aws

Guía para desarrolladores

Amazon Cloud Directory

Amazon Cloud Directory: Guía para desarrolladores

Table of Contents

¿Qué es Amazon Cloud Directory?

Amazon Cloud Directory es un almacén de AWS basado en directorios, de varios inquilinos y de alta disponibilidad. Estos directorios se amplían automáticamente a cientos de millones de objetos según precisen las aplicaciones. Esto permite al personal de operaciones centrarse en el desarrollo y la implementación de aplicaciones que hacen que el negocio alcance, no en gestionar la infraestructura de los directorios. A diferencia de los sistemas de directorios tradicionales, Cloud Directory no limita la organización de objetos del directorio a una sola jerarquía fija.

Con Cloud Directory, puede organizar los objetos del directorio en múltiples jerarquías para que pueda haber muchas relaciones y movimiento organizativo en todo el ámbito de la información del directorio. Por ejemplo, un directorio de usuarios puede proporcionar una vista jerárquica basada en estructuras de informes, en su ubicación y en la pertenencia a proyectos. De forma similar, un directorio de dispositivos puede tener varias vistas jerárquicas según su fabricante, su propietario actual y su ubicación física.

El aspecto central de Cloud Directory es que es un almacén de directorios especializado basado en gráficos que proporciona un bloque base sobre el que ir construyendo la infraestructura a los desarrolladores. Con Cloud Directory, los desarrolladores pueden hacer lo siguiente:

- Crear aplicaciones basadas en el directorio de forma fácil y sin tener que preocuparse por la implementación, la escala global, la disponibilidad y el rendimiento.
- Crear aplicaciones que permitan gestionar usuarios y grupos, permisos o políticas, el registro de dispositivos, clientes, libretas de direcciones y catálogos de aplicaciones o productos.
- Definir nuevos objetos de directorio o ampliar los tipos existentes según las necesidades de la aplicación, reduciendo así el código que necesiten escribir.
- Reducir la complejidad de añadir capas de aplicaciones sobre Cloud Directory.
- Gestionar la evolución de la información de los esquemas a lo largo del tiempo para garantizar en el futuro la compatibilidad para los consumidores.

Cloud Directory incluye un conjunto de operaciones de API para acceder a diferentes objetos y políticas almacenados en sus directorios basados en Cloud Directory. Para ver una lista de las operaciones disponibles, consulteAcciones de la API de Amazon Cloud Directory. Para ver la lista de operaciones y los permisos necesarios para realizar cada acción de API, consulte Permisos de la API de Amazon Cloud Directory: Referencia de acciones, recursos y condiciones.

Para ver una lista de las regiones de Cloud Directory admitidas, consulte laRegiones y puntos de enlace de AWS. Para ver otros recursos, consulte Recursos Cloud Directory.

¿Qué no es Cloud Directory?

Cloud Directory no es un servicio de directorio para los administradores de TI que deseen gestionar o migrar su infraestructura de directorios.

Introducción

En este ejercicio introductorio, va a crear un esquema. You then choose to create a directory from that same schema or from any of the sample schemas that are available in the AWS Directory Service console. Aunque no es necesario, le recomendamos que consulte Descripción de los conceptos clave de Cloud Directory antes de empezar a utilizar la consola para que se familiarice con las características y terminología principales.

Temas

- Crear un esquema
- Crear un directorio en Amazon Cloud Directory
- Uso de los puntos de enlace de la VPC de tipo interfaz

Crear un esquema

Amazon Cloud Directory permite cargar un archivo JSON compatible con la creación de esquemas. Para crear un esquema nuevo, puede crear su propio archivo JSON desde cero o descargar uno de los esquemas existentes registrados en la consola. A continuación, cárguelo como un esquema personalizado. Para obtener más información, consulte Esquemas personalizados.

También puede crear, eliminar, descargar, listar, publicar, actualizar y actualizar esquemas mediante las API de Cloud Directory. Para obtener más información sobre las operaciones de API relacionadas con los esquemas, consulte la guía de referencia de API de Amazon Cloud Directory.

Elija uno de los procedimientos que aparece a continuación, según prefiera.

Para crear un esquema personalizado

1. En el navegadorAWS Directory Servicepanel de navegación, enDirectorio en la nube, elijaEsquemas de.
2. Cree un archivo JSON con todas las nuevas definiciones de esquema. Para obtener más información acerca de cómo dar formato a un archivo JSON, consulte Formato de esquemas JSON.
3. En la consola de, elijaCargar nuevo esquema.
4. En el navegadorCargar nuevo esquemaEscriba un nombre para el esquema.

5. SelectSeleccione el archivoSeleccione el nuevo archivo JSON que acaba de crear y, a continuación, elijaOpen.
6. Seleccione Upload. Al hacerlo, se agrega un esquema nuevo a la biblioteca de esquemas y se pone en el cuadro de diálogoDesarrolloestado. Para obtener más información sobre los estados de esquema, consulte Ciclo de vida de esquemas.

Para crear un esquema personalizado basado en otro existente en la consola

1. En el navegadorAWS Directory Servicepanel de navegación, enDirectorio en la nube, elijaEsquemas de.
2. En la tabla donde se enumeran los esquemas, seleccione la opción situada cerca del esquema que quiera copiar.
3. Elija Actions (Acciones).
4. SeleccionarDescargar esquema.
5. Cambie el nombre del archivo JSON, edítelo según sea necesario y, a continuación, guarde el archivo. Para obtener más información acerca de cómo dar formato a un archivo JSON, consulte Formato de esquemas JSON.
6. En la consola de, elijaCargar nuevo esquemaSeleccione el archivo JSON que acaba de editar y, a continuación, elijaOpen.

   Al hacerlo, se agrega un esquema nuevo a la biblioteca de esquemas y se pone en el cuadro de diálogoDesarrolloestado. Para obtener más información sobre los estados de esquema, consulte Ciclo de vida de esquemas.

Crear un directorio en Amazon Cloud Directory

Antes de crear un directorio en Amazon Cloud Directory, AWS Directory Service requiere que le aplique un esquema primero. Un directorio no se puede crear sin un esquema y normalmente tiene un esquema aplicado. No obstante, también puede servirse de las operaciones de API de Cloud Directory para aplicar esquemas adicionales a un directorio. Para obtener más información, consulte ApplySchema en la guía de referencia de API de Amazon Cloud Directory.

Para crear un Cloud Directory

1. En el navegadorAWS Directory Servicepanel de navegación, enDirectorio en la nube, elijaDirectorios.

2. SeleccionarConfigurar Cloud Directory.
3. UNDERElija un esquema para aplicar al nuevo directorio, escriba el nombre sencillo de su directorio, como por ejemplo`User Repository`y, a continuación, elija una de las siguientes opciones:

   - Esquema administrado
   - Esquema de ejemplo
   - Esquema personalizado

   Los esquemas de ejemplo y los esquemas personalizados se colocan en el cuadro de diálogoDesarrollo, de forma predeterminada. Para obtener más información sobre los estados de esquema, consulte Ciclo de vida de esquemas. Antes de que un esquema se pueda aplicar a un directorio, debe convertirse al estado Published (Publicado). Para publicar correctamente un esquema de muestra mediante la consola, debe disponer de permisos para las siguientes acciones:

   - `clouddirectory:Get*`
   - `clouddirectory:List*`
   - `clouddirectory:CreateSchema`
   - `clouddirectory:CreateDirectory`
   - `clouddirectory:PutSchemaFromJson`
   - `clouddirectory:PublishSchema`
   - `clouddirectory:DeleteSchema`

   Dado que los esquemas de muestra son plantillas de solo lectura proporcionadas por AWS, no se pueden publicar directamente. En su lugar, cuando elija crear un directorio en función de un esquema de muestra, la consola crea una copia temporal del esquema de muestra que ha seleccionado y la coloca en el cuadroDesarrolloestado. A continuación, crea una copia de dicho esquema de desarrollo y la coloca en el estado Published (Publicado). Una vez publicado, el esquema de desarrollo se elimina, por lo que la acción `DeleteSchema` es necesaria a la hora de publicar un esquema de muestra.
4. Seleccione Siguiente.
5. Revise la información del directorio y haga los cambios necesarios. Cuando la información sea correcta, elija Create (Crear).

Uso de los puntos de enlace de la VPC de tipo interfaz

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus recursos de AWS, puede establecer una conexión privada entre su VPC y Cloud Directory. Puede utilizar esta conexión para habilitar Cloud Directory para comunicarse con sus recursos en la VPC sin pasar por la Internet pública.

Amazon VPC es un servicio de AWS que puede utilizar para lanzar recursos de AWS en una red virtual que usted defina. Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de ruteo y las gateways de red. Para conectar su VPC a Cloud Directory, debe definir unpunto de enlace de la VPC de la interfazpara Cloud Directory. El punto de enlace ofrece conectividad escalable de confianza con Cloud Directory sin necesidad de utilizar una gateway de Internet, una instancia de conversión de las direcciones de red (NAT) o una conexión de VPN. Para obtener más información, consulte¿Qué es Amazon VPC?en laGuía del usuario de Amazon VPC.

Los puntos de conexión de la VPC de tipo interfaz utilizan la tecnología de AWS PrivateLink, una tecnología de AWS que permite la comunicación privada entre los servicios de AWS mediante una elastic network interface con direcciones IP privadas. Para obtener más información, consulteAWS PrivateLink para servicios de AWS.

Los siguientes pasos son para usuarios de Amazon VPC. Para obtener más información, consulteIntroducción a Amazon VPCen laGuía del usuario de Amazon VPC.

Availability

Cloud Directory actualmente admite puntos de conexión de la VPC en las regiones siguientes:

- US East (Ohio)
- US East (N. Virginia)
- US West (Oregon)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)

- AWS GovCloud (EE.UU. Oeste)

Creación de una VPC para Cloud Directory

Para comenzar a utilizar Cloud Directory con su VPC, utilice la consola de Amazon VPC para crear un punto de enlace de la VPC de tipo interfaz para Cloud Directory. Para obtener más información, consulte Creación de un punto de enlace de interfaz.

- ParaCategoría de servicio, elijaServicios de AWS.
- En Service Name (Nombre de servicio), seleccione **`com.amazonaws.region.clouddirectory`**, Esto crea un extremo de VPC para las operaciones de Cloud Directory.

Para obtener información general, consulte¿Qué es Amazon VPC?en laGuía del usuario de Amazon VPC.

Controlar el acceso a su terminal de VPC de Cloud Directory

Una política de punto de enlace de la VPC es una política de recursos de IAM que puede asociar a un punto de enlace cuando crea o modifica el punto de enlace. Si no asocia una política al crear un punto de enlace, se asociará automáticamente una política predeterminada que conceda acceso completo al servicio. Una política de punto de enlace no anula ni sustituye a las políticas de usuario de IAM ni las políticas específicas de los servicios. Se trata de una política independiente para controlar el acceso desde el punto de enlace al servicio especificado.

Las políticas de punto de conexión deben escribirse en formato JSON. Para obtener más información, consulte Controlar el acceso a servicios con puntos de enlace de la VPC en la Guía del usuario de Amazon VPC.

A continuación, se muestra un ejemplo de una política de punto de enlace para Cloud Directory. Esta política permite los usuarios que se conectan a Cloud Directory a través de la VPC enumerar directorios y les impide realizar otras acciones de Cloud Directory.

```
{
  "Statement": [
    {
      "Sid": "ReadOnly",
      "Principal": "*",
```

```
      "Action": [
        "clouddirectory:ListDirectories"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Para modificar la política de punto de enlace de la VPC para Cloud Directory

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Endpoints (Puntos de enlace).
3. Si todavía no ha creado el punto de enlace para Cloud Directory, elijaCreación de un punto de enlace. A continuación, seleccione**`com.amazonaws.region.clouddirectory`**y elijaCreación de un punto de enlace.
4. Seleccione la**`com.amazonaws.region.clouddirectory`**y elija la opciónPolítica deen la mitad inferior de la pantalla.
5. Elija Edit Policy (Editar política) y realice los cambios en la política.

Para obtener más información, consulte Controlar el acceso a servicios con puntos de enlace de la VPC en la Guía del usuario de Amazon VPC.

Descripción de los conceptos clave de Cloud Directory

Amazon Cloud Directory es un almacén de datos basado en directorios que puede crear diversos tipos de objetos según esquemas.

Temas

Schema

Un esquema es un conjunto de facetas que definen los objetos que se pueden crear en un directorio y cómo se organizan. El esquema también obliga a mantener la integridad de los datos y que puedan operar unos con otros. Un solo esquema se puede aplicar a más de un directorio a la vez. Para obtener más información, consulte Schemas.

Facets

Una faceta es un conjunto de atributos, restricciones y enlaces definidos dentro de un esquema. Combinadas, las facetas definen los objetos dentro de un directorio. Por ejemplo, Persona y Dispositivo pueden ser facetas para definir a los empleados de una empresa con la asociación de varios dispositivos. Para obtener más información, consulte Facets.

Esquema administrado

Un esquema proporcionado para facilitar el desarrollo y el mantenimiento rápido de sus aplicaciones. Para obtener más información, consulte Esquema administrado.

Esquemas de ejemplo

Conjunto de esquemas de ejemplo proporcionados de forma predeterminada en la consola de AWS Directory Service. Por ejemplo, Persona, Organización y Dispositivo son esquemas de ejemplo. Para obtener más información, consulte Esquemas de ejemplo.

Esquemas personalizados

Uno o varios esquemas definidos por un usuario que se pueden cargar en la sección Esquemas o durante el proceso de creación de Cloud Directory de la consola de AWS Directory Service. También pueden crearse mediante llamadas al API.

Directory

Un directorio es un almacén de datos basado en esquema que contiene determinados tipos de objetos organizados en una estructura multijerárquica (consulte Estructura de directorios para obtener más información). Por ejemplo, un directorio de usuarios puede proporcionar una vista jerárquica basada en estructuras de informes, en su ubicación y en la pertenencia a proyectos. De forma similar, un directorio de dispositivos puede tener varias vistas jerárquicas según su fabricante, su propietario actual y su ubicación física.

El directorio define el límite lógico del almacén de datos, aislándolo completamente del resto de directorios del servicio. También define los límites atribuibles a una solicitud individual. Las transacciones y las consultas se ejecutan en sí como una unidad en el contexto de un único directorio. Un directorio no se puede crear sin un esquema y normalmente tiene un esquema aplicado. No obstante, también puede servirse de las operaciones de API de Cloud Directory para aplicar esquemas adicionales a un directorio. Para obtener más información, consulte`ApplySchema`en laAmazon Cloud Directory API Reference Guide.

Objects

Los objetos son una entidad de datos estructurada dentro de un directorio. El cometido de un objeto dentro de un directorio es capturar metadatos (o atributos) acerca de una entidad física o lógica con el fin de, generalmente, detectar información y forzar la aplicación de políticas. Por ejemplo, los usuarios, dispositivos, aplicaciones, cuentas de AWS, instancias de EC2 y buckets de Amazon S3 se pueden representar como distintos tipos de objetos dentro de un directorio.

La información acerca del tipo de objeto y su estructura se expresa como un conjunto de facetas. Puede usar `Path` u `ObjectIdentifier` para obtener acceso a los objetos. Los objetos también pueden tener atributos, que son una unidad de metadatos que define el usuario. Por ejemplo, el objeto de usuario puede tener un atributo llamado email-address. Los atributos siempre se asocian a un objeto.

Policies

Las políticas son un tipo de objeto especializado útiles para almacenar permisos o capacidades. Las políticas ofrecen la acción de API `LookupPolicy`. La acción de búsqueda de políticas toma la referencia a cualquier objeto como información de inicio. A continuación, recorre el directorio hasta llegar a la raíz. La acción recopila todos los objetos de política que encuentra en cada ruta hasta la raíz. Cloud Directory no interpreta ninguna de estas políticas de forma alguna. Son los usuarios de Cloud Directory los que interpretan las políticas mediante su propia lógica de negocio especializada.

Por ejemplo, imagine un sistema que almacena información sobre empleados. Los empleados se agrupan según las funciones que desempeñan. Queremos establecer diferentes permisos para los miembros del grupo de Recursos Humanos y el grupo de Contabilidad. Los miembros del grupo de Recursos Humanos tendrán acceso a información de nóminas, mientras que los del grupo de Contabilidad tendrán acceso a información de contabilidad. Para establecer estos permisos, asociamos objetos de política a cada uno de estos grupos. Llegado el momento de evaluar los permisos de un usuario, podemos utilizar la acción de API `LookupPolicy` en el objeto de ese usuario. La`LookupPolicy`La acción de API recorre el árbol desde el objeto de política especificado hasta la raíz. Se detiene en cada nodo, comprueba si hay políticas asociadas y las devuelve.

Asociaciones de políticas

Las políticas se pueden asociar a otros objetos de dos formas: asociaciones normales principal-secundario y asociaciones de políticas especiales. Las asociaciones normales principal-secundario permiten asociar las políticas a un nodo principal. Esto suele ser útil como mecanismo sencillo para localizar las políticas dentro del directorio de datos. Las políticas no pueden tener elementos secundarios. Las políticas con asociación principal-secundario no se devolverán en las llamadas a la API `LookupPolicy`.

Los objetos de las políticas también se pueden asociar a otros objetos a través de asociaciones de políticas. Puede administrar estas asociaciones de políticas mediante las acciones de API `AttachPolicy` y `DetachPolicy`. Las asociaciones de políticas permiten localizar nodos de políticas cuando se usa la acción de API LookupPolicy.

Especificación de esquemas de políticas

Para comenzar a usar políticas, primero debe añadir una faceta al esquema que permita crear políticas. Para hacerlo, cree una faceta estableciendo el elemento `objectType` de la faceta en POLICY. Crear objetos mediante una faceta con el tipo POLICY garantiza que el objeto tenga capacidades de política.

Las facetas de políticas heredan dos atributos además de cualquier atributo que añada a la definición:

- policy_type (cadena, obligatorio): este es un identificador que puede proporcionar para distinguir entre distintos usos de políticas. Si las políticas corresponden de forma lógica a categorías claras, recomendamos configurar debidamente el atributo de tipo de política. La API `LookupPolicy` devuelve el tipo de política de las políticas adjuntas (consulte `PolicyAttachment`). Esto permite filtrar con facilidad el tipo de política específico que se busca. También permite utilizar policy_type para decidir cómo debe procesarse o interpretarse el documento.
- policy_document (binario, obligatorio): puede almacenar los datos específicos de la aplicación en este atributo, como concesiones de permisos asociadas a la política. Si lo prefiere, también puede almacenar datos relacionados con la aplicación en atributos normales en su faceta.

Información general acerca de las acciones de API de políticas

Para trabajar con políticas dispone de diversas acciones de API especializadas. Para ver la lista de las operaciones disponibles, consulte Amazon Cloud Directory Actions (Acciones de Amazon Cloud Directory).

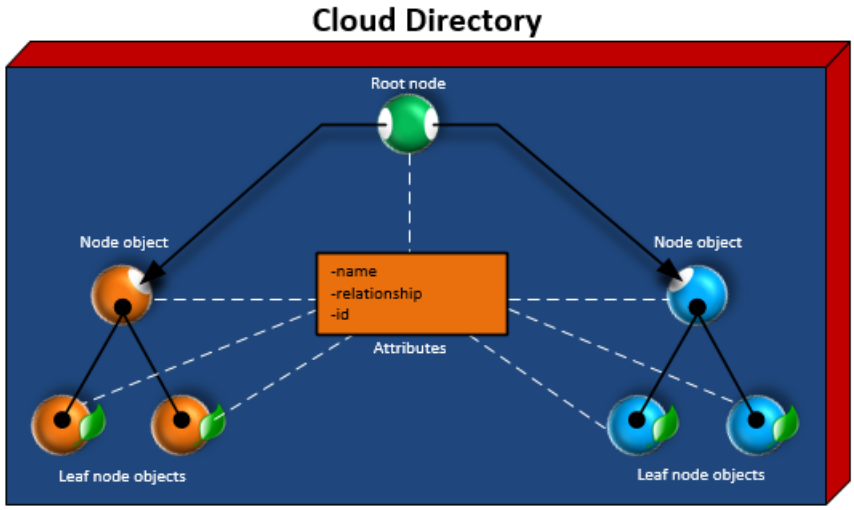
Para crear un objeto de política, utilice la acción de API `CreateObject` con una faceta adecuada:

- Para asociar o desasociar una política de un objeto, use las acciones `AttachPolicy` y `DetachPolicy`, respectivamente.
- Para buscar políticas que estén asociadas a objetos a lo largo del árbol, use la acción de API `LookupPolicy`.
- Para obtener una lista de las políticas asociadas a un objeto determinado, use la acción de API `ListObjectPolicies`.

Para ver la lista de operaciones y los permisos necesarios para realizar cada acción de API, consulte Permisos de la API de Amazon Cloud Directory: Referencia de acciones, recursos y condiciones.

Estructura de directorios

Los datos dentro de un directorio se estructuran jerárquicamente en un modelo de árbol que consta de nodos, nodos hijo y vínculos entre los nodos, como se muestra en la siguiente ilustración. Esto resulta útil en el desarrollo de aplicaciones para modelar, almacenar y recorrer rápidamente los datos jerárquicos.



Nodo raíz

La raíz es el nodo superior en un directorio y se usa para organizar los nodos principales y los secundarios de la jerarquía. Parecido a la organización de carpetas con subcarpetas y archivos en un sistema de archivos .

Node

Un nodo representa un objeto que puede tener objetos secundarios. Por ejemplo, un nodo puede representar de forma lógica un grupo de administradores donde diversos objetos de usuario son los elementos secundarios o nodos hijos. Un objeto nodo solo puede tener un elemento principal.

Nodo hijo

Un nodo hijo representa un objeto sin elementos secundarios que puede estar o no conectado directamente a un nodo principal. Por ejemplo, un objeto de usuario o de dispositivo. Un objeto nodo hijo puede tener varios elementos principales. Aunque no es necesario que los objetos nodo hijo estén conectados a un nodo principal, se recomienda encarecidamente que así sea, ya que sin una ruta desde la raíz, solo se puede acceder al objeto a través de su `NodeId`. Si se pierde el identificador de dicho objeto, no habrá manera de localizarlo de nuevo.

Enlace entre nodos

La conexión entre un nodo y otro. Cloud Directory admite varios tipos de enlace entre los nodos, incluidos enlaces principal-secundario, enlaces de política y enlaces de atributo de índice.

Schemas

Con Amazon Cloud Directory, los esquemas definen los tipos de objetos que se pueden crear en un directorio (usuarios, dispositivos y organizaciones), aplican la validación de datos para cada clase de objetos y gestionan los cambios en el esquema a lo largo del tiempo. En concreto, un esquema define lo siguiente:

- Uno o más tipos de facetas que podrían estar asignadas a objetos dentro de un directorio (como por ejemplo Person, Organization_Person)
- Atributos que podrían estar asignados a objetos dentro de un directorio (por ejemplo, Name, Description). Es posible que se requieran atributos o que sean opcionales en distintos tipos de facetas y se definen en el contexto de una faceta.
- Restricciones que podrían ser obligatorias en los atributos de objeto (como por ejemplo, Required, Integer, String)

Cuando un esquema se ha aplicado a un directorio, todos los datos dentro de dicho directorio deben adecuarse a dicho esquema aplicado. De esta forma, la definición de esquema es básicamente un proyecto que se puede utilizar para construir múltiples directorios con esquemas aplicados. Una vez creado, los esquemas aplicados pueden variar respecto al proyecto original, cada uno de distinta forma.

Los esquemas aplicados se pueden actualizar posteriormente con el control de versiones y volverse a aplicar a todos los directorios que los utilizan. Para obtener más información, consulte Actualización de esquema in situ.

Cloud Directory proporciona operaciones de API para crear, leer, actualizar y eliminar esquemas. Esto permite que los agentes de programación consuman con facilidad el contenido del esquema. Dichos agentes acceden al directorio para descubrir todo el conjunto de facetas, atributos y restricciones que se aplican a los datos en el directorio. Para obtener más información sobre las API de esquemas, consulte la Guía de referencia de API de Amazon Cloud Directory.

Cloud Directory permite cargar un archivo JSON compatible con la creación de esquemas. También puede crear y administrar esquemas utilizando la consola de AWS Directory Services. Para obtener más información, consulte Crear un directorio en Amazon Cloud Directory.

Temas

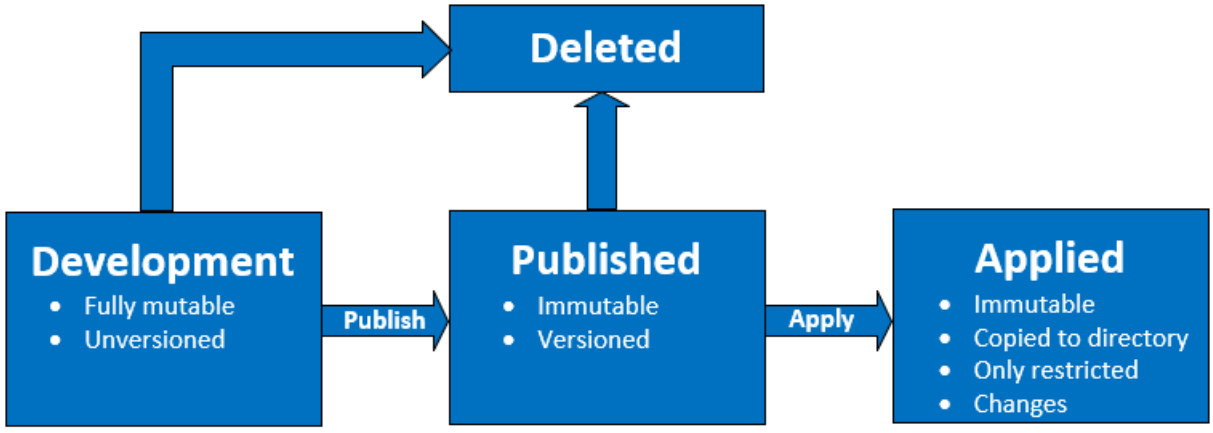
- Ciclo de vida de esquemas

- Facets
- Actualización de esquema in situ
- Esquema administrado
- Esquemas de ejemplo
- Esquemas personalizados
- Referencias de atributo
- Reglas de atributos
- Especificación de formato

Ciclo de vida de esquemas

Cloud Directory ofrece un ciclo de vida de esquemas para ayudar al desarrollo de esquemas. Este ciclo de vida consta de tres estados: Desarrollo, Publicado y Aplicado. Estos estados se han diseñado para facilitar la construcción y la distribución de los esquemas. Cada uno de estos estados tiene diferentes características que ayudan en esta tarea.

En el siguiente diagrama se muestran las posibles transiciones y texto. Todas las transiciones de esquema son de copia en escritura. Por ejemplo, la publicación de un esquema de desarrollo no altera ni elimina el esquema de desarrollo.



Puede eliminar un esquema cuando se encuentra en estado de desarrollo o publicado. La eliminación de un esquema no se puede deshacer ni se puede restaurar una vez que se ha eliminado.

Los esquemas en los estados de desarrollo, publicado y aplicado tienen ARN que los representan. Estas ARN se utilizan en operaciones de las API para describir el esquema en que opera la API. Es fácil discernir el estado de un esquema observando una ARN de esquema.

- Desarrollo: `arn:aws:clouddirectory:us-east-1:1234567890:schema/development/SchemaName`
- Publicado: `arn:aws:clouddirectory:us-east-1:1234567890:schema/published/SchemaName/Version`
- Aplicado: `arn:aws:clouddirectory:us-east-1:1234567890:directory/directoryid/schema/SchemaName/Version`

Estado de desarrollo

Los esquemas se crearon inicialmente en el estado de desarrollo. Los esquemas en este estado son totalmente mutables. Puede añadir o eliminar libremente facetas y atributos. La mayoría del diseño de esquema se produce en este estado. Los esquemas en este estado tienen nombre pero sin versión.

Estado Publicado

El estado de esquema publicado almacena esquemas que están listos para aplicarse a los directorios de datos. Los esquemas se publican desde el estado de desarrollo al estado publicado. Los esquemas no se pueden cambiar en el estado publicado. Puede aplicar esquemas publicados a cualquier número de directorios de datos.

Los esquemas publicados y aplicados deben tener una versión asociada a ellos. Para obtener más información acerca de las versiones, consulte Control de versiones de esquemas.

Estado Aplicado

Se puede aplicar un esquema publicado a los directorios de datos. Un esquema que se ha aplicado a un directorio de datos se dice que se ha aplicado. Una vez que se aplica un esquema a un directorio de datos, puede utilizar las facetas del esquema al crear objetos. Puede aplicar varios esquemas al mismo directorio de datos. Solo se permiten los siguientes cambios en un esquema aplicado.

- Añadir una faceta a un esquema aplicado
- Añadir un atributo no obligatorio a un esquema aplicado

Facets

Las facetas son la abstracción más básica dentro de un esquema. Representan un conjunto de atributos que se pueden asociar con un objeto en el directorio y son similares en concepto a las clases de objetos LDAP. Cada objeto de directorio puede tener hasta un determinado número de facetas asociadas. Para obtener más información, consulte Límites en Amazon Cloud Directory.

Cada faceta mantiene su propio conjunto de atributos independiente. Cada faceta consta de metadatos fundamentales, como el nombre de la faceta, información de la versión y comportamientos. La combinación de ARN de esquema, facetas y atributos definen la unicidad del objeto.

El conjunto de facetas de objetos, sus restricciones y las relaciones entre ellos constituyen una definición de esquema abstracta. Las facetas de esquema se utilizan para definir limitaciones a través de las siguientes cosas:

1. Atributos permitidos en un objeto
2. Tipos de políticas que se permiten aplicar a un objeto

Una vez que haya añadido las facetas necesarias a su esquema, puede aplicar el esquema al directorio y crear los objetos aplicables. Por ejemplo, puede definir un esquema de dispositivos añadiendo facetas como equipos, teléfonos y tablets. A continuación, puede utilizar estas facetas para crear objetos de equipo, objetos de teléfono y objetos de tablet en el directorio en el que se aplica el esquema.

La compatibilidad con el esquema de Cloud Directory facilita la tarea de añadir o modificar facetas y atributos sin necesidad de preocuparse de romper las aplicaciones. Para obtener más información, consulte Actualización de esquema in situ.

Actualización de esquema in situ

Cloud Directory ofrece la actualización de atributos de esquema y facetas existentes para ayudar a integrar sus aplicaciones con servicios proporcionados por AWS. Los esquemas que están en estado publicado o aplicado tienen versiones y no pueden modificarse. Para obtener más información, consulte Ciclo de vida de esquemas.

Control de versiones de esquemas

Una versión de esquema indica un identificador único para un esquema que los desarrolladores pueden especificar al programar sus aplicaciones para cumplir determinadas reglas y el formato de los datos. Es importante que los desarrolladores comprendan dos diferenciadores clave en el modo en el que funciona el control de versiones con Cloud Directory. Estos diferenciadores (la versión principal y la secundaria) pueden determinar cómo afectan las futuras actualizaciones del esquema a la aplicación.

Versión principal

La versión principal es el identificador de versión que se utiliza para realizar el seguimiento de los cambios de la versión principal de un esquema. Puede tener hasta 10 caracteres de longitud. Las distintas versiones del mismo esquema son totalmente independientes. Por ejemplo, dos esquemas con el mismo nombre y distintas versiones se tratan como esquemas completamente diferentes, con sus propios espacios de nombres.

Cambios no compatibles con versiones anteriores

Recomendamos realizar cambios en la versión principal solo cuando los esquemas no sean compatibles. Por ejemplo, al cambiar el tipo de datos de un atributo existente (como, por ejemplo, cambiar de `string` a `integer`) o suprimir un atributo obligatorio de su esquema. Los cambios no compatibles con versiones anteriores requieren la migración de los datos del directorio de una versión del esquema anterior a la nueva versión del esquema.

Versión secundaria

La versión secundaria es el identificador de versión utilizado para actualizar in situ esquemas o cuando desea realizar actualizaciones compatibles con versiones anteriores, como añadir atributos adicionales o añadir facetas. Se puede aplicar un esquema actualizado con una versión secundaria en todos los directorios que lo utilicen sin interrumpir ninguna aplicación en ejecución. Esto incluye los directorios que se utilizan en entornos de producción. Para ver un ejemplo de caso de uso, consulte«How to Easiy Apply Amazon Cloud Directory Schema Changes with In-Place Schema Upgrades»en el blog de Cloud Directory

La información de la versión secundaria y el historial se guardan con la demás información del esquema en el repositorio de metadatos del esquema. No se conserva en los objetos ninguna información de versión secundaria. La ventaja de introducir la versión secundaria es que el código de cliente funciona perfectamente siempre que no cambie la versión principal.

Límites de la versión secundaria

Cloud Directory conserva y, por lo tanto, limita hasta cinco versiones secundarias. Sin embargo, los límites de versión secundaria se aplican de forma diferente para los esquemas publicados y aplicados de las siguientes maneras:

- Esquemas aplicados: Una vez superado el límite de versión secundaria, Cloud Directory elimina automáticamente la versión secundaria más antigua.
- Esquemas publicados: Una vez superado el límite de versiones menores, Cloud Directory no elimina ninguna de las versiones secundarias, pero sí informa al usuario a través de un`LimitExceededException`que se ha superado el límite. Una vez que supere los límites de versiones secundarias, puede eliminar el esquema mediante la herramientaDeleteSchemaAPI o solicitar un aumento de límite.

Uso de operaciones API de actualización de esquema

Puede utilizar la llamada a la API `UpgradePublishedSchema` para actualizar esquemas publicados. Las actualizaciones de esquema se aplican in situ en los directorios que lo utilizan mediante la llamada a la API `UpgradeAppliedSchema`. También puede conseguir la versión principal y secundaria de un esquema aplicado llamando a `GetAppliedSchemaVersion` O bien puede ver los ARN de esquema asociados y el historial de revisiones del esquema de un directorio llamando a`ListAppliedSchemaArns`. Cloud Directory mantiene las cinco versiones más recientes de los cambios aplicados en el esquema.

Para obtener un ejemplo ilustrativo, consulte«How to Easiy Apply Amazon Cloud Directory Schema Changes with In-Place Schema Upgrades»en el blog de Cloud Directory La publicación del blog demostrará cómo se realiza una actualización de esquema in situ y se utilizan las versiones de esquema en Cloud Directory. Trata acerca de cómo añadir atributos adicionales a una faceta existente, añadir una nueva faceta a un esquema, publicar el nuevo esquema y aplicarlo a directorios en ejecución para completar la actualización de un esquema in situ. También muestra cómo ver el historial de la versión de un esquema de directorios, lo que contribuye a garantizar que la flota de directorios está ejecutando la misma versión del esquema y que tiene aplicado el historial de cambios de esquema correcto.

Esquema administrado

Cloud Directory facilita el desarrollo rápido de aplicaciones mediante la utilización de un esquema administrado. Con un esquema administrado, puede crear un directorio y comenzar a crear y recuperar objetos de él a un ritmo más rápido. Para obtener más información, consulte Creación de su directorio.

En la actualidad, hay un esquema administrado, denominado `QuickStartSchema`. Puede crear un modelo de datos jerárquicos completo y establecer relaciones entre objetos mediante construcciones como, por ejemplo, Enlaces con tipo. A continuación, puede consultar cualquier información en los datos atravesando la jerarquía.

El esquema administrado `QuickStartSchema` está representada por el JSON siguiente:

```
QuickStartSchema: {
    "facets": {
        "DynamicObjectFacet": {
            "facetStyle": "DYNAMIC"
        },
        "DynamicTypedLinkFacet": {
            "facetAttributes": {
                "DynamicTypedLinkAttribute": {
                    "attributeDefinition": {
                        "attributeRules": {},
                        "attributeType": "VARIANT",
                        "isImmutable": false
                    },
                    "requiredBehavior": "REQUIRED_ALWAYS"
                }
            },
            "identityAttributeOrder": [
                "DynamicAttribute"
            ]
        }
    }
}
```

ARN de QuickStartSchema

El esquema administrado `QuickStartSchema` utiliza el ARN siguiente:

```
String QUICK_START_SCHEMA_ARN = "arn:aws:clouddirectory:::schema/managed/
quick_start/1.0/001" ;
```

Por ejemplo, puede utilizar este ARN para crear un directorio denominado `ExampleDirectory`, como se muestra a continuación:

```
CreateDirectoryRequest createDirectoryRequest = new CreateDirectoryRequest()
    .withName("ExampleDirectory") // Directory name
    .withSchemaArn(QUICK_START_SCHEMA_ARN);
```

Estilos de faceta

Existen dos estilos diferentes que se pueden definir en cualquier faceta específica: `Static` y `Dynamic`.

Facetas estáticas

Las facetas estáticas son la mejor opción cuando tiene todos los detalles de su modelo de datos para su directorio, como, por ejemplo, una lista de atributos con sus tipos de datos y también desea definir restricciones para sus atributos como, por ejemplo, los campos obligatorios o únicos. Cloud Directory aplicará las restricciones de datos y la comprobación de reglas durante la creación o el cambio de objetos.

Facetas dinámicas

Puede utilizar una faceta dinámica cuando necesita flexibilidad para cambiar el número de atributos o para cambiar los valores de datos que se almacenan en sus atributos. Cloud Directory no aplica restricciones de datos ni comprobación de reglas durante la creación o el cambio de objetos.

Después de crear un esquema con facetas dinámicas, puede definir todos los atributos que necesite al crear objetos. El Cloud Directory aceptará los atributos como pares clave-valor y los almacenará en los objetos proporcionados.

Puede añadir una faceta dinámica a un esquema nuevo o existente. También puede combinar facetas estáticas y dinámicas en un único esquema para beneficiarse de cada estilo de faceta de su directorio.

Cuando se crea algún atributo con una faceta dinámica, se crean como tipos de datos `Variant`. Para almacenar valores para el atributo definido como`Variant`Puede utilizar valores de

cualquiera de los tipos de datos primitivos que se admiten en Cloud Directory, como, por ejemplo,`String`or`Binary`. Con el tiempo, también puede cambiar el valor del atributo por otro tipo de datos. No se aplica la validación de datos.

Puede utilizar facetas dinámicas para definir los objetos del siguiente tipo:

- `NODE`
- `LEAF_NODE`
- `POLICY`

Para obtener más información acerca de esquemas administrados, facetas dinámicas o tipos de datos variante y ver ejemplos de casos de uso, consulte[How to rapiddevelop applications on Amazon Cloud Directory with AWS Managed Schema](#)en el blog de Amazon Cloud Directory.

Esquemas de ejemplo

Cloud Directory dispone de ejemplos de esquemas para Organizaciones, Personas y Dispositivos. En la siguiente sección se enumeran los distintos esquemas de ejemplo y se muestran las diferencias para cada una de ellas.

Organizations

En las tablas siguientes se muestran las facetas que se incluyen en el esquema de ejemplo Organizations.

Faceta "Organization"	Tipo de datos	Length	¿Comp miento obligat io?	Descripción
id_cuenta	Cadena	1024	N	Identificador único para Organización
account_name	Cadena	1024	N	Nombre de la organización
organization_status	Cadena	1024	N	Estado como, por ejemplo, "active", "suspended", "inactive", "closed"

Faceta "Organization"	Tipo de datos	Length	¿Comp miento obligat io?	Descripción
mailing_address (street1)	Cadena	1024	N	Una dirección postal física para esta compañía/entidad
mailing_address (street2)	Cadena	1024	N	Una dirección postal física para esta compañía/entidad
mailing_address (city)	Cadena	1024	N	Una dirección postal física para esta compañía/entidad
mailing_address (state)	Cadena	1024	N	Una dirección postal física para esta compañía/entidad
mailing_address (country)	Cadena	1024	N	Una dirección postal física para esta compañía/entidad
mailing_address (postal_code)	Cadena	1024	N	Una dirección postal física para esta compañía/entidad
email	Cadena	1024	N	El identificador de correo electrónico para la organización
web_site	Cadena	1024	N	URL de sitio web
telephone_number	Cadena	1024	N	Número de teléfono para organización
Descripción	Cadena	1024	N	Descripción de organización

Faceta "Legal_Entity"	Tipo de datos	Length	¿Comp miento obligat io?	Descripción
registered_company _name	Cadena	1024	N	Nombre de la entidad legal
mailing_address (street1)	Cadena	1024	N	Un domicilio social físico para esta compañía/entidad
mailing_address (street2)	Cadena	1024	N	Un domicilio social físico para esta compañía/entidad
mailing_address (city)	Cadena	1024	N	Un domicilio social físico para esta compañía/entidad
mailing_address (state)	Cadena	1024	N	Un domicilio social físico para esta compañía/entidad
mailing_address (country)	Cadena	1024	N	Un domicilio social físico para esta compañía/entidad
mailing_address (postal_code)	Cadena	1024	N	Un domicilio social físico para esta compañía/entidad
industry_vertical	Cadena	1024	N	Segmento industrial
billing_currency	Cadena	1024	N	Divisa de facturación
tax_id	Cadena	1024	N	Número de identificación fiscal

Person

En las tablas siguientes se muestran las facetas que se incluyen en el esquema de ejemplo Person.

Faceta "Person"	Tipo de datos	Length	¿Comportamiento obligatorio?	Descripción
display_name	Cadena	1024	N	El nombre del usuario, adecuado para mostrar a los usuarios finales.
first_name	Cadena	1024	N	El nombre del usuario o nombre en la mayoría de idiomas occidentales
last_name	Cadena	1024	N	El apellido del usuario en la mayoría de idiomas occidentales
middle_name	Cadena	1024	N	El segundo nombre del usuario
nickname	Cadena	1024	N	La forma coloquial de dirigirse al usuario en la vida real, por ejemplo, "Bob" o "Bobby" en lugar de "Robert".
email	Cadena	1024	N	Dirección de correo electrónico del usuario
mobile_phone_number	Cadena	1024	N	Número de teléfono del usuario
home_phone_number	Cadena	1024	N	Número de teléfono del usuario
nombre de usuario	Cadena	1024	S	identificador único para el usuario

Faceta "Person"	Tipo de datos	Length	¿Comportamiento obligatorio?	Descripción
profile	Cadena	1024	N	Un URI que es un localizador uniforme de recursos y que apunta a una ubicación que representa el perfil en línea del usuario (como una página web)
picture	Cadena	1024	N	Una URI que es un localizador uniforme de recursos que apunta a una ubicación de recursos que representa la imagen del usuario.
sitio web	Cadena	1024	N	URL
timezone	Cadena	1024	N	La zona horaria del usuario
locale	Cadena	1024	N	Se utiliza para indicar la ubicación predeterminada del usuario para fines de localización tales como divisa, formato de fecha y hora o representaciones numéricas.
dirección (street1)	Cadena	1024	N	Una dirección postal física para este usuario.
address (street2)	Cadena	1024	N	Una dirección postal física para este usuario.
address (city)	Cadena	1024	N	Una dirección postal física para este usuario.

Faceta "Person"	Tipo de datos	Length	¿Comportamiento obligatorio?	Descripción
address (state)	Cadena	1024	N	Una dirección postal física para este usuario.
address (country)	Cadena	1024	N	Una dirección postal física para este usuario.
address (postal_code)	Cadena	1024	N	Una dirección postal física para este usuario.
user_status	Cadena	1024	N	Valor que indica el estado administrativo del usuario

Faceta "Organization_Person"	Tipo de datos	Length	¿Comportamiento obligatorio?	Descripción
title	Cadena	1024	N	Tratamiento en la organización
preferred_language	Cadena	1024	N	Indica los idiomas escritos o hablados que prefiere el usuario y generalmente se utilizan para seleccionar una interfaz de usuario localizada.
employee_id	Cadena	1024	N	Un identificador de cadena, normalmente numérica o alfanumérica, asignado a una persona

Faceta "Organization_Person"	Tipo de datos	Length	¿Comportamiento obligatorio?	Descripción
cost_center	Entero	1024	N	Identifica el centro de costos
department	Cadena	1024	N	Identifica el nombre de un departamento
manager	Cadena	1024	N	El jefe del usuario
company_name	Cadena	1024	N	Identifica el nombre de una organización
company_address (street1)	Cadena	1024	N	Una dirección postal física para la organización
company_address (street2)	Cadena	1024	N	Una dirección postal física para la organización
company_address (city)	Cadena	1024	N	Una dirección postal física para la organización
company_address (state)	Cadena	1024	N	Una dirección postal física para la organización
company_address (country)	Cadena	1024	N	Una dirección postal física para la organización
company_address (postalCode)	Cadena	1024	N	Una dirección postal física para la organización

Device

En la siguiente tabla se muestran las facetas que se incluyen en el esquema de ejemplo Device.

Faceta "Device"	Tipo de datos	Length	¿Comportamiento obligatorio?	Descripción
device_id	Cadena	1024	N	Identificador alfanumérico único del dispositivo
name (nombre)	Cadena	1024	N	Nombre sencillo del dispositivo
Descripción	Cadena	1024	N	Descripción del dispositivo
X.509_certificates	Cadena	1024	N	Certificado X.509
device_version	Cadena	1024	N	Versión del dispositivo
device_os_type	Cadena	1024	N	Sistema operativo en el dispositivo
device_os_version	Cadena	1024	N	Número de versión del sistema operativo en el dispositivo
serial_number	Cadena	1024	N	Número de serie del dispositivo
device_status	Cadena	1024	N	Estado del dispositivo (como active, not_active, suspended, shutdown, off)

Esquemas personalizados

El primer paso en la creación de un esquema personalizado consiste en definir exactamente los campos que se deben indexar. Estos campos obligatorios definen los elementos del esqueleto del esquema, a los que añadir sus propios campos. Asigne el nombre y el tipo de cada campo (como, por ejemplo, cadena, entero, booleano) a la estructura de su objeto. Puede definir un esquema

con tipos y restricciones y, a continuación, aplicarlos a un directorio. Once defined, Cloud Directory performs validation for attributes.

Para obtener más información, consulte Crear un esquema.

Referencias de atributo

Las facetas de Amazon Cloud Directory contienen atributos. Los atributos pueden ser una definición de atributo o una referencia de atributo. Las definiciones de atributo son atributos que declaran su nombre y tipo primitivo (string, binary, Boolean, DateTime o number). Opcionalmente, también pueden declarar el comportamiento requerido, el valor predeterminado, el indicador inmutable y reglas de atributo (por ejemplo, longitud mínima o máxima).

Las referencias de atributo son atributos que obtienen su tipo primitivo, el valor predeterminado, el indicador inmutable y las reglas de atributo de otra definición de atributo previamente existente. No tienen su propio tipo primitivo, ni valores predeterminados, ni indicador inmutable ni reglas, ya que dichas propiedades provienen de la definición de atributo de destino.

Las referencias de atributos pueden anular el comportamiento requerido de una definición de destino (más detalles acerca de esto a continuación).

Al crear una referencia de atributo, proporciona solo un nombre de atributo y la definición de atributo de destino (que incluye el nombre de la faceta y el nombre de atributo de la definición de atributo de destino). Las referencias de atributo pueden no hacer referencia a otras referencias de atributo. Además, en este momento, es posible que las referencias de atributo no se dirijan a definiciones de atributo de otro esquema.

Puede usar una referencia de atributo si desea que dos o más atributos de un objeto hagan referencia a la misma ubicación de almacenamiento. Por ejemplo, imagine un objeto que tenga aplicadas una faceta User y otra EnterpriseUser. La faceta User tiene una definición de atributo FirstName, mientras que EnterpriseUser tiene una referencia de atributo que apunta a User.FirstName. Como ambos atributos FirstName hacen referencia a la misma ubicación de almacenamiento del objeto, cualquier cambio que se produzca en User.FirstName o en EnterpriseUser.FirstName tendrá el mismo efecto.

Ejemplo de API

En el ejemplo siguiente se muestra el uso de las referencias de atributo mediante la API de Cloud Directory. En este ejemplo, una faceta base contiene una definición de atributo y otra faceta contiene un atributo que hace referencia a un atributo de la faceta base. Tenga en cuenta que el atributo de

referencia puede marcarse como Required (obligatorio), aunque la faceta base lo esté como Not Required (no obligatoria).

```
   // create base facet
   CreateFacetRequest req1 = new CreateFacetRequest()
     .withSchemaArn(devSchemaArn)
     .withName("baseFacet")
     .withAttributes(List(
       new FacetAttribute()
         .withName("baseAttr")
         .withRequiredBehavior(RequiredAttributeBehavior.NOT_REQUIRED)
         .withAttributeDefinition(new
FacetAttributeDefinition().withType(FacetAttributeType.STRING))))
     .withObjectType(ObjectType.DIRECTORY)
   cloudDirectoryClient.createFacet(req1)

   // create another facet that refers to the base facet
   CreateFacetRequest req2 = new CreateFacetRequest()
     .withSchemaArn(devSchemaArn)
     .withName("facetA")
     .withAttributes(List(
       new FacetAttribute()
         .withName("ref")
         .withRequiredBehavior(RequiredAttributeBehavior.REQUIRED_ALWAYS)
         .withAttributeReference(new FacetAttributeReference()
           .withTargetFacetName("baseFacet")
           .withTargetAttributeName("baseAttr"))))
     .withObjectType(ObjectType.DIRECTORY)
   cloudDirectoryClient.createFacet(req2)
```

Ejemplo de JSON:

En el ejemplo siguiente se muestra el uso de referencias de atributo en un modelo JSON. El esquema representado por este modelo es idéntico al modelo anterior.

```
{
 "facets" : {
   "baseFacet" : {
     "facetAttributes" : {
       "baseAttr" : {
         "attributeDefinition" : {
           "attributeType" : "STRING"
```

```
          },
          "requiredBehavior" : "NOT_REQUIRED"
        }
      },
      "objectType" : "DIRECTORY"
    },
    "facetA" : {
      "facetAttributes" : {
        "ref" : {
          "attributeReference" : {
            "targetFacetName" : "baseFacet",
            "targetAttributeName" : "baseAttr"
          },
          "requiredBehavior" : "REQUIRED_ALWAYS"
        }
      },
      " objectType" : "DIRECTORY"
    }
}
```

Cuestiones acerca de las referencias de atributo

Las referencias de atributo deben dirigirse a una definición de atributo que ya exista en el mismo esquema.

- Las referencias de atributo pueden dirigirse a una definición de atributo que ya exista en la misma faceta o en otra distinta.
- Las referencias de atributo pueden no dirigirse a otras referencias de atributo.
- Las facetas que contienen definiciones de atributo que son el destino de la referencia de atributo de otra faceta no se pueden eliminar hasta que se hayan eliminado todas las referencias.

Puede usar las referencias de atributo de la misma forma en que usa las definiciones de atributo tradicionales, mediante la creación de objetos o la aplicación de facetas a objetos existentes.

> **Note**
>
> Puede aplicar facetas con referencias a otras facetas, pero no es necesario aplicar las facetas de destino directamente. Si no se aplica la faceta de destino, no se producirá ningún

cambio en el comportamiento de la referencia de atributo (debe aplicar facetas de destino solo si desea que los demás atributos de esa faceta existan en el objeto).

Establecimiento de los valores de referencia de atributo

Puede llamar a la acción de API UpdateObjectAttributes si desea cambiar el valor de un atributo. La actualización (o eliminación) de la definición o de cualquier otra referencia a esa misma definición de ese objeto tiene el mismo efecto.

Obtención de los valores de referencia de atributo

Puede llamar a la acción de API ListObjectAttributes para recuperar alias de almacenamiento. Esta llamada devuelve una lista de tuplas. Cada una de ellas contiene una clave de atributo y un valor asociado. Las claves de atributo corresponden a la lista de alias de almacenamiento presentes en ese objeto.

> **Note**
>
> Es posible que se devuelva una clave de atributo para una faceta que no se ha aplicado de forma explícita a un objeto. Esto puede ocurrir cuando las referencias de atributo se dirigen a facetas que no se aplican al objeto.

Por ejemplo, imagine que tiene una faceta User y otra EnterpriseUser. El atributo EnterpriseUser.FirstName hace referencia a User.FirstName. Luego se aplican tanto la faceta User como EnterpriseUser a un objeto, se establece User.FirstName en Robert y, posteriormente, se establece EnterpriseUser.FirstName en Bob. Cuando llama a ListObjectAttributes, solo ve "User.FirstName = Bob", ya que solo hay un alias de almacenamiento para ambos atributos FirstName.

Uso de índices con las referencias de atributo

Solo puede crear índices con una definición de atributo, no con una referencia. La lista de un índice no devuelve claves de atributo para las referencias de atributo. No obstante, devuelve claves de atributo para cualquier definición de atributo a la que se dirijan las referencias que existan en el objeto indexado. En otras palabras, en la capa de índice, las referencias de atributo se tratan simplemente como el identificador alternativo de un atributo, que se resuelve como el identificador de definición de atributo correcto en el tiempo de ejecución.

Por ejemplo, imagine que tiene un índice para el atributo FirstName de la faceta User. Usted asocia un objeto teniendo aplicada únicamente la faceta EnterpriseUser. A continuación, establece el valor para el atributo EnterpriseUser.FirstName de ese objeto en Bob. Por último, llama a la acción ListIndex. Los resultados contienen únicamente "User.FirstName = Bob".

Comportamiento requerido para las referencias de atributo

Una referencia de atributo puede tener un comportamiento requerido que sea distinto de su definición de atributo de destino. Esto permite que la definición de base sea opcional, al mismo tiempo que puede requerirse una referencia a esa misma definición. Cuando un objeto tiene una definición de base y una o varias referencias a la misma definición de base, la definición de base y todas las referencias deben adherirse al comportamiento requerido en todos los atributos relacionados.

- Al igual que con las definiciones de atributo, debe proporcionar los valores de las definiciones de atributo requeridas al crear el objeto o al añadir una faceta a un objeto existente.
- Por comodidad, si más de un atributo de un objeto hace referencia a la misma ubicación de almacenamiento, solo debe proporcionar un valor para uno de los atributos de esa ubicación de almacenamiento.
- De la misma forma, si proporciona varios valores para la misma ubicación de almacenamiento, estos deben ser iguales.

Reglas de atributos

Las reglas describen los valores permitidos de un tipo de atributo y limitan los valores permitidos para cualquier atributo determinado. Debe especificar reglas como parte de una definición de atributo al crear una faceta. Cloud Directory es compatible con los siguientes tipos de regla:

- Longitud de cadena
- Longitud binario
- Cadena de conjunto
- Comparación de números

Longitud de cadena

Limita la longitud de un valor de atributo de cadena.

Claves de parámetro de regla permitidos: mín, máx

Valores de parámetro de regla permitidos: número

Longitud binario

Limita la longitud de matriz de bytes de un valor de atributo binario.

Claves de parámetro de regla permitidos: mín, máx

Valores de parámetro de regla permitidos: número

Cadena de conjunto

Limita el valor de un atributo de cadena al conjunto permitido de cadenas especificadas.

Claves de parámetros de regla permitidos: allowedValues

Valores de parámetro de regla permitidos: Conjunto de cadenas con cada cadena que cifrar con UTF-8

Los valores permitidos están delimitados por comas y pueden ir entre comillas. Esto resulta útil cuando los valores permitidos incluyen comas. Por ejemplo:

- Uno, dos, tres = corresponde a uno dos o tres
- "con,coma","sin coma" = corresponde a "con,coma" o "sincoma"
- con”comillas,sin comillas corresponde a ‘con”comillas’ o ‘sincomillas’

Comparación de números

Limita el valor numérico permitido para un atributo de número.

Claves de parámetro de regla permitidos: mín, máx

Valores de parámetro de regla permitidos: número

Especificación de formato

Un esquema de Cloud Directory añade estructura a los datos en los directorios de datos. Cloud Directory ofrece dos mecanismos para que pueda definir su esquema. Los desarrolladores pueden especificar operaciones de API específicas para construir un esquema o pueden cargar un esquema completamente con las capacidades de carga de esquemas. Los documentos de esquemas se pueden cargar a través de llamadas de API o a través de la consola. En esta sección se describe el formato que utilizar cuando se cargan todos los documentos de esquema.

Formato de esquemas JSON

Un documento de esquema es un documento JSON con el siguiente formato general.

```
{
    "facets": {
        "facet name": {
            "facetAttributes": {
                "attribute name": Attribute JSON Subsection
            }
        }
    }
}
```

Un documento de esquema contiene una asignación de nombres de faceta para facetas. Cada faceta, a su vez, contiene una asignación que contiene atributos. Todos los nombres de facetas dentro de un esquema deben ser únicos. Todos los nombres de atributo en una faceta deben ser únicos.

Subsección JSON de atributo

Las facetas contienen atributos. Cada atributo define el tipo de valor que se puede almacenar en un atributo. El siguiente formato JSON describe un atributo.

```
{
    "attributeDefinition": Attribute Definition Subsection,
    "attributeReference": Attribute Reference Subsection,
    "requiredBehavior": "REQUIRED_ALWAYS" or "NOT_REQUIRED"
}
```

Debe proporcionar una definición de atributo o una referencia de atributo. Consulte las subsecciones relacionadas para obtener más información sobre cada una de ellas.

El campo de comportamiento obligatorio indica si este atributo es obligatorio o no. Debe proporcionar este campo. Los valores posibles son los siguientes:

- `REQUIRED_ALWAYS`: este atributo se debe proporcionar cuando se crea el objeto o cuando se añade una faceta al objeto. No puede eliminar este atributo.
- `NOT_REQUIRED`: este atributo puede estar o no presente.

Subsección de definición de atributos

Un atributo define el tipo y las reglas asociadas a un valor de atributo. El siguiente diseño JSON describe el formato.

```
{
    "attributeType": One of "STRING", "NUMBER", "BINARY", "BOOLEAN" or "DATETIME",
    "defaultValue": Default Value Subsection,
    "isImmutable": true or false,
    "attributeRules": "Attribute Rules Subsection"
}
```

Subsección Valor predeterminado

Especifique exactamente uno de los siguientes valores predeterminados. Los valores de tipo largo y booleanos se deben suministrar fuera de las comillas (como sus tipos de Javascript respectivos en lugar de cadenas). Los valores binarios se proporcionan mediante una cadena de cifrado Base64 URL-segura (como se describe en RFC 4648). Los valores de fecha y hora se proporcionan en el número de milisegundos desde la fecha de inicio (00:00:00 UTC el 1 de enero de 1970).

```
{
 "stringValue": "a string value",
 "longValue": an integer value,
 "booleanValue": true or false,
 "binaryValue": a URL-safe Base64 encoded string,
 "datetimeValue": an integer value representing milliseconds since epoch
}
```

Subsección Reglas de atributo

Las reglas de atributo definen limitaciones sobre los valores de atributo. Puede definir varias reglas para cada atributo. Las reglas de atributo contienen un tipo de regla y un conjunto de parámetros para la regla. Puede encontrar más información en la sección Reglas de atributos.

```
{
    "rule name": {
        "parameters": {
            "rule parameter key 1": "value",
            "rule parameter key 2": "value"
        },
        "ruleType": "rule type value"
```

```
    }
}
```

Subsección Referencia de atributo

Las referencias de atributo son una característica avanzada. Permiten que varias facetas compartan una definición de atributo y un valor guardado. Consulte la sección Referencias de atributo para obtener más información. Puede definir una referencia de atributo en el esquema JSON con la siguiente plantilla.

```
{
 "targetSchemaArn": "schema ARN"
 "targetFacetName": "facet name"
 "targetAttributeName": "attribute name"
}
```

Ejemplos de documentos de esquemas

A continuación, se indican ejemplos de documentos de esquema que muestran el formato JSON válido.

> **Note**
>
> Todos los valores expresados en la cadena `allowedValues` deben estar separados por comas y sin espacios. Por ejemplo, `"SENSITIVE,CONFIDENTIAL,PUBLIC"`.

Documento de esquema básico

```
{
    "facets": {
        "Employee": {
            "facetAttributes": {
                "Name": {
                    "attributeDefinition": {
                        "attributeType": "STRING",
                        "isImmutable": false,
                        "attributeRules": {
                            "NameLengthRule": {
                                "parameters": {
                                    "min": "3",
```

```
                                "max": "100"
                            },
                            "ruleType": "STRING_LENGTH"
                        }
                    }
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
            "EmailAddress": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {
                        "EmailAddressLengthRule": {
                            "parameters": {
                                "min": "3",
                                "max": "100"
                            },
                            "ruleType": "STRING_LENGTH"
                        }
                    }
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
            "Status": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": false,
                    "attributeRules": {
                        "rule1": {
                            "parameters": {
                                "allowedValues": "ACTIVE,INACTIVE,TERMINATED"
                            },
                            "ruleType": "STRING_FROM_SET"
                        }
                    }
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            }
        },
        "objectType": "LEAF_NODE"
    },
    "DataAccessPolicy": {
        "facetAttributes": {
```

```
                "AccessLevel": {
                    "attributeDefinition": {
                        "attributeType": "STRING",
                        "isImmutable": true,
                        "attributeRules": {
                            "rule1": {
                                "parameters": {
                                    "allowedValues": "SENSITIVE,CONFIDENTIAL,PUBLIC"
                                },
                                "ruleType": "STRING_FROM_SET"
                            }
                        }
                    },
                    "requiredBehavior": "REQUIRED_ALWAYS"
                }
            },
            "objectType": "POLICY"
        },
        "Group": {
            "facetAttributes": {
                "Name": {
                    "attributeDefinition": {
                        "attributeType": "STRING",
                        "isImmutable": true
                    },
                    "requiredBehavior": "REQUIRED_ALWAYS"
                }
            },
            "objectType": "NODE"
        }
    }
}
```

Documento de esquema de enlaces con tipo

```
{
    "sourceSchemaArn": "",
    "facets": {
        "employee_facet": {
            "facetAttributes": {
                "employee_login": {
                    "attributeDefinition": {
                        "attributeType": "STRING",
```

```
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
            "employee_id": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
            "employee_name": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
            "employee_role": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            }
        },
        "objectType": "LEAF_NODE"
    },
    "device_facet": {
        "facetAttributes": {
            "device_id": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
            "device_type": {
                "attributeDefinition": {
```

```
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            }
        },
        "objectType": "NODE"
    },
    "region_facet": {
        "facetAttributes": {},
        "objectType": "NODE"
    },
    "group_facet": {
        "facetAttributes": {
            "group_type": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            }
        },
        "objectType": "NODE"
    },
    "office_facet": {
        "facetAttributes": {
            "office_id": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
            "office_type": {
                "attributeDefinition": {
                    "attributeType": "STRING",
                    "isImmutable": true,
                    "attributeRules": {}
                },
                "requiredBehavior": "REQUIRED_ALWAYS"
            },
```

```
                "office_location": {
                    "attributeDefinition": {
                        "attributeType": "STRING",
                        "isImmutable": true,
                        "attributeRules": {}
                    },
                    "requiredBehavior": "REQUIRED_ALWAYS"
                }
            },
            "objectType": "NODE"
        }
    },
    "typedLinkFacets": {
        "device_association": {
            "facetAttributes": {
                "device_type": {
                    "attributeDefinition": {
                        "attributeType": "STRING",
                        "isImmutable": false,
                        "attributeRules": {}
                    },
                    "requiredBehavior": "REQUIRED_ALWAYS"
                },
                "device_label": {
                    "attributeDefinition": {
                        "attributeType": "STRING",
                        "isImmutable": false,
                        "attributeRules": {}
                    },
                    "requiredBehavior": "REQUIRED_ALWAYS"
                }
            },
            "identityAttributeOrder": [
                "device_label",
                "device_type"
            ]
        }
    }
}
```

Objetos de directorio

Los objetos de directorio de modelo de desarrolladores que utilizan esquemas ampliables para aplicar automáticamente las restricciones de corrección de datos, facilitando la programación. Amazon Cloud Directory ofrece búsqueda de información completa en función de los atributos indexados definidos, permitiendo de este modo recorridos rápidos y búsquedas dentro de los árboles de directorio. Las datos de Cloud Directory están cifrados en reposo y en tránsito.

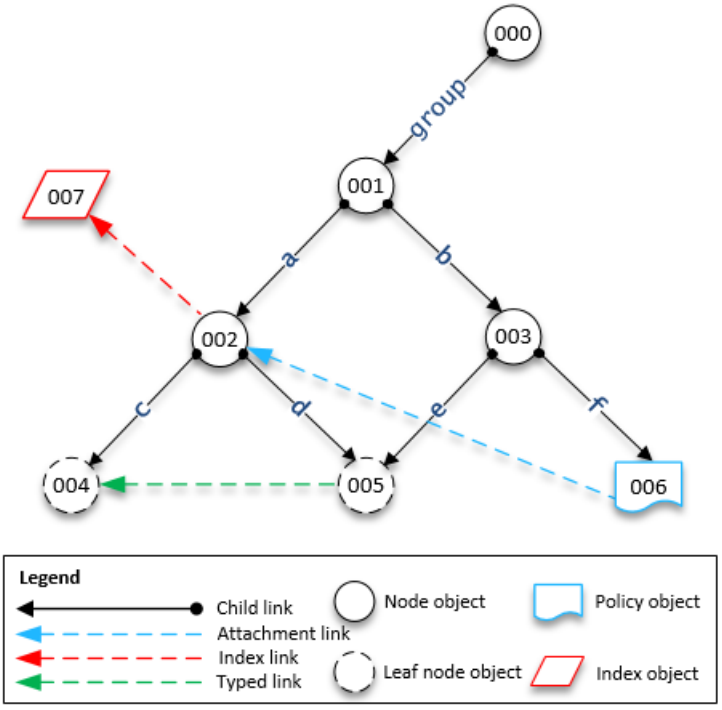
Un objeto es un elemento básico de Cloud Directory. Cada objeto tiene un identificador único globalmente, que especifica el identificador de objetos. Un objeto es una recopilación de cero o más facetas con sus claves de atributos y valores. Un objeto se puede crear a partir de una o varias facetas dentro de un único esquema aplicado o a partir de facetas de varios esquemas aplicados. Durante la creación de objetos, debe especificar todos los valores de atributo necesarios. Los objetos pueden tener un número limitado de facetas. Para obtener más información, consulte Límites en Amazon Cloud Directory.

Un objeto puede ser un objeto normal, un objeto de política o un objeto de índice. Un objeto también puede ser un objeto de nodo o un objeto de nodo hijo. El tipo del objeto se infiere a partir del tipo de objeto de las facetas que tiene adjuntas.

Temas

Links

Un enlace es un borde dirigido entre dos objetos que define una relación. Cloud Directory admite actualmente los siguientes tipos de enlace.



Enlaces secundarios

Un enlace secundario crea una relación principal-secundario entre los objetos que conecta. Por ejemplo, en la ilustración anterior, el enlace secundario b conecta los objetos `001` y `003`. Los enlaces secundarios definen la jerarquía en Cloud Directory. Los enlaces secundarios tienen nombres cuando participan en la definición de la ruta del objeto al que apunta el enlace.

Enlaces de adjunto

Un enlace de adjunto aplica un objeto de política de nodo de hoja a otro nodo de hoja o a un objeto de nodo. Los enlaces de adjunto no definen la estructura jerárquica de Cloud Directory. Por ejemplo, en la ilustración anterior, el enlace de adjunto aplica la política almacenada en el objeto de nodo de hoja de política `006` en el objeto de nodo `002`. Cada objeto puede tener varias políticas adjuntas, pero se puede adjuntar más de una política de cualquier tipo de política dado.

Enlaces de índice

Los enlaces de índice permiten buscar información completa en función de un objeto de índice y los atributos indexados definidos, permitiendo de este modo recorridos rápidos y búsquedas dentro de los árboles de directorio. Conceptualmente, los índices son similares a los nodos con hijos: Los enlaces a los nodos indexados se etiquetan de acuerdo con los atributos indexados, en lugar de recibir una etiqueta cuando se adjunta el objeto secundario. No obstante, los enlaces de índice

no son extremos principal-secundario, sino que tienen su propio conjunto de operaciones API de enumeración. Para obtener más información, consulte Indexación y búsqueda.

Enlaces con tipo

Los enlaces con tipo le permiten establecer una relación entre objetos dentro o entre las jerarquías en Cloud Directory. A continuación, puede utilizar estas relaciones para consultas de información como, por ejemplo, Qué usuarios tienen el dispositivo 'xyz' o Cuáles son los dispositivos que posee el usuario 'abc'.

Puede utilizar enlaces con tipo para modelar relaciones entre diferentes objetos en su directorio. Por ejemplo, en la ilustración anterior, considere la relación entre el objeto `004`, que representa un usuario y el objeto `005`, que representa un dispositivo.

Podríamos utilizar un enlace con tipo para modelar una relación de propiedad entre los dos objetos. Se podrían añadir atributos al enlace con tipo para representar el costo de una compra o si el dispositivo es alquilado o comprado. Existen dos tipos de atributos asociados a enlaces con tipo:

- Atributos basados en identidad: un atributo de un enlace con tipo que lo distingue de otros enlaces (por ejemplo, enlaces secundarios, de adjunto, de índice). Cada faceta de enlace con tipo define un conjunto ordenado de atributos de identidad. La identidad de un enlace con tipo es el ID del objeto de origen, un identificador de faceta (tipo), los valores de sus atributos de identidad (definidos por su faceta) y el ID del objeto de destino. Los identificadores deben ser únicos dentro de un directorio único.
- Atributos opcionales: un atributo que almacena las características de seguimiento acerca del enlace con tipo que no están relacionadas con la identidad del enlace. Por ejemplo, un atributo opcional podría identificar la fecha en que se creó por primera vez el enlace con tipo o cuándo se modificó por última vez.

Al igual que con los objetos, debe crear una faceta de enlace con tipo mediante la API `CreateTypedLinkFacet` para definir la estructura del enlace con tipo y sus atributos. Las facetas de los enlaces con tipo requieren un nombre de faceta único y un conjunto de atributos que se asocian al enlace. Al diseñar la estructura del enlace con tipo, puede definir un conjunto ordenado de atributos en la faceta del enlace con tipo. Para ver un esquema de muestra de enlaces con tipo, consulte Documento de esquema de enlaces con tipo.

Se pueden utilizar atributos de enlace con tipo si necesita realizar cualquiera de las siguientes acciones:

- Permitir el filtrado de enlaces con tipo de entrada o salida. Para obtener más información, consulte Lista de enlaces con tipo.
- Representar la relación entre dos objetos.
- Realizar un seguimiento de datos administrativos sobre el enlace con tipo como, por ejemplo, la fecha en que se creó el enlace.

Tenga en cuenta lo siguiente al decidir si los enlaces con tipo son idóneos para su caso de uso:

- Los enlaces con tipo no se pueden utilizar en la especificación de objetos basados en rutas. En su lugar, debe seleccionar enlaces con tipo mediante las operaciones API `ListOutgoingTypedLinks` o `ListIncomingTypedLinks`.
- Los enlaces con tipo no participan en las operaciones API `LookupPolicy` o `ListObjectParentPaths`.
- Los enlaces con tipo entre los dos mismos objetos y en la misma dirección puede que no tengan los mismos valores de atributo. Esto puede ayudarle a evitar enlaces con tipo duplicados entre los mismos objetos.
- Los atributos adicionales se pueden utilizar cuando desee añadir información opcional.
- El tamaño combinado de todos los valores de los atributos de identidad está limitado a 64 bytes. Para obtener más información, consulte Límites en Amazon Cloud Directory.

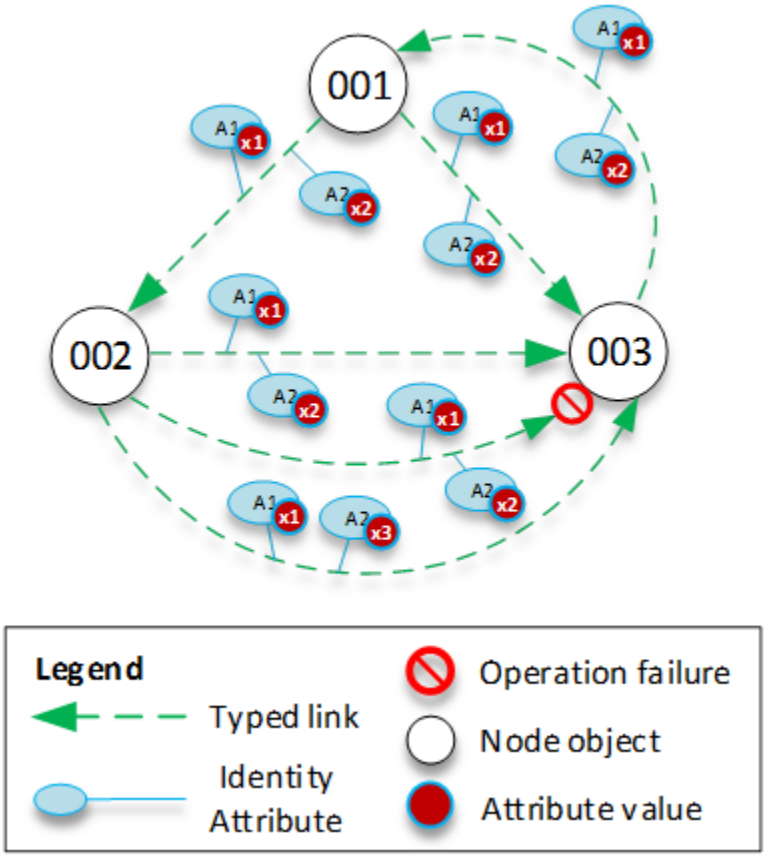
Artículo relacionado con el blog del Cloud Directory

- Use Amazon Cloud Directory Typed Links to Create and Search Relationships Across Hierarchies

Identidad de enlaces con tipo

La identidad es lo que define de forma exclusiva si un enlace con tipo puede existir entre dos objetos. La excepción es cuando conecta dos objetos en una dirección con los mismos valores de atributos. Los atributos deben configurarse como `REQUIRED_ALWAYS`.

Los enlaces con tipo que se crean a partir de distintas facetas de enlace con tipo nunca entran en conflicto unos con otros. Por ejemplo, fíjese en el siguiente diagrama:



- El objeto `001` tiene enlaces con tipo y atributos (A1 y A2) con los mismos valores de atributos (x1 y x2) que van hacia diferentes objetos (`002` y `003`). Esta operación sería correcta.
- Los objetos `002` y `003` tienen un enlace con tipo entre ellos. Esta operación daría error porque no pueden existir entre objetos dos enlaces con tipo en la misma dirección con los mismos atributos.
- Los objetos `001` y `003` tiene dos enlaces con tipo entre ellos con los mismos atributos. Sin embargo, dado que los enlaces van en diferentes direcciones, esta operación sería correcta.
- Los objetos `002` y `003` tienen enlaces con tipo entre ellos con el mismo valor para A1, pero distintos valores para A2. La identidad de los enlaces con tipo tiene en cuenta todos los atributos, por lo que esta operación sería correcta.

Reglas de enlaces con tipo

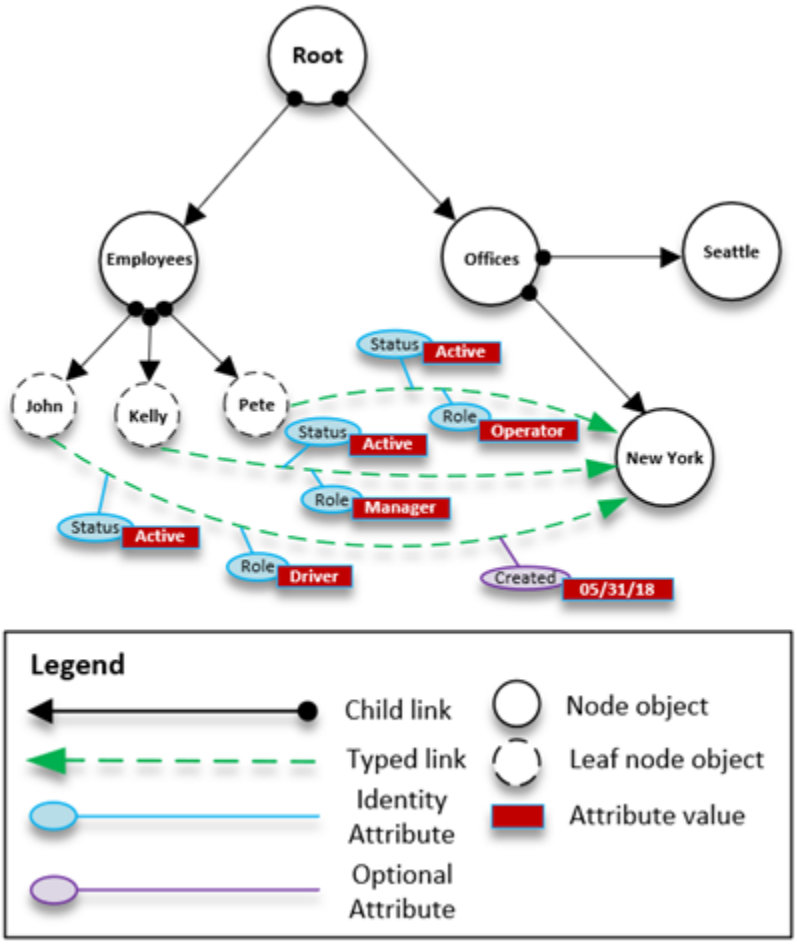
Puede añadir reglas a los atributos de enlaces con tipo si desea añadir restricciones a los atributos de los enlaces. Estas reglas son equivalentes a las reglas en los atributos de objetos. Para obtener más información, consulte Reglas de atributos.

Lista de enlaces con tipo

Cloud Directory proporciona operaciones API que puede utilizar para seleccionar enlaces con tipo entrantes o salientes desde un objeto. Puede seleccionar un subconjunto específico de enlaces con tipo en lugar de iterar en cada vínculo con tipo. También puede especificar una faceta concreta de enlace con tipo para filtrar solo los enlaces de ese tipo.

Puede filtrar los enlaces con tipo según el orden en el que se definen los atributos en la faceta del enlace con tipo. Puede proporcionar filtros por rango para varios atributos. Al proporcionar rangos para una selección de enlaces con tipo, cualquier rango inexacto debe especificarse al final. Cualquier atributo sin un rango especificado, se supone que coincide con todo el rango. Los filtros se interpretan en el orden de los atributos que se definen en la faceta del enlace con tipo, no en el orden que se proporcionan en cualquier llamada a la API.

Por ejemplo, en el siguiente diagrama, imagine un Cloud Directory que se utilice para almacenar información sobre empleados y sus habilidades.



Imaginemos que modelamos las habilidades de nuestro empleado con un enlace con tipo denominado `EmployeeCapability`, que se configura con tres atributos de cadena: `Status`, `Role` y `Created`. Los siguientes filtros se admiten en las operaciones API `ListIncomingTypedLinks` y `ListOutgoingTypedLinks`.

- Faceta = `EmployeeCapability`, Estado = `Active`, Rol = `Driver`
  - Selecciona los empleados activos que son conductores. Este filtro incluye dos coincidencias exactas.
- Faceta = `EmployeeCapability`, Estado = `Active`, Rol = `Driver`, Creado = `05/31/18`
  - Selecciona los empleados activos que son conductores y de quién son las facetas que se crearon a partir del 31 de mayo de 2018.
- Faceta = `EmployeeCapability`, Estado = `Active`
  - Selecciona todos los empleados activos.
- Faceta = `EmployeeCapability`, Estado = `Active`, Rol = `A` a `M`
  - Selecciona los empleados activos con roles desde `A` hasta `M`.
- Faceta = `EmployeeCapability`
  - Esto selecciona todos los enlaces con tipo del tipo `EmployeeCapability`.

NO se admitirían los siguientes filtros:

- Faceta = `EmployeeCapability`, Estado entre `A` y `C`, Rol = `Driver`
  - Este filtro no está permitido porque los rangos deben aparecer al final del filtro.
- Faceta = `EmployeeCapability`, Rol = `Driver`
  - Este filtro no está permitido porque el rango de estado implícito no es una coincidencia exacta y no aparece al final de la lista de rangos.
- Estado = `Active`
  - Este filtro no está permitido porque la faceta de enlace con tipo no está especificada.

Esquema de enlaces con tipo

Puede crear facetas de enlaces con tipo de dos modos. Puede administrar las faceta de enlaces con tipo a partir de llamadas a API, incluidas `CreateTypedLinkFacet`, `DeleteTypedLinkFacet` y `UpdateTypedLinkFacet`. También puede cargar un documento JSON que represente el esquema en una única llamada a la API `PutSchemaFromJson`. Para obtener más información, consulte

Formato de esquemas JSON. Para ver un esquema de muestra de enlaces con tipo, consulte Documento de esquema de enlaces con tipo.

Los tipos de cambios permitidos en diferentes fases del ciclo de vida de desarrollo del esquema son similares a los cambios permitidos para la manipulación de las facetas de objetos. Los esquemas en el estado de desarrollo admiten cualquier cambio. Los esquemas en el estado publicado son inmutables y no se admite ningún cambio. Solo se permiten determinados cambios en los esquemas que se aplican a un directorio de datos. Una vez que define el orden y los atributos en una faceta de enlace con tipo aplicado, ese orden no se puede cambiar.

Otras dos operaciones API enumeran las facetas y sus atributos:

- `ListTypedLinkFacetAttributes`
- `ListTypedLinkFacetNames`

Interacción de enlaces con tipo

Una vez que se ha creado una faceta de enlace con tipo, está listo para empezar a crear e interactuar con enlaces con tipo. Para adjuntar y separar enlaces con tipo, utilice las operaciones API `AttachTypedLink` y `DetachTypedLink`.

El `TypedLinkSpecifier` es una estructura que contiene toda la información para identificar de forma exclusiva un enlace con tipo. Dentro de esa estructura, puede encontrar `TypedLinkFacet`, `SourceObjectID`, `DestinationObjectID` e `IdentityAttributeValues`. Se utilizan para especificar de forma exclusiva el enlace con tipo en el que se está operando. La operación API `AttachTypedLink` devuelve un especificador de enlace con tipo, mientras que la operación API `DetachTypedLink` acepta uno como entrada. De forma similar, las operaciones API `ListIncomingTypedLinks` y `ListOutgoingTypedLinks` proporcionan especificadores de enlace con tipo como salida. También puede crear desde cero un especificador de enlace con tipo. La lista completa de operaciones API relacionadas con enlaces con tipo incluye lo siguiente:

- `AttachTypedLink`
- `CreateTypedLinkFacet`
- `DeleteTypedLinkFacet`
- `DetachTypedLink`
- `GetLinkAttributes`
- `GetTypedLinkFacetInformation`

- ListIncomingTypedLinks
- ListOutgoingTypedLinks
- ListTypedLinkFacetNames
- ListTypedLinkFacetAttributes
- UpdateLinkAttributes
- UpdateTypedLinkFacet

> **Note**
>
> No se admiten referencias de atributo ni la actualización de enlaces con tipo. Para actualizar un enlace con tipo, debe eliminarlo y añadir la versión actualizada.

Filtros de rango

Varias API de lista de Cloud Directory permiten la especificación de un filtro en forma de rango. Estos filtros le permiten seleccionar con eficacia subconjuntos de enlaces que se adjuntan al nodo especificado.

Los rangos por lo general se suministran como un mapa (matriz de pares clave-valor) cuyas claves son identificadores de atributos y cuyos valores son los rangos correspondientes. Esto permite filtrar enlaces cuyas identidades constan de uno o más atributos. Por ejemplo, una configuración TypedLink para modelar una relación de rol para determinar permisos podría tener atributos RoleType y Authorizer. Una llamada ListOutgoingTypedLinks podría especificar a continuación rangos para filtrar el resultado en RoleType:”Admin” y Authorizer:”Julia”. El mapa de rangos utilizado para filtrar una única solicitud de lista debe contener solo atributos que definan la identidad del enlace (una OrderedIndexedAttributeList de un índice o un IdentityAttributeOrder de TypedLink), pero no debe contener rangos para todos. Los rangos que falten se rellenarán automáticamente con rangos que abarquen todos los valores posibles (desde FIRST a LAST).

Si piensa en cada atributo como definitorio de un dominio plano e independiente de valores, las estructuras de rango definen dos puntos lógicos en ese dominio (los puntos de inicio y de fin) y el rango coincide con todos los puntos posibles entre esos puntos. El StartValue y el EndValue de la estructura de rangos definen la base de estos dos puntos y los “modos” los definen aún más para indicar si cada punto debe incluirse o excluirse del rango. En el ejemplo anterior RoleType:”Admin”, los valores para el atributo RoleType serían ambos “Admin” y los modos son

en ambos casos “INCLUSIVE” (escrito como [“Admin” to “Admin”]). Un filtro para una llamada ListIndex donde el índice se define en el LastName de una faceta User podría utilizar StartValue=”D”, StartMode=INCLUSIVE, EndValue:”G”, EndMode:EXCLUSIVE para delimitar la lista a nombres que comiencen por D, E o F.

El punto de inicio de un rango siempre debe preceder o ser igual al punto final. Cloud Directory devolverá un error si EndValue precede a StartValue. Los valores también deben ser del mismo tipo primitivo que el atributo al que filtran, valores String para un atributo String, Integer para un atributo Integer, y así sucesivamente. Por ejemplo, StartValue=“D”, StartMode=EXCLUSIVE, EndValue=“D”, EndMode=INCLUSIVE no es válido, porque el punto final incluye el valor, mientras que el punto de inicio sigue el valor.

Existen tres modos especiales que pueden utilizar los puntos inicial o final. Los siguientes modos no requieren que se especifique el campo de valor correspondiente, ya que implican una posición por sí solos.

- FIRST: precede a todos los valores posibles en el dominio. Cuando se utiliza para el punto de inicio, coincide con todos los valores posibles desde el principio del dominio hasta el punto final. Cuando se utiliza para el punto final, ningún valor en el domino coincidirá con el rango.
- LAST: sigue todos los valores posibles en el dominio. Cuando se utiliza para el punto final, coincide con todos los valores posibles que siguen el punto de inicio, incluidos los valores que faltan. Cuando se utiliza para el punto de inicio, ningún valor en el domino coincidirá con el rango.
- LAST_BEFORE_MISSING_VALUES: este modo solo es útil para atributos opcionales cuando el valor puede omitirse (consulte Valores que faltan). Corresponde al punto entre los valores que faltan y los valores de dominio reales. Cuando se utiliza para el punto final, coincide con todos los valores del domino que no faltan que siguen al punto de inicio. Cuando se utiliza para el punto de inicio, excluye todos los valores de dominio que no faltan. Si el atributo es obligatorio, este modo equivale a LAST, ya que no pueden faltar valores.

Limitaciones de rango múltiple

Cloud Directory limita los patrones con múltiples atributos para garantizar un procesamiento de solicitudes eficiente y de baja latencia. Cada enlace con varios atributos identificativos los especifica en un orden bien definido. El ejemplo del rol anterior define el atributo RoleType como el más importante y el atributo Authorizer como el menos importante. Una solicitud de List puede especificar un solo rango de "cualificación" que no es 1) un único valor ni 2) abarca todos los valores posibles (puede haber múltiples rangos que coincidan con estos dos requisitos). Los rangos de atributos

más importantes que el atributo de rango de cualificación deben especificar un solo valor y los rangos de rangos menos importantes deben abarcar todos los valores posibles. En el ejemplo de rol, los conjuntos de filtros (RoleType:”Admin”, Authorizer:[“J” to “L”]) (valor único + rango de cualificación), (RoleType:[”Admin” to “User”]) (rango de cualificación + rango implícito que abarca todo) y (RoleType:[FIRST to LAST]) (dos rangos que abarcan valores, uno implícito) son todos ejemplos válidos de conjuntos de filtro válidos. (RoleType:[FIRST to LAST], Authorizer:”Julia”) no es un conjunto válido, ya que el rango que abarca es más importante que el rango de valor único.

Entre algunos patrones útiles al rellenar las estructuras de rango se incluyen los siguientes:

Coincidentes con un solo valor

Especifique el valor tanto para StartValue como para EndValue y defina ambos modos en "INCLUSIVE".

Ejemplo: `StartValue=“Admin”, StartMode=INCLUSIVE, EndValue=“Admin”, EndMode=INCLUSIVE`

Coincidente con un prefijo

Especifique el prefijo como StartValue con el modo INCLUSIVE y el primer valor tras el prefijo como EndValue con modo EXCLUSIVE.

Ejemplo: `StartValue=“Jo”, StartMode=INCLUSIVE, EndValue=“Jp”, EndMode=EXCLUSIVE (“p” is the next character value after “o”)`

Filtrado de superior a un valor

Especifique el valor para StartValue con el modo EXCLUSIVE y LAST como EndMode (o LAST_BEFORE_MISSING_VALUES para excluir los valores que faltan, si procede).

Ejemplo: `StartValue=127, StartMode=EXCLUSIVE, EndValue=null, EndMode=LAST`

Filtrado de menos o igual a un valor

Especifique el valor de EndValue con el modo INCLUSIVE y FIRST como StartMode.

Valores que faltan

Cuando un atributo está marcado como opcional en el esquema, es posible que el valor "falte", ya que no era necesario suministrarlo cuando la faceta se adjuntó o el atributo podría haberse

eliminado posteriormente. Si el objeto cuyo valor falta se encuentra adjunto a un índice, el enlace de índice sigue presente, pero se traslada al final del conjunto de enlaces. Una llamada `ListIndex` devolverá primero cualquier vínculo en el que estén presentes todos los atributos indexados antes de devolver enlaces en los que falte uno o más. Esto es más o menos similar a un valor NULL de una base de datos relacional, con estos valores ordenados después de los valores que no sean NULL. Puede especificar si un rango incluye estos valores que faltan o no eligiendo los modos LAST o LAST_BEFORE_MISSING_VALUES. Por ejemplo, puede proporcionar un filtro en una llamada ListIndex para devolver solo los valores que faltan en un índice filtrando con el rango [LAST_BEFORE_MISSING_VALUES to LAST].

Acceso a objetos

Se puede acceder a los objetos de un directorio mediante la ruta o bien `objectIdentifier`.

RutaRuta: cada objeto de un árbol de Cloud Directory se puede identificar y encontrar por medio del nombre de ruta que describe cómo llegar al mismo. La ruta parte de la raíz del directorio (Nodo `000` en la ilustración anterior). La notación de la ruta comienza con el enlace etiquetado con una barra inclinada (/) y siguen los enlaces secundarios separados mediante el separador de ruta (también una barra inclinada) hasta llegar a la última parte de la ruta. Por ejemplo, el objeto `005` de la figura anterior se puede identificar utilizando la ruta `/group/a/d`. Varias rutas pueden identificar un objeto, ya que los objetos pueden tener varios nodos hijos de varios padres. La siguiente ruta también se puede utilizar para identificar el objeto `005` : /group/b/e

ObjectIDentifier— cada objeto del directorio tiene un identificador global único, que es el`ObjectIdentifier`.`ObjectIdentifier`se devuelve como parte del`CreateObject`Llamada a la API. También puede obtener el `ObjectIdentifier` utilizando la llamada a API `GetObjectInformation`. Por ejemplo, para recuperar el identificador de objeto del objeto `005`, puede llamar a `GetObjectInformation` con la referencia de objeto como la ruta que conduce al objeto, que es group/b/e o group/a/d.

```
GetObjectInformationRequest request = new GetObjectInformationRequest()
    .withDirectoryArn(directoryArn)
    .withObjectReference("/group/b/e")
    .withConsistencyLevel(level)
GetObjectInformationResult result = cdClient.getObjectInformation(request)
String objectIdentifier = result.getObjectIdentifier()
```

Rellenar objetos

Se pueden añadir nuevas facetas a un objeto utilizando la llamada a API `AddFacetToObject`. El tipo del objeto se determina en función de las facetas que se adjuntan al objeto. El adjunto de objeto en un directorio funciona en función del tipo del objeto. Al adjuntar un objeto, recuerde estas reglas:

- Un objeto de nodo hijo no puede tener secundarios.
- Un objeto de nodo puede tener varios secundarios.
- Un objeto del tipo de política no puede tener secundarios y puede tener cero o un principal.

Actualización de objetos

Puede actualizar un objeto de varias formas:

1. Utilice la operación `UpdateObjectAttributes` para actualizar los atributos de faceta individuales de un objeto.
2. Utilice la operación `AddFacetToObject` para añadir nuevas facetas a un objeto.
3. Utilice la operación `RemoveFacetFromObject` para eliminar las facetas existentes de un objeto.

Eliminación de objetos

Un objeto adjunto debe cumplir determinadas condiciones antes de poder eliminarlo de un directorio:

1. Debe desconectar el objeto del árbol. Puede desconectar un objeto solo cuando no tenga secundarios. Si el objeto tiene secundarios, debe desconectar todos los secundarios en primer lugar.
2. Puede eliminar un objeto desconectado solo si se eliminan todos los atributos de ese objeto. Puede eliminar atributos en un objeto eliminando cada faceta adjunta a dicho objeto. Puede obtener una lista de las facetas adjuntas a un objeto llamando a `GetObjectInformation`.
3. Un objeto también debe no tener ningún principal, ningún adjunto de política y ningún adjunto de índice.

Dado que un objeto se debe desconectar totalmente del árbol para que se elimine, debe utilizar el identificador de objetos para eliminarlo.

Consulta de objetos

En esta sección se explican diversos elementos relevantes para consultar objetos en un directorio.

Recorrido de directorios

Dado que Cloud Directory es un árbol, puede consultar objetos de arriba abajo usando la herramienta`ListObjectChildren`o de abajo hacia arriba usando el método`ListObjectParents`Operación API.

Búsqueda de política

Dada una referencia de objeto, la operación de API `LookupPolicy` devuelve todas las políticas que se han adjuntado a lo largo de su ruta (o rutas) a la raíz de arriba abajo. Todas las rutas que no dirigen hacia la raíz se ignoran. Se devuelven todos los objetos de tipo de política.

Si el objeto es un nodo hijo, puede tener varias rutas a la raíz. Esta llamada devuelve solo una única ruta para cada llamada. Para obtener rutas adicionales, utilice el token de paginación.

Consulta de índice

Cloud Directory admite una potente funcionalidad de consulta de índice con el uso de los siguientes rangos:

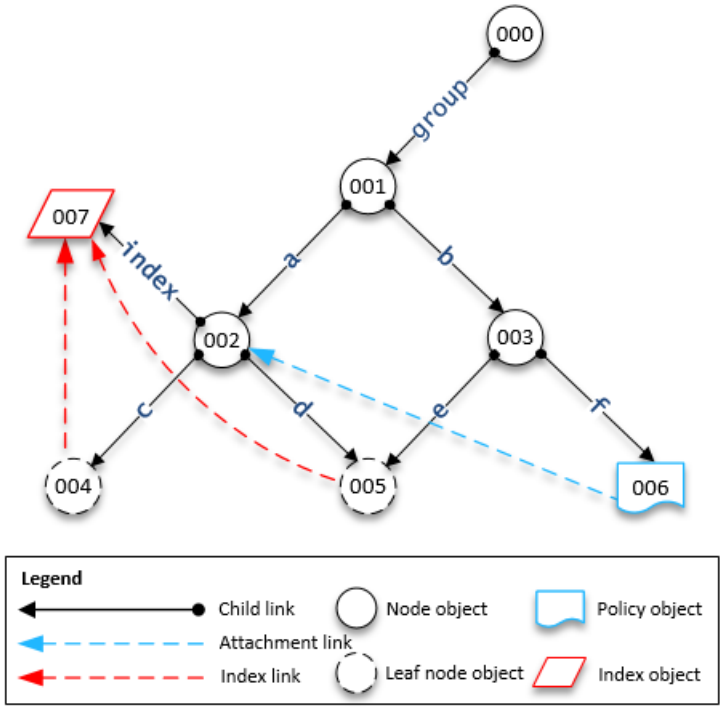
- FIRST: empieza desde el primer valor de atributo indexado. El valor de atributo de inicio es opcional.
- LAST: devuelve valores de atributo hasta el final del índice, incluidos los valores que faltan. El valor de atributo final es opcional.
- LAST_BEFORE_MISSING_VALUES: devuelve valores de atributo hasta el final del índice, excluidos los valores que faltan.
- INCLUSIVE: incluye el valor de atributo que se está especificando.
- EXCLUSIVE: excluye el valor de atributo que se está especificando.

Listado de ruta principal

Con la llamada a la API `ListObjectParentPaths`, puede recuperar todas las rutas principales disponibles para cualquier tipo de objeto (nodo, nodo hijo, nodo de políticas, nodo de índice). Esta operación de la API puede ser útil cuando necesita evaluar todos los principales de un objeto. La llamada devuelve todos los objetos desde la raíz del directorio hasta el objeto solicitado. También

devuelve el número de rutas en función de `MaxResults` definido por el usuario, en caso de varias rutas a la principal. El orden de las rutas y los nodos devuelto es coherente entre varias llamadas de API a menos que los objetos se eliminen o se muevan. Las rutas que no dirigen a la raíz del directorio se ignoran desde el objeto de destino.

Para ver un ejemplo de cómo funciona, supongamos que un directorio tiene una jerarquía de objetos similar a la ilustración que se muestra a continuación.



Las formas numeradas representan los diferentes objetos. El número de flechas entre dicho objeto y la raíz del directorio (`000`) representa la ruta completa y se expresaría en la salida. En la siguiente tabla se muestran las solicitudes y respuestas de consultas realizadas a los objetos de nodo hijo específico de la jerarquía.

Consultas de ejemplo en objetos

Solicitud	Respuesta
`004, PageToken : null, MaxResults: 1`	`[{/group/a/c], [000, 001, 002, 004]}], PageToken: null`

Solicitud	Respuesta
`005, PageToken : null, MaxResults: 2`	`[{/group/a/d, [000, 001, 002, 005]}, { /group/b/e, [000, 001, 003, 005]}], PageToken: null` **Note** En este ejemplo, el objeto `005` tiene ambos nodos `002` y `003` como principales. Asimismo, puesto que `MaxResults` es 2, ambas rutas muestran objetos en una lista.
`005, PageToken : null, MaxResults: 1`	`[{/group/a/d, [000, 001, 002, 005]}], PageToken: <encrypted_next_token>`
`005, PageToken : <encrypted_next_token>, MaxResults: 1`	`[{/group/b/e, [000, 001, 003, 005]}], PageToken: null` **Note** En este ejemplo, el objeto `005` tiene ambos nodos `002` y `003` como principales. Asimismo, puesto que `MaxResults` es 1, se realizarán varias llamadas paginadas con tokens de página para obtener todas las rutas con una lista de objetos.
`006, PageToken : null, MaxResults: 1`	`[{/group/b/f, [000, 001, 003, 006]}], PageToken: null`

Solicitud	Respuesta
`007, PageToken : null, MaxResults: 1`	`[{/group/a/index, [000, 001, 002, 007]}], PageToken: null`

Niveles de coherencia

Amazon Cloud Directory es un almacén de directorios distribuido. Los datos se distribuyen en varios servidores en distintas zonas de disponibilidad. Una solicitud de escritura correcta actualiza los datos en todos los servidores. Los datos finalmente están disponibles en todos los servidores, normalmente en un segundo. Para ayudar a los usuarios del servicio, Cloud Directory ofrece dos niveles de coherencia para operaciones de lectura. En esta sección se describen los distintos niveles de coherencia y finalmente la naturaleza coherente de Cloud Directory.

Niveles de aislamiento de lectura

Al leer datos desde Cloud Directory, debe especificar el nivel de aislamiento desde el que desee leer. Los distintos niveles de aislamiento presentan compromisos entre latencia y frescura de los datos.

- FINAL— El nivel de aislamiento de la instantánea lee los datos que estén disponibles inmediatamente.Proporciona la menor latencia de cualquier nivel de aislamiento. También proporciona una vista potencialmente antigua de los datos en el directorio. El aislamiento EVENTUAL no proporciona coherencia de lectura tras escritura. Esto significa que no se garantiza que pueda leer los datos inmediatamente después de escribirlos.
- SERIALIZABLESERIALIZABLE: el nivel de aislamiento serializable proporciona el mayor nivel de coherencia que ofrece Cloud Directory. Las lecturas realizadas en el nivel de aislamiento SERIALIZABLE garantizan que reciba datos de cualquier escritura correcta. Si se ha realizado un cambio en los datos que ha solicitado y dicho cambio aún no está disponible, el sistema rechaza la solicitud con `RetryableConflictException`. Le recomendamos que vuelva a intentar estas excepciones (consulte la sección siguiente). Cuando se vuelve a intentar correctamente, las lecturas SERIALIZABLE ofrecen coherencia de lectura tras escritura.

Solicitudes de escritura

Cloud Directory garantiza que varias solicitudes de escritura no actualicen simultáneamente el mismo objeto u objetos. Si se encuentran dos solicitudes de escritura operando en los mismos objetos, una de las operaciones falla con una `RetryableConflictException`. Le recomendamos que vuelva a intentar estas excepciones (consulte la sección siguiente).

> **Note**
>
> Las respuestas `RetryableConflictException` recibidas durante las operaciones de escritura no se pueden utilizar para detectar las condiciones de carrera. Dado un caso de uso que haya mostrado precipitar esta situación, no hay garantías de que siempre se produzca una excepción. Que una excepción se produzca o no depende del orden de cada solicitud que se está procesando internamente.

RetryableConflictExceptions

Al realizar las operaciones de escritura o de lectura con un nivel de aislamiento SERIALIZABLE después de una escritura en el mismo objeto, Cloud Directory podría responder con una`RetryableConflictException`. Esta excepción indica que los servidores de Cloud Directory no han procesado todavía el contenido de la escritura anterior. Estas situaciones son transitorias y se corrigen a sí mismas con rapidez. Es importante tener en cuenta que `RetryableConflictException` no se puede utilizar para detectar ningún tipo de coherencia de lectura tras escritura. No se garantiza que un determinado caso de uso provoque esta excepción.

Le recomendamos que configure sus clientes de Cloud Directory para volver a intentar la prueba de la`RetryableConflictException`. Esta configuración proporciona un comportamiento sin errores durante la operación. El siguiente código de ejemplo demuestra cómo se puede hacer esta configuración en Java.

```
     RetryPolicy retryPolicy = new RetryPolicy(new CloudDirectoryRetryCondition(),
         PredefinedRetryPolicies.DEFAULT_BACKOFF_STRATEGY,
         PredefinedRetryPolicies.DEFAULT_MAX_ERROR_RETRY,
               true);

       ClientConfiguration clientConfiguration = new
ClientConfiguration().withRetryPolicy(retryPolicy);
```

```
 AmazonCloudDirectory client = new AmazonCloudDirectory (
   new BasicAWSCredentials(…), clientConfiguration);


public static class CloudDirectoryRetryCondition extends SDKDefaultRetryCondition {

 @Override
 public boolean shouldRetry(AmazonWebServiceRequest originalRequest,
AmazonClientException exception,
   int retriesAttempted) {

  if (exception.getCause() instanceof RetryableConflictException) {
   return true;
  }

  return super.shouldRetry(originalRequest, exception, retriesAttempted);
 }
}
```

Indexación y búsqueda

Amazon Cloud Directory es compatible con dos métodos de indexación: basada en valores y basada en tipos. La indexación basada en valores es la forma más común. Con ella, podrá indexar y buscar objetos en el directorio en función de los valores de los atributos de objeto. Con el indexado basado en tipos, podrá indexar y buscar objetos en el directorio en función de los tipos de objetos. Las facetas ayudan a definir los tipos de objetos. Para obtener más información acerca de esquemas y facetas, consulte Schemas y Facets.

Los índices de Cloud Directory permiten contar con una lista simple de otros objetos según los valores de sus atributos o facetas. Cada índice se define en el momento de crearse para que funcione con un atributo o faceta específicos. Por ejemplo, un índice puede definirse en el atributo "email" de la faceta "Person". Los índices son objetos de primera clase, lo que significa que los clientes pueden crearlos, modificarlos, listarlos y eliminarlos de forma flexible de acuerdo con las necesidades de la lógica de la aplicación.

Conceptualmente, los índices son similares a los nodos con elementos secundarios, donde los enlaces a los nodos indexados se etiquetan de acuerdo con los atributos indexados, en lugar de recibir una etiqueta cuando se adjunta el objeto secundario. No obstante, los enlaces de índice no son extremos principal-secundario, sino que tienen su propio conjunto de operaciones API de enumeración.

Es importante entender que los índices de Cloud Directory no se rellenarán automáticamente, ya que pueden estar en otros sistemas. En su lugar, utilice llamadas a la API para conectar y desconectar directamente los objetos a o desde el índice. Aunque sea un poco más laborioso, esto aporta flexibilidad para definir ámbitos de índice diferentes. Por ejemplo, puede definir un índice que solo realice un seguimiento de los elementos secundarios directos de un nodo específico. También puede definir un índice que realice un seguimiento de todos los objetos de una determinada ramificación bajo una raíz local, como todos los nodos dentro de un departamento. También puede hacer ambas cosas al mismo tiempo.

Temas

- Ciclo de vida del índice
- Indexación basada en facetas
- Índices únicos y no únicos

Ciclo de vida del índice

Puede utilizar las siguientes llamadas a la API para ayudar en el ciclo de vida de desarrollo de índices.

1. Puede crear índices con la llamada al API `CreateIndex`. Puede indicar una estructura de definición del índice que describa los atributos de los objetos adjuntos que rastreará el índice. La definición también indica si el índice debe imponer una unicidad o no. El resultado es un ID de objeto para el nuevo índice que debería adjuntarse inmediatamente a la jerarquía, como con cualquier otro objeto. Por ejemplo, puede ser una ramificación específica para albergar índices.
2. Puede adjuntar objetos al índice de forma manual con la llamada al API `AttachToIndex`. El índice, a continuación, realiza un seguimiento automático de los valores de sus atributos definidos en cada objeto adjunto.
3. Para utilizar los índices para buscar objetos con una enumeración más eficiente, llame a `ListIndex` y especifique un rango de valores que le interesen.
4. Utilice la llamada a la API `ListAttachedIndices` para enumerar los índices adjuntos a un objeto determinado.
5. Utilice la llamada a la API `DetachFromIndex` para eliminar objetos del índice de forma manual.
6. Una vez que desasocie todos los objetos del índice, podrá eliminarlo con la llamada a la API `DeleteObject`.

No existe ningún límite para el número de índices que puede incluir un directorio, aparte del límite en cuanto al espacio que ocupan todos los objetos. Los índices y sus adjuntos consumen espacio, pero parecido al que consumen los nodos y los enlaces principalsecundario. Sí hay límite en cuanto al número de índices que se pueden adjuntar a un objeto concreto. Para obtener más información, consulte Límites en Amazon Cloud Directory.

Indexación basada en facetas

Con la indexación y la búsqueda basadas en facetas puede optimizar las búsquedas en su directorio y buscar únicamente en un subconjunto del mismo. Para ello, utilice una faceta de esquema. Por ejemplo, en lugar de buscar en todos los objetos de usuarios en su directorio, puede buscar únicamente los objetos de usuario que contengan una faceta de empleado. Esto ayuda a reducir el tiempo de latencia y la cantidad de datos recuperados para la consulta.

Con la indexación basada en facetas puede utilizar las operaciones API de índices de Cloud Directory para crear y adjuntar un índice a las facetas de los objetos. También puede enumerar los resultados del índice y, a continuación, filtrar los resultados en función de determinadas facetas. Esto puede reducir de forma eficaz los tiempos de consulta y la cantidad de datos, restringiendo el ámbito de búsqueda únicamente a los objetos que contengan un determinado tipo de facetas.

El atributo "`facets`" que se utiliza con las llamadas a la API `CreateIndex` y `ListIndex` muestra el conjunto de facetas aplicadas a un objeto. Este atributo está disponible para su uso exclusivamente con las llamadas a la API `CreateIndex` y `ListIndex`. Como se ve en el siguiente código de muestra, el ARN del esquema utiliza la región, la cuenta propietaria y el ID del directorio para hacer referencia al esquema del Cloud Directory. El esquema proporcionado por el servicio no aparece en las listas.

```
String cloudDirectorySchemaArn = String.format("arn:aws:clouddirectory:%s:%s:directory/
%s/schema/CloudDirectory/1.0", region, ownerAccount, directoryId);
```

Por ejemplo, el siguiente código de muestra crea un índice basado en facetas específico de su cuenta de AWS y del directorio con el que puede enumerar todos los objetos creados con la faceta `SalesDepartmentFacet`.

> **Note**
>
> Asegúrese de utilizar el valor de facetas dentro de los parámetros, tal y como se muestra a continuación. Las instancias de «facetas» que se ve en el código de muestra se refieren a un valor proporcionado y controlado por el servicio Cloud Directory. Puede utilizarlas para la indexación, pero podrían tener acceso de solo lectura.

```
// Create a facet-based index
String cloudDirectorySchemaArn = String.format("arn:aws:clouddirectory:%s:%s:directory/
%s/schema/CloudDirectory/1.0",
    region, ownerAccount, directoryId);

facetIndexResult = clouddirectoryClient.createIndex(new CreateIndexRequest()
  .withDirectoryArn(directoryArn)
  .withOrderedIndexedAttributeList(List(new AttributeKey()
        .withSchemaArn(cloudDirectorySchemaArn)
        .withFacetName("facets")
        .withName("facets")))
```

```
        .withIsUnique(false)
        .withParentReference("/")
        .withLinkName("MyFirstFacetIndex"))
facetIndex = facetIndexResult.getObjectIdentifier()

// Attach objects to the facet-based index
clouddirectoryClient.attachToIndex(new
 AttachToIndexRequest().withDirectoryArn(directoryArn)
  .withIndexReference(facetIndex).withTargetReference(userObj))

// List all objects
val listResults = clouddirectoryClient.listIndex(new ListIndexRequest()
  .withDirectoryArn(directoryArn)
  .withIndexReference(facetIndex)
  .getIndexAttachments()

// List the index results filtering for a certain facet
val filteredResults = clouddirectoryClient.listIndex(new ListIndexRequest()
  .withDirectoryArn(directoryArn)
  .withIndexReference(facetIndex)
  .withRangesOnIndexedValues(new ObjectAttributeRange()
    .withAttributeKey(new AttributeKey()
      .withFacetName("facets")
      .withName("facets")
      .withSchemaArn(cloudDirectorySchemaArn))
    .withRange(new TypedAttributeValueRange()
      .withStartMode(RangeMode.INCLUSIVE)
      .withStartValue("MySchema/1.0/SalesDepartmentFacet")
      .withEndMode(RangeMode.INCLUSIVE)
      .withEndValue("MySchema/1.0/SalesDepartmentFacet")
    )))
```

Índices únicos y no únicos

Los índices únicos se diferencian de los no únicos en que requieren que los valores de los atributos indexados sean únicos para los objetos adjuntados al índice. Por ejemplo, puede que quiera utilizar dos índices para objetos Persona, uno único con el atributo "email", y uno no único con el atributo "apellido". El índice del apellido permite adjuntar varios objetos Persona que tengan el mismo apellido. Por otra parte, la llamada `AttachToIndex` que se dirige al índice de email devuelve un error `LinkNameAlreadyInUseException` si ya tiene adjuntada una Persona con el mismo atributo de email. Tenga en cuenta que el error no elimina el objeto de Persona. Por tanto, una aplicación

podría crear el objeto Persona, adjuntarlo a la jerarquía y a los índices, todo en una única solicitud de lote. De este modo, se garantiza que, si se infringe la unicidad en cualquiera de los índices, el objeto y todos sus adjuntos se devuelvan automáticamente.

Cómo Cloud Directory

En esta sección se presentan todos los procedimientos de uso y mantenimiento de un entorno en Cloud Directory.

Temas

Administrar sus directorios

En esta sección se describe cómo realizar las tareas de directorio comunes para su entorno de Cloud Directory.

Temas

Creación de su directorio

Antes de crear un directorio en Amazon Cloud Directory, AWS Directory Service requiere que le aplique un esquema primero. Un directorio no se puede crear sin un esquema y normalmente tiene un esquema aplicado. No obstante, también puede servirse de las operaciones de API de Cloud Directory para aplicar esquemas adicionales a un directorio. Para obtener más información, consulte ApplySchema en la guía de referencia de API de Amazon Cloud Directory.

Para crear un Cloud Directory

1. En el navegadorAWS Directory Servicepanel de navegación, enDirectorio en la nube, elijaDirectorios.
2. SeleccionarConfigurar el Cloud Directory.

3. UNDERElija un esquema para aplicar al nuevo directorioEn, escriba el nombre sencillo de su directorio, como por ejemplo`User Repository`En y, a continuación, elija una de las siguientes opciones:

   - Esquema administrado
   - Esquema de muestra
   - Esquema personalizado

   Los esquemas de ejemplo y los esquemas personalizados se colocan en el cuadro de diálogoDesarrollo, de forma predeterminada. Para obtener más información sobre los estados de esquema, consulte Ciclo de vida de esquemas. Antes de que un esquema se pueda aplicar a un directorio, debe convertirse al estado Published (Publicado). Para publicar correctamente un esquema de muestra mediante la consola, debe disponer de permisos para las siguientes acciones:

   - `clouddirectory:Get*`
   - `clouddirectory:List*`
   - `clouddirectory:CreateSchema`
   - `clouddirectory:CreateDirectory`
   - `clouddirectory:PutSchemaFromJson`
   - `clouddirectory:PublishSchema`
   - `clouddirectory:DeleteSchema`

   Dado que los esquemas de muestra son plantillas de solo lectura proporcionadas por AWS, no se pueden publicar directamente. En su lugar, cuando elija crear un directorio en función de un esquema de muestra, la consola crea una copia temporal del esquema de muestra que ha seleccionado y la coloca en el cuadro de diálogoDesarrolloestado. A continuación, crea una copia de dicho esquema de desarrollo y la coloca en el estado Published (Publicado). Una vez publicado, el esquema de desarrollo se elimina, por lo que la acción `DeleteSchema` es necesaria a la hora de publicar un esquema de muestra.

4. Seleccione Siguiente.
5. Revise la información del directorio y haga los cambios necesarios. Cuando la información sea correcta, elija Create (Crear).

Eliminar su directorio

Utilice el siguiente procedimiento para eliminar un directorio de Cloud Directory.

> **Note**
>
> Para poder eliminar un directorio, primero debe deshabilitar. Para obtener instrucciones, consulte Deshabilitar su directorio.

Para eliminar un directorio

1. En el navegadorAWS Directory Servicepanel de navegación, enDirectorio en la nubeEn, seleccioneDirectorios.
2. Seleccione la opción de la tabla situada junto al ID de directorio que desea eliminar.
3. Elija Actions (Acciones).
4. SeleccionarEliminar
5. En el navegadorEliminar directorio, confirme la operación escribiendo el nombre del directorio y, a continuación, elijaEliminar.

Deshabilitar su directorio

Utilice el siguiente procedimiento para deshabilitar un directorio de Cloud Directory.

Para deshabilitar un directorio

1. En el navegadorAWS Directory Servicepanel de navegación, enDirectorio en la nubeEn, seleccioneDirectorios.
2. Seleccione la opción de la tabla situada junto al ID de directorio que desea deshabilitar.
3. Elija Actions (Acciones).
4. SeleccionarDeshabilitar

Habilitar su directorio

Utilice el siguiente procedimiento para habilitar un directorio deshabilitado previamente en Cloud Directory.

Para habilitar un directorio

1. En el navegadorAWS Directory Servicepanel de navegación, enDirectorio en la nubeEn, seleccioneDirectorios.
2. Seleccione la opción de la tabla situada junto al ID de directorio que desea habilitar.
3. Elija Actions (Acciones).
4. SeleccionarHabilitar

Administrar el esquema

En esta sección se describe cómo realizar las tareas de esquema comunes para su entorno de Cloud Directory.

Temas

- Cree el esquema
- Eliminar un esquema
- Descargar un esquema
- Publicar un esquema
- Actualización de su esquema
- Actualización de su esquema

Cree el esquema

Amazon Cloud Directory permite cargar un archivo JSON compatible con la creación de esquemas. Para crear un esquema nuevo, puede crear su propio archivo JSON desde cero o descargar uno de los esquemas existentes registrados en la consola. A continuación, cárguelo como un esquema personalizado. Para obtener más información, consulte Esquemas personalizados.

También puede crear, eliminar, descargar, listar, publicar, actualizar y actualizar esquemas mediante la API de Cloud Directory. Para obtener más información sobre las operaciones de API relacionadas con los esquemas, consulte la guía de referencia de API de Amazon Cloud Directory.

Elija uno de los procedimientos que aparece a continuación, según prefiera.

Para crear un esquema personalizado

1. En el navegadorConsola de AWS Directory Servicepanel de navegación, enDirectorio en la nube, elijaEsquemas de.
2. Cree un archivo JSON con todas las nuevas definiciones de esquema. Para obtener más información acerca de cómo dar formato a un archivo JSON, consulte Formato de esquemas JSON.
3. En la consola de, seleccioneCargar nuevo esquema.
4. En el navegadorCargar nuevo esquemaEscriba un nombre para el esquema.
5. SelectElegir archivoEn, seleccione el nuevo archivo JSON que acaba de crear y, a continuación, elijaOpen.
6. Seleccione Upload. Al hacerlo, se añade un esquema nuevo a la biblioteca de esquemas y se pone enDesarrolloestado. Para obtener más información sobre los estados de esquema, consulte Ciclo de vida de esquemas.

Para crear un esquema personalizado basado en otro existente en la consola

1. En el navegadorConsola de AWS Directory Servicepanel de navegación, enDirectorio en la nube, elijaEsquemas de.
2. En la tabla donde se enumeran los esquemas, seleccione la opción situada cerca del esquema que desea copiar.
3. Elija Actions (Acciones).
4. SeleccionarDescargar esquema.
5. Cambie el nombre del archivo JSON, edítelo según sea necesario y, a continuación, guarde el archivo. Para obtener más información acerca de cómo dar formato a un archivo JSON, consulte Formato de esquemas JSON.
6. En la consola de, seleccioneCargar nuevo esquemaEn, seleccione el archivo JSON que acaba de editar y, a continuación, elijaOpen.

   Al hacerlo, se añade un esquema nuevo a la biblioteca de esquemas y se pone enDesarrolloestado. Para obtener más información sobre los estados de esquema, consulte Ciclo de vida de esquemas.

Eliminar un esquema

Utilice el siguiente procedimiento para eliminar un esquema de Cloud Directory.

Para eliminar un esquema

1. En el navegadorConsola de AWS Directory Servicepanel de navegación, enDirectorio en la nubeEn seleccioneEsquemas de.
2. Seleccione la opción de la tabla situada junto al nombre del esquema que desea eliminar.
3. Elija Actions (Acciones).
4. SeleccionarEliminar
5. En el navegadorEliminar esquema, confirme la operación seleccionandoEliminar.

Descargar un esquema

Utilice el siguiente procedimiento para descargar un esquema.

Para descargar un esquema

1. En el navegadorConsola de AWS Directory Servicepanel de navegación, enDirectorio en la nubeEn seleccioneEsquemas de.
2. Seleccione la opción en la tabla junto al nombre del esquema que desea descargar.
3. Elija Actions (Acciones).
4. SeleccionarDescargar esquema

Publicar un esquema

Utilice el siguiente procedimiento para publicar un esquema en Cloud Directory.

Para publicar un esquema

1. En el navegadorConsola de AWS Directory Servicepanel de navegación, enDirectorio en la nubeEn seleccioneEsquemas de.
2. Seleccione la opción de la tabla junto al nombre del esquema que desea publicar.
3. Elija Actions (Acciones).

4. SeleccionarPublicación
5. En el navegadorPublicar esquemaEn, proporcione la siguiente información:
   a. Nombre del esquema
   b. Versión principal
   c. Versión secundaria
6. Elija Publish.

Actualización de su esquema

Utilice el siguiente procedimiento para actualizar un esquema en Cloud Directory.

Para actualizar un esquema

1. En el navegadorConsola de AWS Directory Servicepanel de navegación, enDirectorio en la nubeEn seleccioneEsquemas de.
2. Seleccione la opción de la tabla junto al nombre del esquema que desea actualizar.
3. Elija Actions (Acciones).
4. Elija Update (Actualizar).
5. En el navegadorActualización de esquema, opcionalmente modifique elNombre del esquemaEn, seleccioneElegir archivoPara aplicar o eliminar facetas y atributos.
6. Elija Update (Actualizar).

Actualización de su esquema

La actualización de un esquema agregará las facetas y atributos que elija al esquema publicado que seleccione. Utilice el siguiente procedimiento para actualizar un esquema publicado.

Para actualizar un esquema

1. En el navegadorConsola de AWS Directory Servicepanel de navegación, enDirectorio en la nubeEn seleccioneEsquemas de.
2. Seleccione la opción de la tabla junto al nombre del esquema que desea actualizar.
3. Elija Actions (Acciones).
4. SeleccionarUpgrade

5. En el navegadorActualización de esquema publicadoEn, elija una de las siguientes opciones y, a continuación, elijaUpgrade:

   - Elija de su lista actual de esquemas de desarrollo
   - Cargar un nuevo archivo de esquema (JSON)

6. SeleccionarActualización de.

Seguridad de Amazon Cloud Directory

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y un centro de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre 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 los servicios de AWS en la nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores de terceros prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información acerca de los programas de conformidad que se aplican a Amazon Cloud Directory, consulteServicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad dependerá del servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Cloud Directory. En los siguientes temas, se le mostrará cómo configurar Cloud Directory para que cumpla sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros servicios de AWS que le ayudarán a monitorear y a proteger los recursos de Cloud Directory.

Temas

Identity and Access Management os en Amazon Cloud Directory

El acceso a Amazon Cloud Directory requiere credenciales que AWS puede utilizar para autenticar las solicitudes. Estas credenciales deben tener permisos para obtener acceso a los recursos de

AWS. En las siguientes secciones presentamos más detalles sobre cómo usarAWS Identity and Access Management (IAM)Para Cloud Directory proteger los recursos de controlando quién puede obtener acceso a ellos:

- Authentication
- Control de acceso

Authentication

Puede obtener acceso a AWS con los siguientes tipos de identidades:

- Usuario de la cuenta raíz de AWS: cuando se crea por primera vez una cuenta de AWS, se comienza con una identidad de inicio de sesión único que tiene acceso completo a todos los servicios y recursos de AWS de la cuenta. Esta identidad recibe el nombre de usuario raíz de la cuenta de AWS y para obtener acceso a ella se inicia sesión con la dirección de correo electrónico y la contraseña que utilizó para crear la cuenta. Le recomendamos encarecidamente que no utilice el usuario raíz en sus tareas cotidianas, ni siquiera en las tareas administrativas. En lugar de ello, es mejor ceñirse a la práctica recomendada de utilizar el usuario final exclusivamente para crear al primer usuario de IAM. A continuación, guarde las credenciales del usuario raíz en un lugar seguro y utilícelas tan solo para algunas tareas de administración de cuentas y servicios.
- Usuario de IAM— UnUsuario de IAMes una identidad dentro de su cuenta de AWS que tiene permisos personalizados específicos (por ejemplo, permisos para crear un directorio en Cloud Directory). Puede utilizar un nombre de usuario de IAM y una contraseña para iniciar sesión en páginas web seguras de AWS, como la consola de administración de AWS, los foros de debate de AWS o el Centro de soporte de AWS.

  Además de un nombre de usuario y una contraseña, también puede generar claves de acceso para cada usuario. Puede utilizar estas claves cuando obtenga acceso a los servicios de AWS mediante programación, ya sea a través de uno de los varios SDK o mediante la interfaz de línea de comandos (CLI) de AWS. El SDK y las herramientas de CLI usan claves de acceso para firmar criptográficamente su solicitud. Si no utiliza las herramientas de AWS, debe firmar usted mismo la solicitud. Compatible con Cloud DirectorySignature Version 4, un protocolo para autenticar solicitudes de API de entrada. Para obtener más información sobre las solicitudes de autenticación, consulte Proceso de firma de Signature Version 4 en la Referencia general de AWS.

- Rol de IAM: un rol de IAM es una identidad de IAM que puede crear en la cuenta y que tiene permisos específicos. Un rol de IAM es similar a un usuario de IAM, ya que se trata de una AWS Identity con políticas de permisos que determinan lo que la identidad puede hacer y lo que no en AWS. Sin embargo, en lugar de asociarse exclusivamente a una persona, la intención es que cualquier usuario pueda asumir un rol que necesite. Además, un rol no tiene asociadas credenciales a largo plazo estándar, como una contraseña o claves de acceso. En su lugar, cuando se asume un rol, este proporciona credenciales de seguridad temporales para la sesión de rol. Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

  - Acceso de usuarios federados: en lugar de crear un usuario de IAM, puede usar identidades existentes de AWS Directory Service, el directorio de usuarios de la compañía o un proveedor de identidad web. Esto se conoce como usuarios federados. AWS asigna un rol a un usuario federado cuando se solicita acceso a través de un proveedor de identidad. Para obtener más información acerca de los usuarios federados, consulte Usuarios y roles federados en la Guía del usuario de IAM.

  - Acceso a los servicios de AWS: un rol de servicio es un rol de IAM que un servicio asume para realizar acciones en su nombre. Los roles de servicio ofrecen acceso solo dentro de su cuenta y no se pueden utilizar para otorgar acceso a servicios en otras cuentas. 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.

  - Aplicaciones que se ejecutan en Amazon EC2: puede utilizar un rol de IAM para administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia EC2 y realizan solicitudes de la API y la CLI de AWS. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar un rol de AWS a una instancia EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia asociado a la misma. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para obtener más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Control de acceso

Aunque disponga de credenciales válidas para autenticar las solicitudes, si no tiene permisos, no podrá crear recursos de Cloud Directory ni obtener acceso a ellos. Por ejemplo, debe tener permiso para crear un directorio en Amazon Cloud Directory.

En las siguientes secciones se describe cómo administrar los permisos para Cloud Directory. Le recomendamos que lea primero la información general.

- Información general de la administración de permisos de acceso a los recursos de Cloud Directory
- Uso de políticas basadas en identidad (políticas de IAM) para Cloud Directory
- Permisos de la API de Amazon Cloud Directory: Referencia de acciones, recursos y condiciones

Información general de la administración de permisos de acceso a los recursos de Cloud Directory

Cada recurso de AWS pertenece a una cuenta de AWS, y los permisos para crear u obtener acceso a los recursos se rigen por las políticas de permisos. Un administrador de cuenta puede asociar políticas de permisos a identidades de IAM (es decir, usuarios, grupos y roles); también hay otros servicios (como AWS Lambda) que permiten asociar políticas de permisos a recursos.

> **Note**
>
> Un administrador de la cuenta (o usuario administrador) es un usuario con privilegios de administrador. Para obtener más información, consulte Prácticas recomendadas de IAM en la Guía del usuario de IAM.

Cuando concede permisos, decide quién debe obtener los permisos, para qué recursos se obtienen permisos y qué acciones específicas desea permitir en esos recursos.

Temas

- Recursos y operaciones de Cloud Directory

- Titularidad de los recursos
- Administración del acceso a los recursos
- Especificación de elementos de política: Acciones, efectos, recursos y entidades principales
- Especificación de las condiciones de una política

Recursos y operaciones de Cloud Directory

En Cloud Directory, los recursos principales son directorios y esquemas. Estos recursos tienen nombres de recursos de Amazon (ARN) únicos asociados a ellos, tal y como se muestra en la siguiente tabla.

Tipo de recurso	Formato de ARN
Directorio	`arn:aws:clouddirectory:` *`region`*`:`*`account-id`* `:directory/` *`directory-id`*
Esquema	`arn:aws:clouddirectory:` *`region`*`:`*`account-id`* `:schema/`*`schema-state`* `/`*`schema-name`*

Para obtener más información sobre los estados de esquema y ARN, consulteEjemplos de ARN deen laInformación sobre la API de Amazon Cloud Directory.

Cloud Directory proporciona un conjunto de operaciones para trabajar con los recursos apropiados. Para obtener una lista de operaciones disponibles, consulte Amazon Cloud Directory Actions (Acciones de Amazon Cloud Directory) o Directory Service Actions (Acciones de Directory Service).

Titularidad de los recursos

El propietario del recurso es la cuenta de AWS que ha creado un recurso. Es decir, el propietario de los recursos es la cuenta de AWS de la entidad principal (cuenta raíz, usuario de IAM o rol de IAM) que autentica la solicitud que crea el recurso. Los siguientes ejemplos ilustran cómo funciona:

- Si utiliza las credenciales de cuenta raíz de su cuenta de AWS para crear un recurso de Cloud Directory, como un directorio, su cuenta de AWS será la propietaria de dicho recurso.
- Si crea un usuario de IAM en su cuenta de AWS y le concede permisos para crear recursos de Cloud Directory, el usuario también podrá crear recursos de Cloud Directory. Sin embargo, su cuenta de AWS, a la que pertenece el usuario, será la propietaria de los recursos de .

- Si crea un rol de IAM en su cuenta de AWS con permisos para crear recursos de Cloud Directory, cualquier persona que pueda asumir el rol podrá crear recursos de Cloud Directory. Su cuenta de AWS, a la que pertenece el rol, será la propietaria de los recursos de Cloud Directory.

Administración del acceso a los recursos

Una política de permisos describe quién tiene acceso a qué. En la siguiente sección se explican las opciones disponibles para crear políticas de permisos.

> **Note**
>
> En esta sección se explica cómo se usa IAM en el contexto de Cloud Directory. No se proporciona información detallada sobre el servicio de IAM. Para ver la documentación completa de IAM, consulte¿Qué es IAM?en laGuía del usuario de IAM. Para obtener más información acerca de la sintaxis y descripciones de las políticas de IAM, consulteReferencia de políticas de AWS IAMen laGuía del usuario de IAM.

Las políticas que se asocian a una identidad de IAM se denominanBasado en identidadesLas políticas de IAM y las políticas que se adjuntan a un recurso se denominanbasado en recursosPolíticas de Cloud Directory solo admite políticas basadas en identidad (políticas de IAM).

Temas

- Políticas basadas en identidad (políticas de IAM)
- Políticas basadas en recursos

Políticas basadas en identidad (políticas de IAM)

Puede asociar políticas a identidades de IAM. Por ejemplo, puede hacer lo siguiente:

- Asociar una política de permisos a un usuario o grupo de su cuenta: un administrador de la cuenta puede utilizar una política de permisos asociada a un usuario determinado para concederle permisos para crear un recurso de Cloud Directory, como un directorio nuevo.
- Asociar una política de permisos a un rol (conceder permisos entre cuentas): puede asociar una política de permisos basada en identidad a un rol de IAM para conceder permisos entre cuentas. Por ejemplo, el administrador de la Cuenta A puede crear un rol para conceder permisos entre

cuentas a otra cuenta de AWS (por ejemplo, a la Cuenta B) o a un servicio de AWS como se indica a continuación:

1. El administrador de la Cuenta A crea un rol de IAM y asocia una política de permisos a dicho rol, que concede permisos sobre los recursos de la Cuenta A.
2. El administrador de la Cuenta A asocia una política de confianza al rol que identifica la Cuenta B como la entidad principal que puede asumir el rol.
3. A continuación, el administrador de la Cuenta B puede delegar permisos para asumir el rol a cualquier usuario de la Cuenta B. De este modo, los usuarios de la Cuenta B podrán crear recursos y obtener acceso a ellos en la Cuenta A. La entidad principal de la política de confianza también puede ser la entidad principal de un servicio de AWS si desea conceder permisos para asumir el rol a un servicio de AWS.

Para obtener más información acerca del uso de IAM para delegar permisos, consulteAdministración de accesosen laGuía del usuario de IAM.

La siguiente política de permisos concede permisos a un usuario para ejecutar todas las acciones que empiezan por `Create`. Estas acciones muestran información acerca de un recurso de Cloud Directory, como un directorio o un esquema. Tenga en cuenta que el carácter comodín (*) en la`Resource`El elemento indica que las acciones están permitidas para todos los recursos de Cloud Directory de que pertenecen a la cuenta.

```
{
   "Version":"2017-01-11",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"clouddirectory:Create*",
         "Resource":"*"
      }
   ]
}
```

Para obtener más información acerca del uso de políticas basadas en identidad con Cloud Directory, consulteUso de políticas basadas en identidad (políticas de IAM) para Cloud Directory. Para obtener más información sobre usuarios, grupos, roles y permisos, consulte Identidades (usuarios, grupos y roles) en la Guía del usuario de IAM.

Políticas basadas en recursos

Otros servicios, como Amazon S3, también admiten políticas de permisos basadas en recursos. Por ejemplo, puede asociar una política a un bucket de S3 para administrar los permisos de acceso a dicho bucket. Cloud Directory no admite políticas basadas en recursos.

Especificación de elementos de política: Acciones, efectos, recursos y entidades principales

Para cada recurso de Cloud Directory (consulteRecursos y operaciones de Cloud Directory), el servicio define un conjunto de operaciones de API. Para ver una lista de operaciones de API disponibles, consulteAcciones de Amazon Cloud DirectoryorAcciones Directory Service. Para conceder permisos a estas operaciones de API, Cloud Directory define un conjunto de acciones que usted puede especificar en una política. Tenga en cuenta que la realización de una operación de la API puede requerir permisos para más de una acción.

A continuación, se indican los elementos básicos de la política:

- Recurso: en una política, se usa un nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política. Para los recursos de Cloud Directory, utilice siempre el carácter comodín (*) en las políticas de IAM. Para obtener más información, consulte Recursos y operaciones de Cloud Directory.
- Acción: utilice palabras clave de acción para identificar las operaciones del recurso que desea permitir o denegar. Por ejemplo, el`clouddirectory:GetDirectory`permiso permite a los usuarios permiso para realizar el Cloud Directory`GetDirectory`.
- Efecto: especifique el efecto que se producirá cuando el usuario solicite la acción específica; puede ser permitir o denegar. Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso para asegurarse de que un usuario no pueda obtener acceso a él, aunque otra política le conceda acceso.
- Entidad principal: en las políticas basadas en identidades (políticas de IAM), el usuario al que se asocia esta política es la entidad principal implícita. Para las políticas basadas en recursos, debe especificar el usuario, la cuenta, el servicio u otra entidad que desee que reciba permisos (se aplica solo a las políticas basadas en recursos). Cloud Directory no admite políticas basadas en recursos.

Para obtener más información acerca de la sintaxis y descripciones de las políticas de IAM, consulteReferencia de políticas de AWS IAMen laGuía del usuario de IAM.

Para ver una tabla con todas las acciones de API de Amazon Cloud Directory y los recursos a los que se aplican, consultePermisos de la API de Amazon Cloud Directory: Referencia de acciones, recursos y condiciones.

Especificación de las condiciones de una política

Al conceder permisos, puede utilizar el lenguaje de la política de acceso para especificar las condiciones en las que se debe aplicar una política. Por ejemplo, es posible que desee que solo se aplique una política después de una fecha específica. Para obtener más información sobre cómo especificar condiciones en un lenguaje de política, consulte Condition en la Guía del usuario de IAM.

Para expresar condiciones, se usan claves de condición predefinidas. No hay claves de condición específicas para Cloud Directory. No obstante, existen claves de condición que se aplican a todo AWS que puede utilizar cuando corresponda. Para ver una lista completa de claves generales de AWS, consulteClaves de condición globales disponiblesen laGuía del usuario de IAM.

Uso de políticas basadas en identidad (políticas de IAM) para Cloud Directory

Este tema contiene ejemplos de políticas basadas en identidades, donde los administradores de cuentas pueden asociar políticas de permisos a identidades de IAM (es decir, a usuarios, grupos y funciones).

> **Important**
>
> Le recomendamos que consulte primero los temas de introducción en los que se explican los conceptos básicos y las opciones disponibles para administrar el acceso a los recursos de Cloud Directory. Para obtener más información, consulte Información general de la administración de permisos de acceso a los recursos de Cloud Directory.

En las secciones de este tema se explica lo siguiente:

- Permisos necesarios para utilizar la consola AWS Directory Service
- AWS administradas (predefinidas) para Amazon Cloud Directory

Permisos necesarios para utilizar la consola AWS Directory Service

Para que un usuario trabaje con la consola AWS Directory Service, dicho usuario debe tener permisos enumerados en la política anterior o los permisos concedidos por el rol Acceso completo a Directory Service o el rol Acceso de solo lectura a Directory Service o Acceso de solo lectura a Directory Service o Acceso de solo lectura a Directory Service o Acceso deAWS administradas (predefinidas) para Amazon Cloud Directory.

Si crea una política de IAM que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para los usuarios con esa política de IAM.

AWS administradas (predefinidas) para Amazon Cloud Directory

AWS aborda muchos casos de uso comunes proporcionando políticas de IAM independientes creadas y administradas por AWS. Las políticas administradas por AWS conceden los permisos necesarios para casos de uso comunes, lo que le evita tener que investigar los permisos que se necesitan. Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Las siguientes políticas administradas por AWS, que puede asociar a los usuarios de su cuenta, son específicas de Amazon Cloud Directory:

- AmazonCloudDirectoryReadOnlyAccess: concede a un usuario o grupo acceso de solo lectura a todos los recursos de Amazon Cloud Directory. Para obtener más información, consulte la página sobre políticas de la consola de administración de AWS.
- AmazonCloudDirectoryFullAccess: concede a un usuario o grupo acceso completo a Amazon Cloud Directory. Para obtener más información, consulte la página sobre políticas de la consola de administración de AWS.

Además, hay otras políticas gestionadas por AWS que son adecuadas para su uso con otros roles de IAM. Estas políticas se asignan a los roles asociados a los usuarios en su Amazon Cloud Directory y son necesarios para que los usuarios tengan acceso a otros recursos de AWS, tales como Amazon EC2.

También puede crear políticas de IAM personalizadas que permitan a los usuarios acceder a las acciones y recursos de la API de necesarios. Puede asociar estas políticas personalizadas a los usuarios o grupos de IAM que requieran esos permisos.

Permisos de la API de Amazon Cloud Directory: Referencia de acciones, recursos y condiciones

Puede usar la siguiente tabla como referencia cuando configure Control de acceso y escriba políticas de permisos que vaya a asociar a una identidad de IAM (políticas basadas en identidad). La lista incluye cada operación de API de Amazon Cloud Directory, las acciones correspondientes a las que puede conceder permisos para realizar la acción y el recurso de AWS al que puede conceder los permisos. Las acciones se especifican en el campo `Action` de la política y el valor del recurso se especifica en el campo `Resource` de la política.

Puede utilizar claves de condiciones generales de AWS en sus políticas de Amazon Cloud Directory para expresar condiciones. Para ver una lista completa de claves generales de AWS, consulteClaves de condición globales disponiblesen laGuía del usuario de IAM.

> **Note**
>
> Para especificar una acción, use el prefijo `clouddirectory:` seguido del nombre de operación de la API (por ejemplo, `clouddirectory:CreateDirectory`).

Registro y monitoreo en Amazon Cloud Directory

Como práctica recomendada, debe monitorear su directorio para asegurarse de que los cambios queden registrados. Esto permite investigar cualquier modificación inesperada y revertir los cambios no deseados. Amazon Cloud Directory admite actualmente AWS CloudTrail, que puede utilizar para supervisar su directorio y cualquier actividad asociada.

Para obtener más información, consulteRegistro de llamadas a la API de Cloud Directory con CloudTrail.

Validación de conformidad para Amazon Cloud Directory

Auditores externos evalúan la seguridad y la conformidad de Amazon Cloud Directory en numerosos programas de conformidad de AWS. Estos incluyen ISO, SOC, PCI, FedRAMP, HIPAA y otros.

Para obtener una lista de los servicios de AWS en el ámbito de programas de conformidad específicos, consulte Servicios de AWS en el ámbito del programa de conformidad. Para obtener información general, consulte AWS Compliance Programs (Programas de conformidad de AWS).

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

Su responsabilidad de conformidad al utilizar Cloud Directory se determina en función de la confidencialidad de los datos, los objetivos de conformidad de su empresa, así como de la legislación y los reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- Guías de inicio rápido deEstas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre arquitectura para seguridad y conformidad de HIPAA– este documento técnico describe cómo las empresas pueden utilizar AWS para crear aplicaciones conformes con HIPAA.
- Recursos de conformidad de AWS: es posible que este conjunto de guías y libros de ejercicios se apliquen a su sector y ubicación.
- AWS Config: este servicio de AWS evalúa cómo las configuraciones de los recursos cumplen con las prácticas internas, las pautas del sector y las regulaciones.
- Centro de seguridad de AWS– este servicio de AWS ofrece una vista integral de su estado de seguridad en AWS que le ayuda a comprobar la conformidad con las normas del sector de seguridad y las prácticas recomendadas.

Resiliencia de Amazon Cloud Directory

La infraestructura global de AWS está conformada por regiones y zonas de disponibilidad de AWS. Las regiones de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Cloud Directory se creó sobre la base de esos principios y está disponible en varias regiones de AWS, que están físicamente aisladas entre sí. Dentro de cada región, el servicio recibe soporte adicional a través de al menos tres zonas de disponibilidad, lo que minimiza el tiempo de inactividad del servicio debido a la no disponibilidad de ninguna zona de disponibilidad única.

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

Seguridad de la infraestructura en Amazon Cloud Directory

Al tratarse de un servicio administrado, Amazon Cloud Directory está protegido por los procedimientos de seguridad de red globales de AWS que se describen en el documento técnicoAmazon Web Services: Información general de los procesos de seguridadDocumento técnico de.

Puede utilizar llamadas a la API publicadas de AWS para obtener acceso a Cloud Directory a través de la red. Los clientes deben admitir Transport Layer Security (TLS). Le recomendamos TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de enlace de FIPS. Para obtener más información acerca de los puntos de enlace de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

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 puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Soporte de transacciones

Con Amazon Cloud Directory, a menudo es necesario añadir nuevos objetos o añadir relaciones entre los objetos nuevos y los existentes para reflejar los cambios en una jerarquía real. Las operaciones por lotes pueden hacer que resulte más sencillo administrar tareas de directorios como esta, lo que aporta los siguientes beneficios:

- Las operaciones por lotes pueden minimizar el número de recorridos de ida y vuelta necesarios para escribir y leer objetos hacia y desde su directorio, lo que mejora el desempeño general de la aplicación.
- La escritura por lotes proporciona la semántica de transacciones equivalente a bases de datos SQL. Todas las operaciones se completan correctamente o, si alguna operación tiene un error, no se aplica ninguna.
- El uso de referencias de lotes le permite crear un objeto y utilizar una referencia al nuevo objeto para realizar otras acciones, como añadirlo a una relación, con lo que se reduce la sobrecarga asociada a usar una operación de lectura antes de una operación de escritura.

BatchWrite

Utilice operaciones BatchWrite para llevar a cabo varias operaciones de escritura en un directorio. Todas las operaciones de escritura por lotes se ejecutan de forma secuencial. Funciona de forma similar a las transacciones de bases de datos SQL. Si una de las operaciones de escritura por lotes falla, el efecto de toda la operación de escritura por lotes en el directorio será nulo. Si una operación de escritura por lotes falla, se produce una excepción de escritura por lotes. La excepción contiene el índice de la operación que ha fallado, además del tipo de excepción y el mensaje. Esta información puede ayudarle a identificar la causa raíz del error.

Las siguientes operaciones API se admiten como parte de la escritura por lotes:

- AddFacetToObject
- AttachObject
- AttachPolicy
- AttachToIndex
- AttachTypedLink
- CreateIndex

- CreateObject
- DeleteObject
- DetachFromIndex
- DetachObject
- DetachTypedLink
- RemoveFacetFromObject
- UpdateObjectAttributes

Nombre de referencia de lote

Los nombres de referencia de lote solo son compatibles para operaciones de escritura por lotes cuando necesite hacer referencia a un objeto como parte de la operación por lotes intermedia. Por ejemplo, suponga que, como parte de una operación de escritura por lotes determinada, se desasocian 10 objetos diferentes para asociarse con otra parte distinta del directorio. Sin la referencia de lote, tendría que leer las 10 referencias de objeto y proporcionarlas como entrada en la nueva asociación como parte de la operación de escritura por lotes. Puede usar una referencia de lote para identificar el recurso desasociado durante el proceso de asociación. Las referencias de lote pueden ser cualquier cadena normal que lleve como prefijo la almohadilla o el símbolo de hashtag (#).

Por ejemplo, en la siguiente muestra de código, un objeto con el nombre de enlace `"this-is-a-typo"` se desasocia de la raíz con un nombre de referencia de lote `"ref"`. Más adelante, el mismo objeto se asocia a la raíz con el nombre de vínculo `"correct-link-name"`. El objeto se identifica con la referencia secundaria establecida en la referencia de lote. Sin la referencia de lote, en un principio tendría que obtener el elemento `objectIdentifier` que se va a desasociar y proporcionar el de la referencia secundaria durante la operación de asociación. Con un nombre de referencia de lote evitaría esta segunda operación de lectura adicional.

```
BatchDetachObject batchDetach = new BatchDetachObject()
        .withBatchReferenceName("ref")
        .withLinkName("this-is-a-typo")
        .withParentReference(new ObjectReference().withSelector("/"));
BatchAttachObject batchAttach = new BatchAttachObject()
        .withParentReference(new ObjectReference().withSelector("/"))
        .withChildReference(new ObjectReference().withSelector("#ref"))
        .withLinkName("correct-link-name");
```

```
BatchWriteRequest batchWrite = new BatchWriteRequest()
        .withDirectoryArn(directoryArn)
        .withOperations(new ArrayList(Arrays.asList(batchDetach, batchAttach)));
```

BatchRead

Utilice operaciones BatchRead para realizar varias operaciones de lectura en un directorio. Por ejemplo, en la siguiente muestra de código, los elementos secundarios de objeto con la referencia "`/managers`" se leen junto con los atributos de objeto con la referencia "`/managers/bob`" en una sola operación de lectura por lotes.

```
BatchListObjectChildren listObjectChildrenRequest = new BatchListObjectChildren()
        .withObjectReference(new ObjectReference().withSelector("/managers"));
BatchListObjectAttributes listObjectAttributesRequest = new BatchListObjectAttributes()
        .withObjectReference(new ObjectReference().withSelector("/managers/bob"));
BatchReadRequest batchRead = new BatchReadRequest()
        .withConsistencyLevel(ConsistencyLevel.SERIALIZABLE)
        .withDirectoryArn(directoryArn)
        .withOperations(new ArrayList(Arrays.asList(listObjectChildrenRequest,
 listObjectAttributesRequest)));
BatchReadResult result = cloudDirectoryClient.batchRead(batchRead);
```

BatchRead admite las siguientes operaciones API:

- GetObjectInformation
- ListAttachedIndices
- ListIncomingTypedLinks
- ListIndex
- ListObjectAttributes
- ListObjectChildren
- ListObjectParentPaths
- ListObjectPolicies
- ListOutgoingTypedLinks
- ListPolicyAttachments
- LookupPolicy

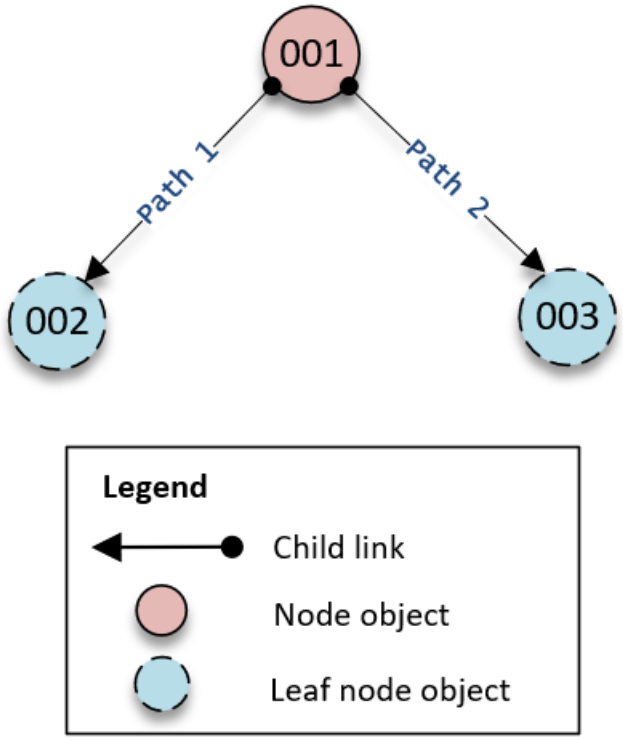
Límites de las operaciones por lotes

Cada solicitud al servidor (incluidas las solicitudes de lotes) tiene un número máximo de recursos en los que se puede operar, independientemente del número de operaciones en la solicitud. Esto le permite componer solicitudes de lotes con una alta flexibilidad mientras no supere las cantidades máximas de recursos. Para obtener más información sobre las cantidades máximas de recursos, consulte Límites en Amazon Cloud Directory.

Los límites se calculan sumando las escrituras o lecturas para cada operación dentro del lote. Por ejemplo, el límite de operación de lectura es actualmente de 200 objetos por cada llamada a API. Imaginemos que desea componer un lote que añade 9 llamadas a la API ListObjectChildren y cada llamada requiere la lectura de 20 objetos. Dado que el número total de objetos de lectura (9 x 20 = 180) no supera los 200, la operación por lotes se realizaría correctamente.

El mismo concepto se aplica con el cálculo de operaciones de escritura. Por ejemplo, el límite de la operación de escritura es actualmente 20. Si configura el lote para añadir 2 llamadas a la API UpdateObjectAttributes con 9 operaciones de escritura cada una, también se realizaría correctamente. En cualquier caso, si la operación por lotes supera el límite, la operación dará error y se generará una `LimitExceededException`.

La forma correcta de calcular el número de objetos que se incluyen dentro de un lote consiste en incluir los objetos del nodo real o nodo_hijo y si se usa un enfoque basado en ruta para iterar el árbol de directorios, también tiene que incluir cada ruta que se itere, dentro del lote. Por ejemplo, tal y como se muestra en la siguiente ilustración de un árbol de directorios básico, para leer un valor de atributo para el objeto `003`, el recuento de lectura total de objetos sería tres.



El recorrido de lecturas a lo largo del árbol funciona así:

1. Leer el objeto `001` para determinar la ruta al objeto `003`

2. Bajar a `Path 2`

3. Read object `003`

Del mismo modo, para el número de atributos necesitamos contar el número de atributos en objetos `001` y `003` para asegurarnos de no superar el límite.

Tratamiento de excepciones

A veces, pueden dar error las operaciones Batch en Cloud Directory. En estos casos, es importante saber cómo tratar este tipo de errores. El método que se utiliza para resolver errores difiere para las operaciones de lectura y de escritura.

Errores de operación de escritura por lotes

Si una operación de escritura por lotes da error, Cloud Directory da error en toda la operación por lote y devuelva una excepción. La excepción contiene el índice de la operación que ha dado error,

además del tipo de excepción y el mensaje. Si ve `RetryableConflictException`, puede volver a intentarlo con retardo exponencial. Una manera sencilla de hacerlo es duplicar la cantidad de tiempo que espera cada vez que se obtiene una excepción o un error. Por ejemplo, si la primera operación de escritura por lotes da error, espere 100 milisegundos y vuelva intentar la solicitud. Si la segunda solicitud da error, espere 200 milisegundos y vuelva a intentarlo. Si la tercera solicitud da error, espere 400 milisegundos y vuelva a intentarlo.

Errores de operación de lectura por lotes

Si una operación de lectura por lotes da error, la respuesta contiene una respuesta correcta o una respuesta de excepción. Los errores de operaciones de lectura por lotes no hacen que toda la operación de lectura por lotes da error: Cloud Directory devuelve respuestas de error o de éxito individuales para cada operación.

Artículos relacionados con el blog de Cloud Directory

- Write and Read Multiple Objects in Amazon Cloud Directory by Using Batch Operations
- How to Use Batch References in Amazon Cloud Directory to Refer to New Objects in a Batch Request

Amazon Cloud Directory

Amazon Cloud Directory ha sido sometido a auditorías para los siguientes estándares y puede se parte de su solución cuando necesite obtener una certificación de conformidad.

	Amazon Cloud Directory (FedRAMP) cumple con los requisitos de seguridad del programa federal de administración de riesgos y autorizaciones (FedRAMP) y ha recibido una autorización provisional para operar (P-ATO) de la Junta de Autorización Conjunta (JAB) de FedRAMP en la referencia FedRAMP. Para obtener más información acerca de FedRAMP, consulte Conformidad con FedRAMP.
	Amazon Cloud Directory dispone de declaración de conformidad para el estándar de seguridad de datos del sector de tarjetas de pago (PCI DSS) versión 3.2 de nivel de proveedor de servicios 1. Los clientes que utilizan los productos y servicios de AWS para almacenar, procesar o transmitir datos de titulares de tarjetas pueden utilizar Cloud Directory al administrar su propia certificación de conformidad con PCI DSS. Para obtener más información acerca de PCI DSS y sobre cómo solicitar una copia del Paquete de conformidad con PCI de AWS, consulte PCI DSS Nivel 1.

	AWS ha ampliado el programa de conformidad con la Ley de portabilidad y responsabilidad de seguros Health (HIPAA) para incluir a Amazon Cloud Directory comoServicio compatible con HIPAA. Si ha formalizado un acuerdo de socio empresarial (BAA) con AWS, puede utilizar el Cloud Directory Directory para crear aplicacio nes compatibles con HIPAA. AWS ofrece unDocumento técnico centrado en la HIPAApara los clientes que deseen informarse acerca de cómo pueden aprovechar AWS para procesar y almacenar información relacionada con el estado. Para obtener más información, consulte Conformid ad con HIPAA.
	Amazon Cloud Directory ha obtenido las certificaciones de conformidad ISO/IEC 27001, ISO/IEC 27017, ISO/ IEC 27018, e ISO 9001. Para obtener más información, consulte ISO 27001, ISO 27017, ISO 27018 e ISO 9001.
	Los informes de control de sistemas y organizaciones (SOC) son informes de análisis independientes de terceros que muestran cómo Amazon Cloud Directory logra los controles y objetivos clave de conformidad. La finalidad de estos informes es ayudarle a usted y a sus auditores a entender los controles de AWS que se han establecido como soporte a las operaciones y a la conformidad. Para obtener más información, consulte Conformidad con SOC.

Responsabilidad compartida

La seguridad, incluida la conformidad con HIPAA y PCI, es una responsabilidad compartida. Es importante comprender que el estado de conformidad de Cloud Directory no se aplica

automáticamente a las aplicaciones que ejecute en la nube de AWS. Es necesario asegurarse de que el uso que hace de los servicios de AWS cumpla con los estándares.

Uso de las acciones de API de Cloud Directory

Amazon Cloud Directory incluye un conjunto de operaciones de API que habilitan el acceso mediante programación a las funcionalidades de Cloud Directory. Puede utilizarAmazon Cloud Directory API Reference GuidePara obtener información sobre cómo realizar solicitudes a la API de Cloud Directory para crear y administrar los distintos elementos. También se tratan los componentes de las solicitudes, el contenido de las respuestas y cómo autenticar las solicitudes.

Cloud Directory proporciona todas las operaciones API necesarias para que los desarrolladores puedan crear aplicaciones nuevas. Proporciona llamadas al API dentro de las siguientes categorías:

- Operaciones de creación, lectura, actualización y eliminación (CRUD) en esquemas
- Operaciones CRUD en facetas
- Operaciones CRUD en directorios
- Operaciones CRUD en objetos (nodos, políticas, etc.)
- Operaciones CRUD en definición de índices
- Operaciones de lectura y escritura por lotes

Cómo funciona la facturación con las API de Cloud Directory

La facturación de las llamadas a la API varía en función de los tipos específicos de llamadas a la API que se realicen. Existen tarifas de facturación específicas para las llamadas a la API de lectura consistente final, las llamadas a la API de lectura consistente alta y llamadas a la API de escritura. Las llamadas a la API de metadatos son gratuitas.

Las operaciones consistentes altas se utilizan para la consistencia de lectura tras la escritura al leer un valor. Las operaciones consistentes finales se utilizan para recuperar un valor mientras se realizan actualizaciones. Con las operaciones consistentes finales, es posible que los resultados recuperados no sean los más precisos, porque el host específico desde el que se está leyendo el valor aún está procesando las actualizaciones. Sin embargo, la latencia de dichas operaciones de lectura es baja al recuperar una llamada de desempeño.

Al leer datos de Cloud Directory, debe especificar una operación de tipo de lectura consistente final o lectura consistente alta. El tipo de lectura se basa en el nivel de consistencia. Los dos niveles de consistencia son EVENTUAL para lecturas consistentes finales y SERIALIZABLE para lecturas consistentes altas. Para obtener más información, consulte Niveles de coherencia.

En la siguiente tabla se muestran todas las API de Cloud Directory y cómo pueden influir en la facturación de su cuenta de AWS.

API	Lectura consistente final [1]	Lectura consistente alta [2]	Escritura [3]	Metadatos [4]
AddFacetToObject			X	
ApplySchema				X
AttachObject			X	
AttachPolicy			X	
AttachToIndex			X	
AttachTypedLink			X	
BatchRead	X	X		
BatchWrite			X	
CreateDirectory			X	
CreateFacet				X
CreateIndex			X	
CreateObject			X	
CreateSchema				X
CreateTypedLinkFacet				X
DeleteDirectory				X
DeleteFacet				X

API	Lectura consistente final [1]	Lectura consistente alta [2]	Escritura [3]	Metadatos [4]
DeleteObject			X	
DeleteSchema				X
DetachFromIndex			X	
DetachObject			X	
DetachPolicy			X	
DetachTypedLink			X	
DeleteTypedLinkFacet				X
DisableDirectory				X
EnableDirectory			X	
GetAppliedSchemaVersion				X
GetDirectory				X
GetFacet				X
GetLinkAttributes	X	X		
GetObjectAttributes	X	X		
GetObjectInformation	X	X		

API	Lectura consistente final [1]	Lectura consistente alta [2]	Escritura [3]	Metadatos [4]
GetSchemaAsJson				X
GetTypedLinkFacetInformation				X
ListAppliedSchemaArns				X
ListAttachedIndices	X	X		
ListDevelopmentSchemaArns				X
ListDirectories				X
ListFacetAttributes				X
ListFacetNames				X
ListIncomingTypedLinks	X	X		
ListIndex	X	X		
ListManagedSchemaArns				X
ListObjectAttributes	X	X		

API	Lectura consistente final [1]	Lectura consistente alta [2]	Escritura [3]	Metadatos [4]
ListObjectChildren	X	X		
ListObjectParentPaths	X			
ListObjectParents	X	X		
ListObjectPolicies	X	X		
ListOutgoingTypedLinks	X	X		
ListPolicyAttachments	X	X		
ListPublishedSchemaArns				X
ListTagsForResource				X
ListTypedLinkFacetAttributes				X
ListTypedLinkFacetNames				X
LookupPolicy	X			
PublishSchema				X

API	Lectura consistente final [1]	Lectura consistente alta [2]	Escritura [3]	Metadatos [4]
PutSchemaFromJson				X
RemoveFacetFromObject			X	
TagResource				X
UntagResource				X
UpdateFacet				X
UpdateLinkAttributes			X	
UpdateObjectAttributes			X	
UpdateSchema				X
UpdateTypedLinkFacet				X
UpgradeAppliedSchema				X
UpgradePublishedSchema				X

[1] A las API de lectura consistente final se les llama con el nivel de consistencia EVENTUAL

[2] A las API de lectura consistente alta se les llama con el nivel de consistencia SERIALIZABLE

[3] Las API de escritura se facturan como llamadas a la API de escritura

[4] Las API de metadatos NO se facturan, sino que se clasifican como llamadas a la API de metadatos

Para obtener más información sobre facturación, consulte Precios de Amazon Cloud Directory.

Límites en Amazon Cloud Directory

Los siguientes son los límites predeterminados para Cloud Directory. A menos que se indique lo contrario, cada límite corresponde a una región.

Amazon Cloud Directory

Límites en cuanto a directorios y esquemas

Límite/Concepto	Quantity (Cantidad)
Número de atributos por faceta (incluidas las requeridas)	1 000
Número de facetas por objeto	5
Número de índices únicos con un objeto adjunto	3
Número de facetas por esquema	30
Número de reglas por atributo	5
Número de atributos con valores predeterminados por faceta	10
Número de atributos necesarios por faceta	30
Número de esquemas de desarrollo	20
Número de esquemas publicados	20
Número de esquemas aplicados	5
Número de directorios	100
Máximo de elementos de página	30
Tamaño de entrada máximo (todas las entradas combinadas)	200 KB

Límite/Concepto	Quantity (Cantidad)
Tamaño máximo de respuesta (todos los elementos de salida combinados)	1 MB
Límite de tamaño de archivo JSON para esquemas	200 KB
Longitud de nombre de faceta	64 bytes con cifrado UTF-8
Longitud de nombre de directorio	64 bytes con cifrado UTF-8
Longitud de nombre de esquema	64 bytes con cifrado UTF-8

Límites en cuanto a objetos

Límite/Concepto	Quantity (Cantidad)
Número de objetos escritos	20 por llamada de API
Número de objetos leídos	200 por llamada de API
Número de valores de atributos escritos	1000 por llamada de API
Número de valores de atributos leídos	1000 por llamada de API
Profundidad de ruta	15
Tamaño de entrada máximo (todas las entradas combinadas)	200 KB
Tamaño máximo de respuesta (todos los elementos de salida combinados)	1 MB
Límite de tamaño de políticas	10 KB
Número de atributos que se pueden eliminar durante la eliminación de objetos	30

Límite/Concepto	Quantity (Cantidad)
Longitud del valor de agregación para atributos de identidad de enlaces con tipo	64 bytes con cifrado UTF-8
Longitud de nombre de vínculo o borde	64 bytes con cifrado UTF-8
Longitud de valor para atributos indexados	512 bytes con cifrado UTF-8
Longitud de valor para atributos no indexados	2 KB
Número de políticas adjuntas a un objeto	4

Límites de las operaciones por lotes

No hay ningún límite en cuanto al número de operaciones que puede llamar dentro de un lote. Para obtener más información, consulte Límites de las operaciones por lotes.

Límites que no pueden modificarse

Los límites de Amazon Cloud Directory que no se pueden modificar o aumentar son:

- Longitud de nombre de faceta
- Longitud de nombre de directorio
- Longitud de nombre de esquema
- Máximo de elementos de página
- Longitud de nombre de vínculo o borde
- Longitud de valor para atributos indexados

Recursos Cloud Directory

En las siguientes tablas se incluyen recursos relacionados que le resultarán útiles cuando trabaje con este servicio.

Introducción a Cloud Directory	Enlace
Seminario web Cloud Directory	https://www.youtube.com/watch?v=UANm3DC_IxE
Código Java de ejemplo de Cloud Directory	https://github.com/aws-samples/AmazonCloudDirectory-sample

Publicaciones de blog de Cloud Directory	Descripción
How to rapidly develop applications on Amazon Cloud Directory with Managed Schema	En esta publicación del blog, se explica cómo crear y desarrollar prototipos rápidamente en Cloud Directory con esquemas administrados. También incluye código Java de ejemplo.
How to Search More Efficiently in Amazon Cloud Directory	Esta entrada de blog explica cómo realizar búsquedas de forma más eficiente mediante la indexación basada en facetas. También incluye código Java de ejemplo.
How to Easily Apply Amazon Cloud Directory Schema Changes with In-Place Schema Upgrades	Esta entrada de blog explica cómo realizar una actualización de esquema in situ para cualquier Cloud Directory operativo (en ejecución). También incluye código Java de ejemplo.
Write and Read Multiple Objects in Amazon Cloud Directory by Using Batch Operations	Explica el uso de lecturas y escrituras por lotes. También incluye código Java de ejemplo.

Publicaciones de blog de Cloud Directory	Descripción
How to Use Batch References in Amazon Cloud Directory to References to New Objects in a Batch Request	Explica el uso de la referencia por lotes. También incluye código Java de ejemplo.
Cloud Directory — Support for Typed Links	Explica cómo crear y buscar relaciones entre las jerarquías en Cloud Directory utilizando enlaces con tipo. También incluye código Java de ejemplo.
New Cloud Directory API Makes It Easier to Query Data Along Multiple Dimensions	Explica cómo realizar consultas de datos en varias dimensiones con una sola llamada utilizando la API `ListObjectParentPaths` .
How to Create an Organizational Chart with Separate Hierarchies by Using Amazon Cloud Directory	Explica cómo crear un esquema y un directorio con código Java de ejemplo.
Amazon Cloud Directory – A Cloud-Native Directory for Hierarchical Data	Describe el lanzamiento de Cloud Directory como un nuevo servicio desde AWS.

Documentación Cloud Directory	Enlace
Guía para desarrolladores de Cloud Directory	https://docs.aws.amazon.com/clouddirectory/latest/developerguide/what_is_cloud_directory.html
Referencia a la API de Cloud Directory	https://docs.aws.amazon.com/clouddirectory/latest/APIReference/welcome.html
Límites Cloud Directory	https://docs.aws.amazon.com/clouddirectory/latest/developerguide/limits.html

Cloud Directory	Enlace
Información del producto Cloud Directory	https://aws.amazon.com/cloud-directory/
Precios Cloud Directory	https://aws.amazon.com/cloud-directory/pricing/

Historial de revisión

En la siguiente tabla se describen los cambios que se han realizado en la documentación desde la última versión deGuía para desarrolladores de Amazon Cloud Directory.

- Última actualización de la documentación: 21 de junio de 2018

update-history-change	update-history-description	update-history-date
Nuevo esquema administrado	Se ha añadido contenido para la opción del esquema administrado.	21 de junio de 2018
Se ha migrado contenido a esta guía	Se ha transferido todo el contenido existente en Cloud Directory	20 de junio de 2018
Actualizaciones de esquema in situ	Se ha añadido contenido para aplicar cambios de esquema en los directorios de Amazon Cloud Directory con actualizaciones de esquema in situ.	6 de diciembre de 2017
Indexación basada en facetas	Se ha añadido la sección de índice basado en facetas.	9 de agosto de 2017
Lotes	Se ha actualizado la información sobre lotes para Amazon Cloud Directory.	26 de julio de 2017
Conformidad	Se ha añadido información sobre la conformidad con HIPAA y PCI.	14 de julio de 2017
Enlaces con tipo	Se ha añadido nuevo contenido de enlaces con tipo para Amazon Cloud Directory.	31 de mayo de 2017

Lanzamiento del servicio Amazon Cloud Directory	Se presentó un nuevo tipo de directorio.	26 de enero de 2017

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el glosario de AWS en la referencia general de AWS.

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.