s que están registradas en un servicio. Puede usar estos atributos para enrutar el tráfico de forma compleja, establece r valores de tiempo de espera y reintento y para coordinar los servicios y las integraciones externas.	13 de diciembre de 2024
Se han añadido tutoriales	Se AWS Cloud Map agregaron dos tutoriales que muestran casos de uso comunes para su uso.	27 de marzo de 2024
CloudTrail documentación de integración actualizada	Se ha actualizado la documentación que describe la AWS Cloud Map integraci ón con la actividad de la API CloudTrail para registrar.	20 de marzo de 2024
Actualizaciones de políticas administradas	Se han actualizado las políticas de `AWSCloudM apDiscoverInstance Access` , `AWSCloudM apRegisterInstance`	20 de septiembre de 2023

	`Access` y `AWSCloudMapReadOnlyAccess`.	
Cloud Map y AWS PrivateLink	Ahora puede usar an AWS PrivateLink para crear una conexión privada entre su VPC y. AWS Cloud Map	15 de septiembre de 2023
Actualización de la política administrada	Se ha actualizado la política `AWSCloudMapDiscoverInstanceAccess`.	15 de agosto de 2023
AWS SDK para Python	Se han agregado ejemplos de línea de comandos de Python.	13 de septiembre de 2022
IPv6 soporte	Los puntos de conexión de la API ahora están disponibles en redes de solo `IPv6`.	28 de enero de 2022
Detección de instancias de servicio	AWS Cloud Map se agregó compatibilidad para crear servicios en un espacio de nombres que admita consultas de DNS que solo se pueden detectar mediante la operación de DiscoverInstancesAPI y no mediante consultas de DNS.	24 de marzo de 2021
Etiquetado de recursos	AWS Cloud Map se ha añadido compatibilidad para añadir etiquetas de metadatos a los espacios de nombres y servicios mediante el. AWS Management Console	8 de febrero de 2021

Etiquetado de recursos	AWS Cloud Map se agregó soporte para agregar etiquetas de metadatos a sus espacios de nombres y servicios mediante el comando y. AWS CLI APIs	22 de junio de 2020
Versión inicial	Esta es la primera versión de la Guía para desarrolladores de AWS Cloud Map .	28 de noviembre de 2018

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.