Guia do desenvolvedor

Amazon Cloud Directory

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

Amazon Cloud Directory: Guia do desenvolvedor

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

As marcas comerciais e o visual comercial da Amazon não podem ser usados em conexão com nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa causar confusão entre os clientes ou que deprecie ou desacredite a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, conectados ou patrocinados pela Amazon.

Table of Contents

O que é Amazon Cloud Directory? ... 1
O que o Cloud Directory não é ... 2
Conceitos básicos ... 3
Criar um esquema ... 3
Criar um diretório do ... 4
Usar VPC endpoints de interface do Cloud Directory ... 5
Availability ... 6
Criar uma VPC Directory para o Cloud Directory ... 7
Conceitos chave do Cloud Directory ... 9
Schema ... 9
Facets ... 9
Esquemas gerenciados ... 9
Exemplos de esquemas ... 9
Esquemas personalizados ... 10
Directory ... 10
Objects ... 10
Policies ... 11
Estrutura de diretório ... 12
Nó raiz ... 13
Node ... 13
Nó folha ... 13
Link de nó ... 13
Schemas ... 14
Ciclo de vida do esquema ... 15
Estado de desenvolvimento ... 16
Estado publicado ... 16
Estado aplicado ... 16
Facets ... 17
Atualização de esquema no local ... 17
Versionamento do esquema ... 18
Usando as operações de API de atualização de esquema ... 19
Esquema gerenciado ... 20
Estilos de faceta ... 21
Exemplos de esquemas ... 22

Organizations ... 22
Person ... 24
Device ... 28
Esquemas personalizados ... 29
Referências a atributos ... 30
Exemplo do API ... 30
Exemplo de JSON: ... 31
Regras para atributos ... 34
Especificação do formato ... 35
Formato do esquema JSON ... 36
Exemplos de documentos de esquema ... 38
Objetos do diretório ... 44
Links ... 44
Links filho ... 45
Links de anexo ... 45
Links de índice ... 45
Links tipados ... 46
Filtros de intervalo ... 52
Várias limitações de intervalo ... 53
Valores ausentes ... 55
Acessar objetos ... 55
Preencher objetos ... 56
Atualização de objetos ... 56
Excluir objetos ... 56
Consultar objetos ... 57
Níveis de consistência ... 60
Níveis de isolamento de leitura ... 60
Solicitações de gravação ... 61
RetryableConflictExceptions ... 61
Indexação e pesquisa ... 63
Ciclo de vida do índice ... 63
Indexação baseada em facetas ... 64
Índices exclusivos vs. não exclusivos ... 66
Instruções de uso... ... 67
Gerenciar diretórios do ... 67
Crie seu diretório ... 67

Excluir seu diretório ... 68
Desativar o diretório do ... 69
Habilitar seu diretório ... 69
Gerenciar seu esquema ... 70
Criar seu esquema ... 70
Excluir um esquema ... 71
Fazer download de um esquema ... 72
Publicar um esquema ... 72
Atualize seu esquema ... 72
Atualize o esquema ... 73
Segurança ... 74
Identity and Access Management ... 74
Authentication ... 75
Controle de acesso ... 77
Visão geral do gerenciamento de acesso ... 77
Uso de políticas baseadas em identidade (políticas do IAM) ... 82
Referência de permissões da API Amazon Cloud Directory ... 84
Registro em log e monitoramento ... 84
Validação de conformidade ... 84
Resiliência ... 85
Segurança da infraestrutura ... 86
Suporte de transação ... 87
BatchWrite ... 87
Nome da referência de lote ... 88
BatchRead ... 89
Limites de operações em lote ... 89
Tratamento de exceções ... 91
Falhas de operações de gravação em lotes ... 91
Falhas de operações de leitura em lotes ... 91
Conformidade ... 92
Responsabilidade compartilhada ... 93
Usando as APIs do Cloud Directory ... 95
Como funciona o faturamento com as APIs do Cloud Directory ... 95
Limites ... 102
Amazon Cloud Directory ... 102
Limites de operações em lote ... 104

Limites que não podem ser modificados 104
Directory Recursos da nuvem 105
Histórico do documento 108
Glossário da AWS 110
..... cxi

O que é Amazon Cloud Directory?

O Amazon Cloud Directory é um armazenamento multilocatário altamente disponível baseado em diretório da AWS. Esses diretórios dimensionam automaticamente para centenas de milhões de objetos conforme a necessidade dos aplicativos. Isso permite que a equipe de operações se foque no desenvolvimento e na implantação de aplicativos que orientem os negócios e não no gerenciamento da infraestrutura de diretório. Ao contrário de sistemas tradicionais de diretório, o Cloud Directory não limita os objetos organizacionais do diretório a uma única hierarquia fixa.

Com o Cloud Directory, você pode organizar objetos do diretório em várias hierarquias para dar suporte a muitos pivôs e relacionamentos organizacionais através das informações do diretório. Por exemplo, um diretório de usuários pode oferecer uma visualização hierárquica com base na estrutura, no local e na afiliação de um projeto de relatórios. Da maneira semelhante, um diretório de dispositivos pode ter várias visualizações hierárquicas com base no fabricante, no proprietário atual e no local físico.

Em seu cerne, o Cloud Directory é um armazenamento de diretório baseado em gráfico especializado que fornece um bloco de criação fundacional a desenvolvedores. Com o Cloud Directory, os desenvolvedores podem fazer o seguinte:

- Criar facilmente aplicativos baseados em diretório, sem ter que se preocupar com implantação, escala global, disponibilidade e desempenho
- Criar aplicativos que ofereçam gerenciamento de usuários e grupos, gerenciamento de permissões ou políticas, registro de dispositivo, gerenciamento de clientes, catálogos de endereço, aplicativos ou produtos
- Definir novos objetos do diretório ou estender tipos existentes para atender às necessidades dos aplicativos, reduzindo o volume de código que precisa ser escrito
- Reduzir a complexidade da distribuição em camadas de aplicativos com o Cloud Directory
- Gerenciar a evolução das informações do esquema ao longo do tempo, assegurando a compatibilidade futura para clientes

O Cloud Directory inclui um conjunto de operações da API para acessar vários objetos e políticas armazenados nos diretórios baseados no Cloud Directory. Para obter uma lista das operações disponíveis, consulteAções da Amazon Cloud Directory. Para obter uma lista de operações e permissões necessárias para executar cada ação da API, consulte Permissões da Amazon Cloud Directory: Referência de ações, recursos e condições.

Para obter uma lista das regiões da compatíveis do Cloud Directory, consulte oRegiões e endpoints da AWSdocumentação. Para obter recursos adicionais, consulte Directory Recursos da nuvem.

O que o Cloud Directory não é

O Cloud Directory não é um serviço de diretório para administradores de TI que desejem gerenciar ou migrar sua infraestrutura de diretório.

Conceitos básicos

Neste exercício de conceitos básicos, você cria um 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. Embora não seja necessário, é recomendável examinar Entendendo os principais conceitos do Cloud Directory antes de começar a usar o console para se familiarizar com os principais recursos e terminologia.

Tópicos

- Criar um esquema
- Criar um Amazon Cloud Directory
- Usar VPC endpoints de interface do Cloud Directory

Criar um esquema

O Amazon Cloud Directory oferece suporte ao carregamento de arquivos JSON que sejam compatíveis para a criação de esquemas. Para criar um esquema novo, você pode criar seu próprio arquivo JSON do zero ou fazer download de um dos esquemas existentes listados no console. Em seguida, faça upload dele como esquema personalizado. Para obter mais informações, consulte Esquemas personalizados.

Você também pode criar, excluir, baixar, listar, publicar, atualizar e atualizar esquemas usando as APIs do Cloud Directory. Para obter mais informações sobre as operações da API do esquema, consulte oAmazon Cloud Directory Guide.

Escolha qualquer pessoa procedimentos abaixo, dependendo de seu método preferido.

Para criar um esquema personalizado

1. NoAWS Directory Servicepainel de navegação, emDiretório na nuvem, escolhaSchemas.
2. Crie um arquivo JSON com todas as definições novas do esquema. Para obter mais informações sobre como formatar um arquivo JSON, consulte Formato do esquema JSON.
3. No console, escolhaUpload new esquema.
4. NoUpload new esquema, digite um nome para o esquema.
5. SelectEscolher arquivo, selecione o novo arquivo JSON que você acabou de criar e escolhaAberto.

6. Escolha Upload (Fazer upload). Isso adiciona um novo esquema à sua biblioteca de esquemas e o coloca noDesenvolvimentoEstado. Para obter mais informações sobre estados de esquema, consulte Ciclo de vida do esquema.

Para criar um esquema personalizado com base em um existente no console

1. NoAWS Directory Servicepainel de navegação, emDiretório na nuvem, escolhaSchemas.
2. Na lista de lista os esquemas, selecione a opção perto do esquema que você deseja copiar.
3. Escolha Actions.
4. SelecioneDownload esquema.
5. Renomeie o arquivo JSON, edite-o conforme necessário e salve o arquivo. Para obter mais informações sobre como formatar um arquivo JSON, consulte Formato do esquema JSON.
6. No console, escolhaUpload new esquema, selecione o arquivo JSON que você acabou de editar e escolhaAberto.

   Isso adiciona um novo esquema à sua biblioteca de esquemas e o coloca noDesenvolvimentoEstado. Para obter mais informações sobre estados de esquema, consulte Ciclo de vida do esquema.

Criar um Amazon Cloud Directory

Antes de criar um diretório no Amazon Cloud Directory, o AWS Directory Service requer que você aplique primeiro um esquema a ele. Um diretório não pode ser criado sem um esquema e, normalmente, tem um esquema aplicado a ele. Contudo, você usa as Operações da API do Cloud Directory para aplicar esquemas adicionais a um diretório. Para obter mais informações, consulteApplySchemanoGuia de referência de API do Amazon Cloud Directory.

Para criar um Cloud Directory

1. NoAWS Directory Servicepainel de navegação, emDiretório na nuvem, escolhaDiretórios.
2. SelecioneConfigurar o Cloud Directory.
3. UnderEscolha um esquema para aplicar ao seu novo diretório, digite o nome amigável do diretório, como`User Repositorye`, em seguida, escolha uma das seguintes opções:

   - Esquema gerenciado
   - Exemplos de esquema

- Esquema personalizado

Esquemas de amostra e esquemas personalizados são colocados no arquivoDesenvolvimentoestado, por padrão. Para obter mais informações sobre estados de esquema, consulte Ciclo de vida do esquema. Antes que um esquema seja aplicado a um diretório, ele deve ser convertido ao estado Published. Para publicar com êxito um esquema de exemplo usando o console, você deve ter permissões para as ações a seguir:

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

Visto que os esquemas de exemplo são modelos somente leitura fornecidos pela AWS, eles não podem ser publicados diretamente. Em vez disso, quando você optar por criar um diretório com base em um esquema de exemplo, o console criará uma cópia temporária do esquema de exemplo que você tiver selecionado e a colocará na caixaDesenvolvimentoEstado. Em seguida, ele criará uma cópia daquele esquema de desenvolvimento e a colocará no estado Published. Depois de publicado, o esquema de desenvolvimento será excluído, o que explica o motivo da ação `DeleteSchema` ser necessária quando se publica um esquema de exemplo.

4. Escolha Next (Próximo).
5. Revise as informações do diretório e faça as alterações necessárias. Quando as informações estiverem corretas, selecione Create (Criar).

Usar VPC endpoints de interface do Cloud Directory

Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus recursos da AWS, pode estabelecer uma conexão privada entre a VPC e o Cloud Directory. Você pode usar essa conexão para habilitar o Cloud Directory a se comunicar com os seus recursos na VPC sem passar pela Internet pública.

A Amazon VPC é um serviço da AWS que você pode usar para executar os recursos da AWS em uma rede virtual definida por você. Com a VPC, você tem controle sobre as configurações de rede, como o intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Para conectar a VPC ao Cloud Directory, você define umaVPC endpoint de interfacepara o Cloud Directory. O endpoint fornece uma conectividade confiável e escalável ao Cloud Directory sem a necessidade de um gateway da Internet, de uma instância de conversão de endereço de rede (NAT) ou de uma conexão VPN. Para obter mais informações, consulteO que é Amazon VPC?noGuia do usuário do Amazon VPC Guia.

Os VPC endpoints de interface são desenvolvidos pelo AWS PrivateLink, uma tecnologia da AWS que permite comunicação privada entre os serviços da AWS por meio de uma elastic network interface com endereços IP privados. Para obter mais informações, consulteAWS PrivateLink para Serviços da AWS.

As etapas a seguir são para usuários da Amazon VPC. Para obter mais informações, consulteConceitos básicos da Amazon VPCnoGuia do usuário do Amazon VPC Guia.

Availability

No momento, o Cloud Directory oferece suporte a VPC endpoints nas seguintes regiões:

- 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 (Oeste dos EUA)

Criar uma VPC Directory para o Cloud Directory

Para começar a usar o Cloud Directory com a VPC, use o console da Amazon VPC para criar um VPC endpoint de interface para o Cloud Directory. Para obter mais informações, consulte Criação de um endpoint de interface.

- para oCategoria de serviço, escolhaServiços da AWS.
- Em Service Name (Nome do serviço), escolha **`com.amazonaws.region.clouddirectory`**. Isso cria um endpoint da VPC para operações do Cloud Directory.

Para obter informações gerais, consulteO que é Amazon VPC?noGuia do usuário do Amazon VPC Guia.

Controle o acesso ao seu endpoint da VPC do Cloud Directory

Uma política de VPC endpoint é uma política de recursos do IAM que você anexa a um endpoint quando cria ou modifica o endpoint. Se você não anexar uma política quando criar um endpoint, anexaremos uma política padrão que permita o acesso total ao serviço. Uma política de endpoint não substitui políticas de usuário do IAM ou políticas de serviço específicas. É uma política separada para controlar o acesso do endpoint ao serviço especificado.

Políticas de endpoint devem ser gravadas em formato JSON. Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Veja a seguir um exemplo de uma política de endpoint para o Cloud Directory. Esta política permite que os usuários se conectem ao Cloud Directory por meio da VPC para listar diretórios e impede que outras ações do Cloud Directory sejam executadas.

```
{
  "Statement": [
    {
      "Sid": "ReadOnly",
      "Principal": "*",
      "Action": [
        "clouddirectory:ListDirectories"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

Para modificar a política de VPC endpoint para o Cloud Directory

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, selecione Endpoints.
3. Se você ainda não tiver criado o endpoint para o Cloud Directory, selecioneCriar endpoint. Em seguida, selecione**`com.amazonaws.region.clouddirectory`**e escolhaCriar endpoint.
4. Selecione o**`com.amazonaws.region.clouddirectory`**e escolha oPolíticana metade inferior da tela.
5. Selecione Edit policy (Editar política) e faça as alterações na política.

Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Entendendo os principais conceitos do Cloud Directory

Amazon Cloud Directory Directory é um armazenamento de dados com base em diretório que pode criar vários tipos de objetos em um modo orientado a esquema.

Tópicos

- Schema
- Directory
- Estrutura de diretório

Schema

Um esquema é uma coleção de facetas que define quais objetos podem ser criados em um diretório e como são organizados. Um esquema também aplica a integridade e a interoperabilidade dos dados. Um único esquema pode ser aplicado a mais de um diretório por vez. Para obter mais informações, consulte Schemas.

Facets

Uma faceta é uma coleção de atributos, limitações e links definidos em um esquema. Combinadas, as facetas definem os objetos em um diretório. Por exemplo, Person e Device podem ser facetas para definir funcionários corporativos com associação de vários dispositivos. Para obter mais informações, consulte Facets.

Esquemas gerenciados

Um esquema fornecido para facilitar o desenvolvimento e a manutenção rapidamente de seus aplicativos. Para obter mais informações, consulte Esquema gerenciado.

Exemplos de esquemas

O conjunto de esquemas de exemplo fornecido por padrão no console do AWS Directory Service. Por exemplo, Person, Organization e Device são todos esquemas de exemplo. Para obter mais informações, consulte Exemplos de esquemas.

Esquemas personalizados

Um ou mais esquemas definidos por um usuário que podem ser carregados na seção Esquemas ou durante o processo de criação do Cloud Directory do console do AWS Directory Service ou criado por chamadas da API.

Directory

Um diretório é um armazenamento de dados com base em esquema que contém tipos específicos de objetos organizados em uma estrutura multi-hierárquica (consulte Estrutura de diretório para obter mais detalhes). Por exemplo, um diretório de usuários pode oferecer uma visualização hierárquica com base na estrutura, no local e na afiliação de um projeto de relatórios. Da maneira semelhante, um diretório de dispositivos pode ter várias visualizações hierárquicas com base no fabricante, no proprietário atual e no local físico.

Um diretório define o limite lógico para o armazenamento de dados, completamente isolado de todos diretórios restantes no serviço. Também define os limites de uma solicitação individual. Uma transação ou consulta única é executada no contexto de um único diretório. Um diretório não pode ser criado sem um esquema e, normalmente, tem um esquema aplicado a ele. Contudo, você pode usar as operações da API do Cloud Directory para aplicar esquemas adicionais a um diretório. Para obter mais informações, consulte`ApplySchema`noGuia de referência da API do Amazon Cloud Directory.

Objects

Os objetos são uma entidade de dados estruturados em um diretório. Um objeto em um diretório tem o objetivo de capturar metadados (ou atributos) sobre uma entidade física ou lógica geralmente para fins de descoberta de informações e aplicação de políticas. Por exemplo, usuários, dispositivos, aplicativos, contas da AWS, instâncias do EC2 e buckets do Amazon S3 podem todos ser representados como diferentes tipos de objetos em um diretório.

A estrutura e o tipo de informações de um objeto são expressos como uma coleção de facetas. Você pode usar o `Path` ou o `ObjectIdentifier` para acessar objetos. Os objetos também podem ter atributos, que são uma unidade de metadados definida pelo usuário. Por exemplo, o objeto de usuário pode ter um atributo chamado email-address. Atributos sempre são associados a um objeto.

Policies

As políticas são um tipo especializado de objeto que são úteis para armazenar permissões ou recursos. As políticas oferecem a ação da API `LookupPolicy`. A ação da política de pesquisa usa uma referência a qualquer objeto como sua entrada inicial. Em seguida, ela percorre todo o diretório até a raiz. A ação coleta todos os objetos de política que encontra em cada caminho até a raiz. O Cloud Directory não interpreta nenhuma dessas políticas de nenhuma maneira. Em vez disso, os usuários do Cloud Directory interpretam políticas usando sua própria lógica de negócios especializada.

Por exemplo, imagine um sistema que armazene informações de funcionários. Os funcionários são agrupados em conjunto por função de cargo. Queremos estabelecer permissões diferentes para membros do grupo de recursos humanos e do grupo de contabilidade. Os membros do grupo de recursos humanos terão acesso às informações da folha de pagamento, e o grupo de contabilidade terá acesso às informações do razão. Para estabelecer essas permissões, anexamos objetos de política a cada um desses grupos. Na hora de avaliar as permissões de um usuário, podemos usar a ação da API `LookupPolicy` naquele objeto de usuário. O`LookupPolicy`A ação da API percorre a árvore do objeto da política especificada até a raiz. A ação para em cada nó e verifica se há alguma política anexada e a retorna.

Anexos de políticas

As políticas podem ser anexadas a outros objetos de duas maneiras: anexos pai-filho normais e anexos de políticas especiais. Usando anexos normais de pai-filho, uma política pode ser anexada a um nó pai. Isso é sempre útil para fornecer um mecanismo fácil para localizar políticas no diretório de dados. As políticas não podem ter filhos. As políticas anexadas via anexos pai-filho não serão retornadas durante chamadas da API `LookupPolicy`.

Os objetos de política também podem ser anexados a outros objetos por meio de anexos de política. Você pode gerenciar esses anexos de política usando as ações da API `AttachPolicy` e `DetachPolicy`. Os anexos de política permitem que os nós de política sejam localizados quando você usar a API LookupPolicy.

Especificação de esquema de política

Para começar a usar políticas, você deve primeiro adicionar uma faceta a seu esquema que ofereça suporte à criação de políticas. Para realizar isso, crie uma faceta configurando o `objectType` da faceta como POLICY. A criação de objetos usando uma faceta com o tipo POLICY garante que o objeto tenha recursos de política.

As facetas de políticas herdam dois atributos além de todos os atributos que você adiciona à definição:

- policy_type (string, obrigatório) – esse é um identificador que você pode fornecer para distinguir entre diferentes usos de políticas. Se suas políticas se encaixam logicamente em categorias claras, incentivamos configurar o tipo de atributo das políticas adequadamente. A API `LookupPolicy` retorna o tipo de política das políticas anexadas (consulte `PolicyAttachment`). Isso permite a filtragem fácil do tipo específico de política que você está procurando. Também permite usar policy_type para decidir como o documento deve ser processado ou interpretado.
- policy_document (Binário, obrigatório) – Você pode armazenar dados específicos do aplicativo nesse atributo, como concessões de permissão associadas à política. Se preferir, você também pode armazenar dados relacionados ao aplicativo em atributos normais em sua faceta.

Visão geral da API de política

Várias ações especializadas da API estão disponíveis para trabalhar com políticas. Para obter uma lista das operações disponíveis, consulteAções do Amazon Cloud Directory.

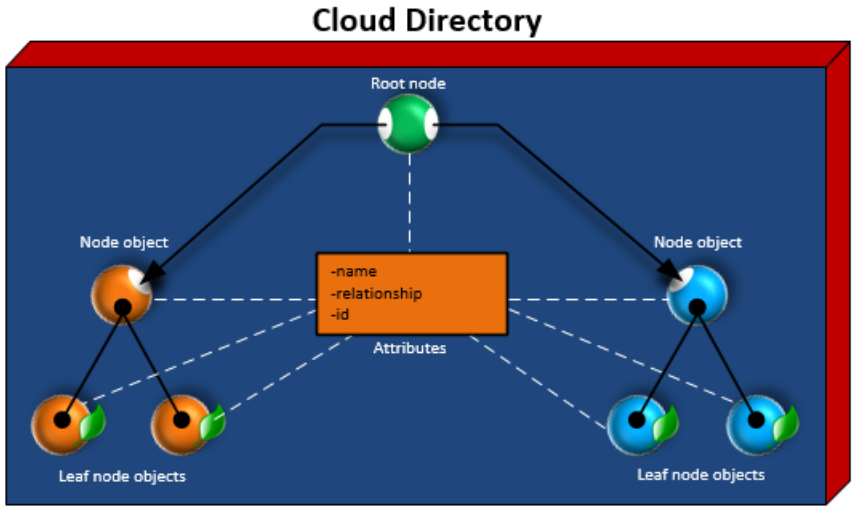
Para criar um objeto de política, use a ação da API `CreateObject` com uma faceta apropriada:

- Para anexar ou desanexar uma política de um objeto, use as ações `AttachPolicy` e `DetachPolicy` respectivamente.
- Para localizar políticas que estão anexadas a objetos na parte superior da árvore, use a ação da API `LookupPolicy`.
- Para listar as políticas que estão anexadas a um objeto específico, use a ação da API `ListObjectPolicies`.

Para obter uma lista de operações e permissões necessárias para executar cada ação da API, consulte Permissões da Amazon Cloud Directory: Referência de ações, recursos e condições.

Estrutura de diretório

Os dados em um diretório são estruturados hierarquicamente em um padrão de árvore que consiste em nós, nós folha e links entre os nós, conforme mostrado na ilustração a seguir. Isso é útil no desenvolvimento de aplicativos para modelar, armazenar e percorrer rapidamente dados hierárquicos.



Nó raiz

A raiz é o nó superior em um diretório que é usado para organizar os nós pai e filho na hierarquia. É semelhante à forma como as pastas em um sistema de arquivos podem conter subpastas e arquivos.

Node

Um nó representa um objeto que pode ter objetos filho. Por exemplo, um nó pode representar logicamente um grupo de gerentes por meio do qual vários objetos User são os filhos ou os nós folha. Um objeto de nó pode ter apenas um pai.

Nó folha

Um nó folha representa um objeto sem nenhum filho que pode ou não estar conectado de maneira diferente a um nó pai. Por exemplo, um objeto de usuário ou de dispositivo. Um objeto de nó folha pode ter vários pais. Embora os objetos de nó folha não precisem conectar-se a um nó pai, recomenda-se veementemente que você o conecte, pois sem um caminho da raiz, o objeto só pode ser acessado por seu `NodeId`. Se você perder o id desse objeto, você não terá como localizá-lo novamente.

Link de nó

A conexão entre um nó e outros. O Cloud Directory oferece suporte a vários tipos de link entre nós, incluindo links pai-filho, links de política e links de atributo de índice.

Schemas

No Amazon Cloud Directory, os esquemas definem que tipos de objetos podem ser criados em um diretório (usuários, dispositivos, e organizações), aplicam a validação de dados para cada classe de objeto e lidam com as alterações efetuadas nos esquemas ao longo do tempo. Mais especificamente, um esquema define o seguinte:

- Um ou mais tipos de facetas que podem ser mapeadas para os objetos em um diretório (como Person, Organization_Person)
- Atributos que podem ser mapeados para objetos em um diretório (como Name, Description). Os atributos podem ser obrigatórios ou opcionais em vários tipos de facetas, e são definidos no contexto de uma faceta.
- Restrições que podem ser aplicadas aos atributos dos objetos (como Required, Integer, String)

Quando um esquema é aplicado a um diretório, todos os dados dentro desse diretório devem passar a obedecer o esquema aplicado. Desta forma, a definição do esquema é essencialmente um diagrama que pode ser usado para criar vários diretórios com esquemas aplicados. Depois de criados, esses esquemas aplicados podem variar em relação ao diagrama original, cada um de uma maneira diferente.

Os esquemas aplicados podem ser atualizados posteriormente usando versionamento e, em seguida, reaplicados a todos os diretórios que os usam. Para obter mais informações, consulte Atualização de esquema no local.

O Cloud Directory fornece operações de API para criar, ler, atualizar e excluir esquemas. Isso permite que o conteúdo do esquema seja consumido facilmente por agentes de programas. Tais agentes acessam o diretório para localizar o conjunto completo de facetas, atributos e restrições que se aplicam aos dados no diretório. Para obter mais informações sobre as APIs de esquemas, consulte oGuia de referência de API do Amazon Cloud Directory.

O Cloud Directory oferece suporte ao carregamento de arquivos JSON que sejam compatíveis para a criação de esquemas. Você também pode criar e gerenciar esquemas usando o console do AWS Directory Services. Para obter mais informações, consulte Criar um Amazon Cloud Directory.

Tópicos

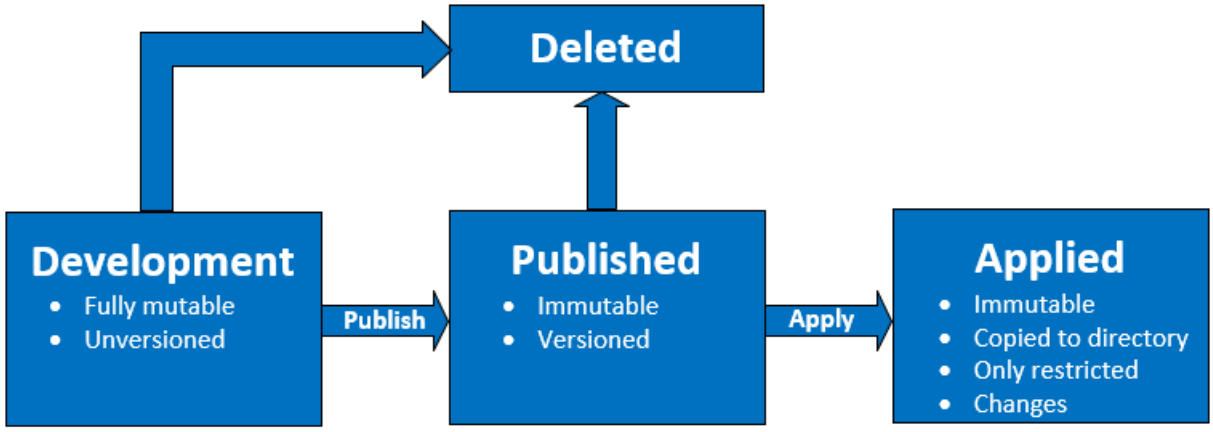
- Ciclo de vida do esquema

- Facets
- Atualização de esquema no local
- Esquema gerenciado
- Exemplos de esquemas
- Esquemas personalizados
- Referências a atributos
- Regras para atributos
- Especificação do formato

Ciclo de vida do esquema

O Cloud Directory oferece um ciclo de vida de esquemas para auxiliar no desenvolvimento de esquemas. Este ciclo de vida consiste em três estados: Desenvolvimento, publicado e aplicado. Esses estados são projetados para facilitar a criação e a distribuição dos esquemas. Cada um dos estados tem características diferentes que contribuem com esse esforço.

O diagrama a seguir descreve as possíveis transições e suas explicações. Todas as transições de esquema são copy-on-write. Por exemplo, a publicação de um esquema de desenvolvimento não altera ou remove o esquema de desenvolvimento.



Você pode excluir um esquema quando ele está no estado de publicado ou de desenvolvimento. A exclusão de um esquema não pode ser desfeita nem ele pode ser restaurado após ter sido excluído.

Esquemas em estados de desenvolvimento, de publicado e de aplicado têm ARNs que os representam. Esses ARNs são usados nas operações de API para descrever o esquema com o qual a API está operando. É fácil distinguir o estado de um esquema observando o ARN do esquema.

- Desenvolvimento: `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 desenvolvimento

Os esquemas são criados inicialmente no estado de desenvolvimento. Os esquemas nesse estado são totalmente mutáveis. Você pode adicionar ou remover livremente facetas e atributos. A maior parte do design de um esquema ocorre nesse estado. Os esquemas nesse estado têm um nome, mas não têm uma versão.

Estado publicado

O estado publicado armazena os esquemas que estão prontos para serem aplicados aos diretórios de dados. Esquemas são publicados a partir do estado de desenvolvimento para o estado publicado. Não é possível alterar esquemas no estado publicado. Os esquemas publicados podem ser aplicados a inúmeros diretórios de dados.

É necessário associar uma versão aos esquemas publicados e aplicados. Para obter mais informações sobre versões, consulte Versionamento do esquema.

Estado aplicado

Um esquema publicado pode ser aplicado a diretórios de dados. Um esquema que é aplicado a um diretório de dados é chamado de esquema aplicado. Após aplicar um esquema a um diretório de dados, você pode usar as facetas do esquema para criar objetos. É possível aplicar vários esquemas ao mesmo diretório de dados. Um esquema aplicado só pode ser alterado da seguinte forma.

- Adição de uma faceta a um esquema aplicado
- Adição de um atributo não obrigatório a um esquema aplicado

Facets

As facetas são as abstrações mais básicas em um esquema. Elas representam um conjunto de atributos que podem ser associados a um objeto no diretório e são conceitualmente semelhantes a classes de objeto de LDAP. Cada objeto de diretório pode ter um determinado número de facetas associadas a ele. Para obter mais informações, consulte Entre os limites do Amazon Cloud Directory.

Cada faceta mantém seu conjunto próprio independente de atributos. Uma faceta consiste em metadados fundamentais, como o nome da faceta, informações da versão e comportamentos. A combinação de ARNs, facetas e atributos do esquema definem a singularidade do objeto.

O conjunto das facetas de um objeto, suas restrições, e os relacionamentos entre elas constituem uma definição de esquema abstrato. As facetas de um esquema são usadas para definir restrições sobre os seguintes elementos:

1. Atributos permitidos em um objeto
2. Tipos de políticas permitidos para aplicação em um objeto

Depois de adicionar as facetas necessárias ao seu esquema, você pode aplicar o esquema a seu diretório e criar os objetos aplicáveis. Por exemplo, você pode definir um esquema de dispositivo adicionando facetas, como computadores, telefones e tablets. Em seguida, você pode usar essas facetas para criar objetos de computador, de celular e de tablet no diretório em que o esquema se aplica.

Cloud Directory’s schema support makes it easy to add or modify facets and attributes without worrying about breaking applications. Para obter mais informações, consulte Atualização de esquema no local.

Atualização de esquema no local

O Cloud Directory oferece a atualização dos atributos do esquema existente atributos e as facetas que ajudam a integrar seus aplicativos com serviços fornecidos pela AWS. Os esquemas que não estão nos estados de publicado ou aplicado têm versões e não podem ser alterados. Para obter mais informações, consulte Ciclo de vida do esquema.

Versionamento do esquema

Uma versão do esquema indica um identificador exclusivo para um esquema que os desenvolvedores podem especificar ao programar seus aplicativos para estar em conformidade com determinadas regras e formatação de dados. Dois principais diferenciais na maneira como o versionamento funciona com o Cloud Directory são importantes para os desenvolvedores compreender. Esses diferenciadores – versão principal e versão secundária – podem determinar como futuras atualizações de esquema afetam o seu aplicativo.

Versão principal

Major version é o identificador de versão usado para controlar alterações na versão principal de um esquema. Ele pode ter até 10 caracteres de comprimento. As diferentes versões do mesmo esquema são completamente independentes. Por exemplo, dois esquemas com o mesmo nome e diferentes versões são tratados como esquemas completamente diferentes, que têm seus próprios namespaces.

Alterações incompatíveis com versões anteriores

Recomendamos fazer alterações à versão principal somente quando os esquemas são incompatíveis. Por exemplo, ao alterar o tipo de dados de um atributo existente (como a alteração de `string` para `integer`) ou retirar um atributo obrigatório do seu esquema. Alterações incompatíveis com versões anteriores exigem migração de dados de diretório a partir de uma versão anterior do esquema para a nova versão do esquema.

Versão secundária

Minor version é o identificador de versão usado para a atualização de esquemas no local ou quando você deseja fazer atualizações compatíveis com as versões anteriores, como a adição de atributos adicionais ou de facetas. Um esquema atualizado usando uma versão secundária pode ser aplicado no local em todos os diretórios que o utilizam sem quebrar nenhum aplicativo em execução. Isso inclui diretórios que são usados em ambientes de produção. Para obter um caso de uso de exemplo, consulte“Como aplicar com facilidade alterações de esquema de Amazon Cloud Directory com atualizações de esquemas no local”No blog de Cloud Directory.

As informações e o histórico da versão secundária são salvos juntamente com outras informações do esquema no repositório de metadados do esquema. Nenhuma informação de versão secundária é retida nos objetos. A vantagem de introdução de versão secundária é que o código do cliente funciona perfeitamente, desde que a versão principal não seja alterada.

Limites da Versão secundária

O Cloud Directory retém e, portanto, limita até cinco versões secundárias. No entanto, limites de versão secundária são aplicados de forma diferente para esquemas publicados e aplicados das seguintes maneiras:

- Esquemas aplicados: Depois que o limite de versão secundária for excedido, o Cloud Directory exclui automaticamente a versão secundária mais antiga.
- Esquemas publicados: Uma vez que o limite de versão secundária tenha sido excedido, o Cloud Directory não exclui nenhuma das versões secundárias, mas informa o usuário por meio de um`LimitExceededException`Que o limite foi excedido. Depois de exceder os limites de versão secundária, você pode excluir o esquema usando a opção`DeleteSchema`API ou solicitar um aumento de limite.

Usando as operações de API de atualização de esquema

Você pode usar a chamada de API `UpgradePublishedSchema` para atualizar esquemas publicados. Atualizações de esquema são aplicadas no local para os diretórios que contam com ele usando a chamada de API `UpgradeAppliedSchema`. Você também pode obter as versões principal e secundária de um esquema aplicado chamando o `GetAppliedSchemaVersion`s. Ou visualize os ARNs do esquema associado e o histórico de revisão do esquema para um diretório chamando`ListAppliedSchemaArns`. O Cloud Directory mantém as cinco versões mais recentes das alterações de esquema aplicadas.

Para obter um exemplo ilustrativo, consulte“Como aplicar com facilidade alterações de esquema de Amazon Cloud Directory com atualizações de esquemas no local”No blog de Cloud Directory. A postagem de blog demonstrará como executar uma atualização de esquema no local e usar versões de esquema no Cloud Directory. Ela abrange como adicionar atributos adicionais a uma faceta existente, adicionar uma nova faceta a um esquema, publicar o novo esquema e aplicá-lo a diretórios em execução para concluir a atualização de um esquema no local. Ele também mostra como visualizar o histórico de versão de um esquema do diretório, o que ajuda a garantir que a frota de diretórios está executando a mesma versão do esquema e tem o histórico correto de alterações de esquema aplicadas a ele.

Esquema gerenciado

O Cloud Directory facilita o rápido desenvolvimento de aplicativos usando um esquema gerenciado. Com um esquema gerenciado, você poderá criar um diretório e iniciar a criação e a recuperação de objetos dele em um ritmo mais rápido. Para obter mais informações, consulte Crie seu diretório.

Atualmente, existe um esquema gerenciado chamado de `QuickStartSchema`. Você pode criar um modelo de dados hierárquicos sofisticado e estabelecer relações entre objetos usando construções, como Links tipados. Em seguida, você pode consultar qualquer informação em seus dados percorrendo a hierarquia.

O esquema gerenciado `QuickStartSchema` é representado pelo seguinte JSON:

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

QuickStartSchema ARN (ARN do QuickStartSchema)

O esquema gerenciado `QuickStartSchema` usa a seguinte ARN:

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

Por exemplo, você pode usar o ARN para criar um diretório chamado de `ExampleDirectory`, como mostrado abaixo:

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

Estilos de faceta

Há dois estilos diferentes que podem ser definidos em qualquer faceta `Static` e `Dynamic`.

Facetas estáticas

Facetas estáticas são a melhor opção quando você tem todos os detalhes dos seus modelos de dados do diretório, como uma lista de atributos com os tipos de dados, e também deseja definir as restrições dos seus atributos, como campos obrigatórios ou únicos. O Cloud Directory aplicará as restrições de dados e a verificação de regras durante a criação ou alteração do objeto.

Facetas dinâmicas

Você pode usar uma faceta dinâmica quando precisar de flexibilidade para alterar o número de atributos ou alterar os valores dos dados armazenados nos atributos. O Cloud Directory não impõe restrições de dados e verificação de regras durante a criação ou alteração do objeto.

Após criar um esquema com facetas dinâmicas, você pode definir os atributos necessários ao criar objetos. O Cloud Directory aceitará os atributos pares de chave/valor e armazenará em seus objetos fornecidos.

Uma faceta dinâmica pode ser adicionada a um novo ou existente esquema. Você também pode combinar as facetas estática e dinâmica em um único esquema para obter benefícios para cada estilo de faceta em seu diretório.

Ao criar qualquer atributo usando a faceta dinâmica, eles são criados como tipo de dados `Variant`. Para armazenar valores para o atributo definido como`Variant`O tipo de dado, você pode usar valores de qualquer um dos tipos de dados primitivos compatíveis com o Cloud Directory, como o`String`ou`Binary`. Com o passar do tempo, o valor do atributo pode ser alterado para outro tipo de dados. Não há execução de validação de dados.

Você pode usar as facetas dinâmicas para definir objetos dos seguintes tipos:

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

Para obter mais detalhes sobre os esquemas gerenciados, facetas dinâmicas ou tipos de dados variantes e para ver exemplos de casos de uso, consulteHow to rapiddevelop applications on Amazon Cloud Directory using AWS Managed Schema (Como desenvolver deno blog do Amazon Cloud Directory.

Exemplos de esquemas

O Cloud Directory oferece exemplos de esquemas para Organizations, pessoas e dispositivos. A seção a seguir lista vários exemplos de esquemas, bem como as diferenças entre eles.

Organizations

As tabelas a seguir listam as facetas que estão incluídas no exemplo de esquema Organizações.

Faceta "Organization"	Tipo de dados	Length	Compo ento obrigat io?	Descrição
account_id	String	1024	N	Identificador exclusivo para a organizaç ão
account_name	String	1024	N	Nome da organização
organization_status	String	1024	N	Status como “ativo“, “suspenso“, “inativo“, “fechado”
mailing_address (street1)	String	1024	N	Um endereço postal físico para esta empresa/entidade
mailing_address (street2)	String	1024	N	Um endereço postal físico para esta empresa/entidade

Faceta "Organization"	Tipo de dados	Length	Componente obrigatório?	Descrição
mailing_address (city)	String	1024	N	Um endereço postal físico para esta empresa/entidade
mailing_address (state)	String	1024	N	Um endereço postal físico para esta empresa/entidade
mailing_address (country)	String	1024	N	Um endereço postal físico para esta empresa/entidade
mailing_address (postal_code)	String	1024	N	Um endereço postal físico para esta empresa/entidade
e-mail	String	1024	N	Identificador de e-mail para a organização
web_site	String	1024	N	URL do site
telephone_number	String	1024	N	Número de telefone da organização
descrição	String	1024	N	Descrição da organização

Faceta “Legal_Entity”	Tipo de dados	Length	Componente obrigatório?	Descrição
registered_company_name	String	1024	N	Nome da entidade legal ou pessoa jurídica.
mailing_address (street1)	String	1024	N	Um endereço físico registrado para esta empresa/entidade

Faceta “Legal_Entity”	Tipo de dados	Length	Comportamento obrigatório?	Descrição
mailing_address (street2)	String	1024	N	Um endereço físico registrado para esta empresa/entidade
mailing_address (city)	String	1024	N	Um endereço físico registrado para esta empresa/entidade
mailing_address (state)	String	1024	N	Um endereço físico registrado para esta empresa/entidade
mailing_address (country)	String	1024	N	Um endereço físico registrado para esta empresa/entidade
mailing_address (postal_code)	String	1024	N	Um endereço físico registrado para esta empresa/entidade
industry_vertical	String	1024	N	Segmento do setor
billing_currency	String	1024	N	Moeda do faturamento
tax_id	String	1024	N	Número de identificação fiscal

Person

As tabelas a seguir listam as facetas que estão incluídas no exemplo de esquema Pessoa.

Faceta “Person”	Tipo de dados	Length	Comportamento obrigatório?	Descrição
display_name	String	1024	N	O nome do usuário, para ser exibido aos usuários finais.

Faceta “Person”	Tipo de dados	Length	Comportamento obrigatório?	Descrição
first_name	String	1024	N	O nome do usuário, ou o primeiro nome, na maioria dos idiomas ocidentais
last_name	String	1024	N	O sobrenome do usuário, ou o último nome, na maioria dos idiomas ocidentais
middle_name	String	1024	N	O(s) nome(s) do meio do usuário
nickname	String	1024	N	Uma maneira casual de falar com o usuário na vida real, por exemplo, “Zé” ou “Zeca” em vez de “José”
e-mail	String	1024	N	Endereço de e-mail para o usuário
mobile_phone_number	String	1024	N	Número de telefone do usuário
home_phone_number	String	1024	N	Número de telefone do usuário
nome de usuário	String	1024	Y	Um identificador exclusivo para o usuário
profile	String	1024	N	Um URI, que é um localizador de recursos uniforme, e que aponta para um local que representa o perfil on-line do usuário (como uma página na web)

Faceta “Person”	Tipo de dados	Length	Comportamento obrigatório?	Descrição
picture	String	1024	N	Um URI, que é um localizador de recursos uniforme, que aponta para um local de recurso que representa a imagem do usuário.
site	String	1024	N	URL
timezone	String	1024	N	O fuso horário do usuário
locale	String	1024	N	Usado para indicar o local padrão do usuário para fins de adaptação de itens como moeda, formato de data e hora ou representações numéricas.
address (street1)	String	1024	N	Um endereço postal físico para este usuário.
address (street2)	String	1024	N	Um endereço postal físico para este usuário.
address (city)	String	1024	N	Um endereço postal físico para este usuário.
address (state)	String	1024	N	Um endereço postal físico para este usuário.
address (country)	String	1024	N	Um endereço postal físico para este usuário.
address (postal_code)	String	1024	N	Um endereço postal físico para este usuário.

Faceta “Person”	Tipo de dados	Length	Comportamento obrigatório?	Descrição
user_status	String	1024	N	Valor que indica o status administrativo do usuário

Faceta "Organization_Person"	Tipo de dados	Length	Comportamento obrigatório?	Descrição
title	String	1024	N	O cargo ou título na organização
preferred_language	String	1024	N	Indica os idiomas preferidos, escritos ou falados, do usuário, usados geralmente para selecionar uma interface de usuário traduzida.
employee_id	String	1024	N	Um identificador de string, normalmente numérico ou alfanumérico, atribuído a uma pessoa
cost_center	Inteiro	1024	N	Identifica o centro de custo
department	String	1024	N	Identifica o nome de um departamento
manager	String	1024	N	O gerente do usuário

Faceta "Organization_Person"	Tipo de dados	Length	Comportamento obrigatório?	Descrição
company_name	String	1024	N	Identifica o nome de uma organização
company_address (street1)	String	1024	N	Um endereço postal físico para a organização
company_address (street2)	String	1024	N	Um endereço postal físico para a organização
company_address (city)	String	1024	N	Um endereço postal físico para a organização
company_address (state)	String	1024	N	Um endereço postal físico para a organização
company_address (country)	String	1024	N	Um endereço postal físico para a organização
company_address (postalCode)	String	1024	N	Um endereço postal físico para a organização

Device

As tabelas a seguir listam as facetas que estão incluídas no exemplo de esquema Dispositivo.

Faceta "Device"	Tipo de dados	Length	Comportamento obrigatório?	Descrição
device_id	String	1024	N	Identificador alfanumérico exclusivo do dispositivo

Faceta "Device"	Tipo de dados	Length	Comportamento obrigatório?	Descrição
name	String	1024	N	Nome amigável para o dispositivo
descrição	String	1024	N	Descrição do dispositivo
X.509_certificates	String	1024	N	O certificado X.509
device_version	String	1024	N	A versão do dispositivo
device_os_type	String	1024	N	Sistema operacional do dispositivo
device_os_version	String	1024	N	Número da versão do sistema operacional no dispositivo
serial_number	String	1024	N	Número de série do dispositivo
device_status	String	1024	N	Status do dispositivo (como, por exemplo, ativo, inativo, suspenso, desligado, desativado)

Esquemas personalizados

O primeiro passo para criar um esquema personalizado é definir exatamente os campos que você deve indexar. Esses campos obrigatórios formam o esqueleto de seu esquema, ao qual você adicionará seus próprios campos. Mapeie o nome e o tipo de cada campo (como string, inteiro, booleano) para a estrutura do objeto. Você pode definir um esquema com tipos e restrições e depois aplicá-los a um diretório. Once defined, Cloud Directory performs validation for attributes.

Para obter mais informações, consulte Criar um esquema.

Referências a atributos

As facetas do Amazon Cloud Directory contêm atributos. Os atributos podem ser uma definição de atributo ou uma referência a atributo. Definições de atributos são atributos que declaram seu nome e tipo primitivo (string, binary, Boolean, DateTime ou number). Opcionalmente, eles também declaram seu comportamento necessário, valor padrão, sinalizador de imutável e regras de atributo (como tamanho mínimo e máximo).

Referências a atributos são os atributos que derivam seu tipo primitivo, valor padrão, sinalizador de imutável e regras de atributo da definição de outro atributo pré-existente. As referências a atributos não têm seus próprios tipos primitivos, valores padrão, sinalizador de imutável ou regras, pois essas propriedades são provenientes da definição do atributo de destino.

As referências a atributos podem substituir o comportamento necessário de uma definição de destino (mais detalhes sobre isso a seguir).

Quando você cria uma referência a atributo, você fornece somente um nome de atributo e a definição do atributo de destino (o que inclui o nome da faceta e o nome do atributo da definição do atributo de destino). As referências a atributos não podem fazer referência a outras referências a atributos. Além disso, no momento, as referências a atributos não podem ter como destino definições de atributos de outro esquema.

Você pode usar uma referência a atributo quando deseja dois ou mais atributos em um objeto para fazer referência ao mesmo local de armazenamento. Por exemplo, imagine um objeto que tem uma faceta User e uma faceta EnterpriseUser aplicadas. A faceta User tem uma definição de atributo FirstName, enquanto a faceta EnterpriseUser tem uma referência a atributo que aponta para User.FirstName. Como os dois atributos FirstName fazem referência ao mesmo local de armazenamento no objeto, qualquer alteração em User.FirstName ou em EnterpriseUser.FirstName tem o mesmo efeito.

Exemplo do API

O exemplo a seguir demonstra o uso de referências a atributos usando a API do Cloud Directory. Neste exemplo, uma faceta base contém uma definição de atributo, e outra faceta contém um atributo que faz referência a um atributo na faceta base. Observe que o atributo de referência pode ser marcado como Required enquanto a faceta é Not required.

```
// 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)
```

Exemplo de JSON:

O exemplo a seguir demonstra o uso de referências a atributos em um modelo JSON. O esquema representado pelo modelo é idêntico ao modelo acima.

```
 {
  "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"
    }
}
```

Considerações sobre referência a atributo

As referências a atributos devem ter como destino uma definição de atributo pré-existente no mesmo esquema.

- As referências a atributos podem ter como destino uma definição de atributo pré-existente na mesma faceta ou em outra faceta.
- As referências a atributos não podem ter como destino outras referências a atributos.
- As facetas que contêm definições a atributos que são o destino de referência a atributo de outra faceta não podem ser excluídas até que todas as referências tenham sido excluídas.

Você pode usar referências a atributos da mesma maneira como usa definições de atributos tradicionais, criando objetos ou aplicando facetas a objetos existentes.

> **Note**
>
> Você pode aplicar facetas com referências a outras facetas, mas não é necessário aplicar as facetas de destino diretamente. Quando a faceta de destino não é aplicada, não há nenhuma alteração no comportamento da referência a atributo. (É necessário aplicar facetas de destino somente quando você desejar que outros atributos dessa faceta existam no objeto.)

Definir valores de referência a atributo

Você pode chamar a ação da API UpdateObjectAttributesquando desejar alterar o valor de um atributo. Atualizar (ou excluir) a definição ou qualquer outra referência a essa mesma definição naquele objeto tem o mesmo efeito.

Obter os valores de referência a atributo

Você pode chamar a ação da API ListObjectAttributes para recuperar alias de armazenamento. Essa chamada retorna uma lista de tuplas, cada uma contendo uma chave de atributo e o valor associado. As chaves de atributos correspondem à lista de alias de armazenamento presente naquele objeto.

> **Note**
>
> É possível que uma chave de atributo seja retornada para uma faceta que não foi aplicada explicitamente a um objeto. Isso pode acontecer quando referências a atributos têm como destino facetas que não são aplicadas ao objeto.

Por exemplo, imagine um você tem uma faceta User e uma faceta EnterpriseUser. O atributo EnterpriseUser.FirstName faz referência a User.FirstName. Em seguida, você aplica as facetas User e EnterpriseUser a um objeto, define User.FirstName como Robert e, mais tarde, define EnterpriseUser.FirstName como Bob. Quando chamar ListObjectAttributes, você verá somente “User.FirstName = Bob” porque só existe um alias de armazenamento para os dois atributos FirstName.

Usar índices com referências a atributos

Você pode criar índices com uma definição de atributo somente, não com uma referência. A listagem de um índice não retorna chaves de atributos para referências a atributos. Mas retorna chaves de atributos para todas as definições de atributos que são destinadas por referências existentes no objeto indexado. Em outras palavras, na camada de índice, as referências a atributos são tratadas meramente como um identificador alternativo para um atributo, o que é resolvido para o identificador da definição correta do atributo em tempo de execução.

Por exemplo, imagine que você tem um índice para FirstName do atributo User da faceta. Você anexa um objeto com apenas a faceta EnterpriseUser aplicada. Em seguida, você define o valor do atributo EnterpriseUser.FirstName daquele objeto como Bob. Finalmente, você chama a ação ListIndex. Os resultados contêm apenas "User.FirstName = Bob".

Comportamento necessário para referências a atributos

As referências a atributos podem ter um comportamento necessário que é diferente da definição do atributo de destino. Isso permite que uma definição base seja opcional, enquanto uma referência à mesma definição pode ser necessária. Quando um objeto tem uma definição base e uma ou mais referências à mesma definição base, a definição base e todas as referências devem aderir ao comportamento mais forte necessário entre todos os atributos relacionados.

- Como com as definições de atributos, você deve fornecer valores para todas as definições de atributos necessárias quando cria o objeto ou quando adiciona uma faceta a um objeto existente.
- Como conveniência, quando mais de um atributo em um objeto fizer referência ao mesmo local de armazenamento, você precisa fornecer apenas um valor para um dos atributos para esse local de armazenamento.
- Da mesma forma, se você fornecer vários valores para o mesmo local de armazenamento, os valores deverão ser iguais.

Regras para atributos

As regras descrevem os valores permitidos para cada tipo de atributo e estabelecem restrições para os valores de um determinado atributo. Você deve especificar as regras como parte da definição de um atributo quando cria uma faceta. O Cloud Directory oferece suporte aos seguintes tipos de regras:

- Tamanho da string
- Tamanho do binário
- String do conjunto
- Comparação de números

Tamanho da string

Restringe o tamanho do valor de um atributo do tipo string.

Chaves permitidas para o parâmetro da regra: min, max

Valores permitidos para o parâmetro da regra: número

Tamanho do binário

Restringe o tamanho da matriz de bytes do valor de um atributo do tipo binário.

Chaves permitidas para o parâmetro da regra: min, max

Valores permitidos para o parâmetro da regra: número

String do conjunto

Restringe o valor de um atributo do tipo string para o conjunto de strings especificadas permitido.

Chaves permitidas para o parâmetro da regra: allowedValues

Valores permitidos para o parâmetro da regra: Conjunto de strings, com cada string codificada em UTF-8

Os valores permitidos são delimitados por vírgula e podem ser colocados entre aspas. Isso é útil quando os valores permitidos incluem vírgula. Por exemplo:

- Um,dois,três = corresponde a Um dois ou três
- “com,vírgula “,“semvírgula” = corresponde a “com,vírgula” ou “semvírgula”
- com”aspas,semaspas corresponde a ‘com”aspas’ ou ‘semaspas’

Comparação de números

Restringe o valor numérico permitido para um atributo numérico.

Chaves permitidas para o parâmetro da regra: min, max

Valores permitidos para o parâmetro da regra: número

Especificação do formato

Um esquema do Cloud Directory do fornece uma estrutura para os dados dos seus diretórios de dados. O Cloud Directory fornece dois mecanismos para você definir seu esquema. Os desenvolvedores podem usar operações de API específicas para criar um esquema ou podem fazer upload de um esquema completo usando os recursos de upload de esquemas. Os documentos de um esquema podem ser carregados por meio de chamadas de API ou do console. Esta seção descreve o formato a ser usado para fazer upload dos documentos de um esquema completo.

Formato do esquema JSON

Um documento de esquema é um documento JSON formatado da seguinte maneira.

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

Um documento de esquema contém um mapa que relaciona os nomes de facetas às facetas. Cada faceta, por sua vez, contém uma mapa que possui atributos. Os nomes de todas as facetas em um esquema devem ser exclusivos. Os nomes de todos os atributos em uma faceta devem ser exclusivos.

Subseção de atributos JSON

As facetas contêm atributos. Cada atributo define o tipo de valor que pode ser armazenado em um atributo. O formato JSON a seguir descreve um atributo.

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

Você deve fornecer uma definição de atributo ou uma referência de atributo. Para obter mais informações, consulte as subseções relacionadas.

O campo de comportamento obrigatório indica se o atributo é necessário ou não. Você deve fornecer este campo. Os valores possíveis são:

- `REQUIRED_ALWAYS`: Este atributo deve ser fornecido quando o objeto é criado ou uma faceta é adicionada ao objeto. Não é possível remover este atributo.
- `NOT_REQUIRED`: Este atributo pode ou não estar presente.

Subseção de definição de atributos

Um atributo define o tipo e as regras associadas a um valor de atributo. O seguinte layout JSON descreve o formato.

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

Subseção de valores padrão

Especifique exatamente um dos seguintes valores padrão. Os valores longos e boolianos devem ser fornecidos sem aspas (como seus tipos respectivos do Javascript, em vez de strings). Os valores binários são fornecidos por meio de uma string codificada em Base64 que pode ser usada como URL (como descrito no RFC 4648). Datas e horas são fornecidas em número de milissegundos, desde o epoch (00:00:00 UTC em 1º de janeiro 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
}
```

Subseção de regras de atributos

As regras de atributos definem restrições em valores de atributo. Você pode definir várias regras para cada atributo. As regras de atributo contêm um tipo de regra e um conjunto de parâmetros para a regra. Você pode encontrar mais detalhes na seção Regras para atributos.

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

```
    }
}
```

Subseção de referência de atributo

A referência de atributo é um recurso avançado. As referências de atributos permitem que várias facetas compartilhem uma definição de atributo e um valor armazenado. Consulte a seção Referências a atributos para obter mais informações. Você pode definir uma referência de atributo no esquema JSON com o seguinte modelo.

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

Exemplos de documentos de esquema

Veja a seguir exemplos de documentos de esquema que mostram a formatação válida para JSON.

> **Note**
>
> Todos os valores expressados na string `allowedValues` devem ser separados por vírgula e sem espaços. Por exemplo, `"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 com links digitados

```
{
    "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 do diretório

Os desenvolvedores modelam objetos de diretório usando esquemas extensíveis para impor limitações de exatidão de dados automaticamente, facilitando a programação deles. O Amazon Cloud Directory oferece consulta de informações sofisticadas com base em seus atributos indexados definidos, habilitando dessa forma travessias de árvore e pesquisas nas árvores de diretórios. Os dados do Cloud Directory são criptografados em repouso e em trânsito.

Um objeto é um elemento básico do Cloud Directory. Cada objeto tem um identificador exclusivo globalmente, que é especificado pelo identificador do objeto. Um objeto é uma coleção de zeros ou mais facetas com suas chaves e valores de atributos. Um objeto pode ser criado de uma ou mais facetas em um único esquema aplicado ou de facetas de vários esquemas aplicados. Durante a criação do objeto, você deve especificar todos os valores de atributos necessários. Os objetos podem ter um número limitado de facetas. Para obter mais informações, consulte Entre os limites do Amazon Cloud Directory.

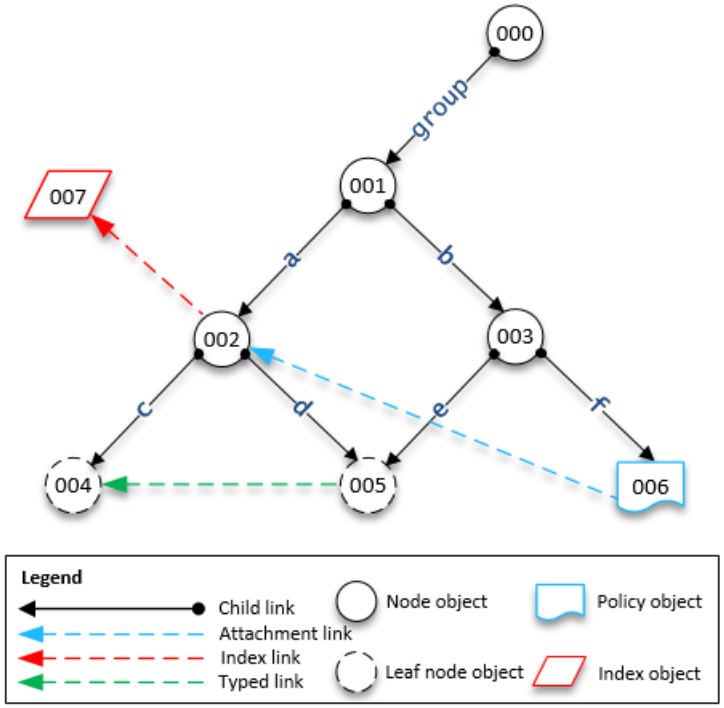
Um objeto pode ser um objeto regular, um objeto de política ou um objeto de índice. Um objeto também pode ser um objeto de nó ou um objeto de nó folha. O tipo do objeto é inferido a partir do tipo de objeto das facetas anexadas a ele.

Tópicos

- Links
- Filtros de intervalo
- Acessar objetos
- Níveis de consistência

Links

Um link é uma borda direcionada entre dois objetos que define uma relação. No momento, o Cloud Directory oferece suporte aos seguintes tipos de link.



Links filho

Um link filho cria uma relação pai-filho entre os objetos que conecta. Por exemplo, na ilustração acima, o link filho b conecta os objetos `001` e `003`. Os links filho definem a hierarquia no Cloud Directory. Os links filho têm nomes quando participam da definição do caminho do objeto para o qual o link aponta.

Links de anexo

Um link de anexo aplica um objeto de política de nó folha a outro nó folha ou a um objeto de nó. Os links de anexo não definem a estrutura hierárquica do Cloud Directory. Por exemplo, na ilustração acima, o link de anexo aplica a política armazenada no objeto do nó folha da política `006` no objeto do nó `002`. Cada objeto pode ter várias políticas anexadas, mas não pode ser anexada mais que uma política de qualquer determinado tipo de política.

Links de índice

Os links de índice fornecem pesquisa de informações sofisticadas com base em um objeto de índice e seus atributos indexados definidos, habilitando dessa forma travessias de árvore e pesquisas nas árvores de diretórios. Conceitualmente, os índices são semelhantes aos nós com filhos: Os links para os nós indexados são rotulados de acordo com os atributos indexados, em vez de receberem um

rótulo quando o filho está anexado. No entanto, os links de índices não são bordas pai-filho e têm seu próprio conjunto de operações da API de enumeração. Para obter mais informações, consulte Indexação e pesquisa.

Links tipados

Os links tipados permitem que você estabeleça uma relação entre os objetos ou nas hierarquias do Cloud Directory. Em seguida, você pode usar esses relacionamentos para consultar informações, como quais usuários têm o dispositivo "xyz" ou quais dispositivos são de propriedade do usuário "abc".

Você pode usar links tipados para modelar relacionamentos entre diferentes objetos no diretório. Por exemplo, na ilustração acima, considere a relação entre o objeto `004`, que representa um usuário, e o objeto `005`, que representa um dispositivo.

Podemos usar um link tipado para modelar uma relação de propriedade entre os dois objetos. Podemos adicionar atributos ao link tipado para representar o custo de uma compra, seja o dispositivo alugado ou comprado. Há dois tipos de atributos associados a links tipados:

- Atributos baseados em identidade – um atributo de um link tipado que o diferencia de outros links (por exemplo, links filho, de anexo ou de índice). Cada faceta do link tipado define um conjunto ordenado de atributos de identidade. A identidade de um link tipado é o ID do objeto de origem, um identificador de faceta (tipo), os valores de seus atributos de identidade (definidos pela faceta) e o ID do objeto de destino. Os identificadores devem ser exclusivos em um único diretório.
- Atributos opcionais – um atributo que armazena características de rastreamento sobre o link tipado que não são relacionadas à identidade do link. Por exemplo, um atributo opcional pode identificar a data em que o link tipado foi estabelecido pela primeira vez ou quando ele foi modificado pela última vez.

Assim como com objetos, você deve criar uma faceta de link tipado usando a API `CreateTypedLinkFacet` para definir a estrutura do link tipado e seus atributos. As facetas de links tipados exigem um nome de faceta exclusivo e um conjunto de atributos associados ao link. Ao criar a estrutura do link tipado, você pode definir um conjunto ordenado de atributos na faceta do link tipado. Para visualizar um esquema de exemplo de links tipados, consulte Documento de esquema com links digitados.

Os atributos de links tipados podem ser usados quando você precisa executar qualquer uma das seguintes ações:

- Permitir a filtragem de links tipados de entrada e de saída. Para obter mais informações, consulte Listagem de links tipados.
- Representar a relação entre dois objetos.
- Rastrear dados administrativos sobre o link tipado, como a data em que o link foi criado.

Considere o seguinte ao decidir se os links tipados são os corretos para seu caso de uso:

- Os links tipados não podem ser usados na especificação de objeto com base em caminho. Em vez disso, você deve selecionar links tipados usando as operações da API `ListOutgoingTypedLinks` ou `ListIncomingTypedLinks`.
- Os links tipados não participam de operações da API `LookupPolicy` ou `ListObjectParentPaths`.
- Links tipados entres os dois mesmos objetos e na mesma direção não podem ter os mesmos valores de atributos. Isso pode ajudar a evitar links tipados duplicados entre os mesmos objetos.
- Os atributos adicionais podem ser usados quando você deseja adicionar informações opcionais.
- O tamanho combinado de todos os valores de atributos de identidade é limitado a 64 bytes. Para obter mais informações, consulte Entre os limites do Amazon Cloud Directory.

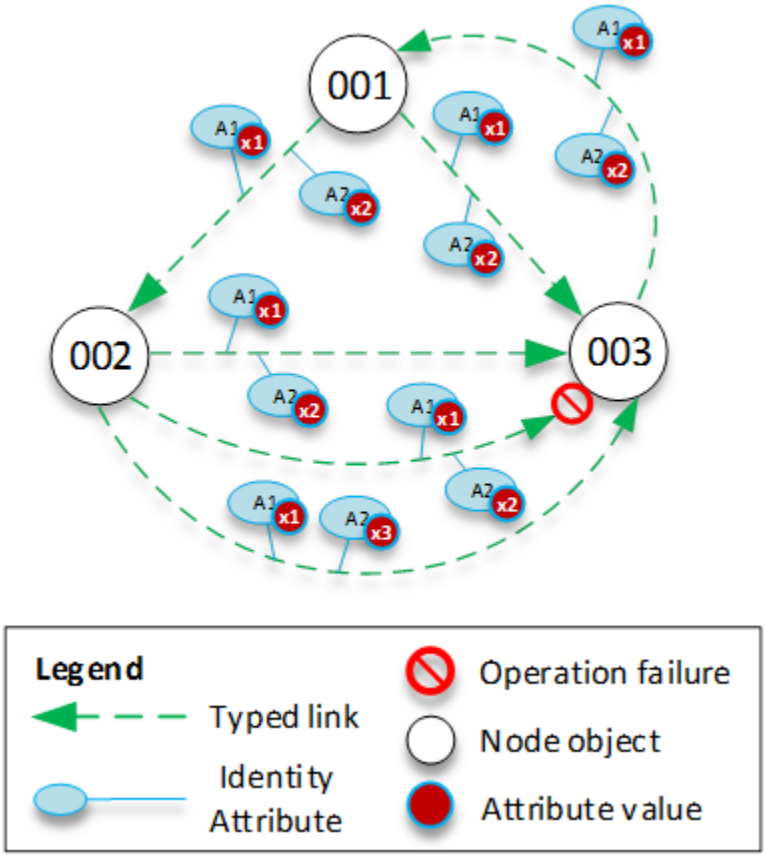
Artigo do blog Cloud Directory relacionado

- Usar links tipados Amazon Cloud Directory para criar e pesquisar relacionamentos em hierarquias

Identidade de link tipado

A identidade é o que define exclusivamente se um link tipado pode existir entre dois objetos. A exceção é quando você conecta dois objetos em uma direção com exatamente os mesmos valores de atributos. Os atributos devem ser configurados como `REQUIRED_ALWAYS`.

Links tipados que são criados de diferentes facetas de links tipados nunca entram em conflito um com o outro. Por exemplo, considere o seguinte diagrama:



- O objeto `001` tem links tipados e atributos (A1 e A2) com os mesmos valores de atributos (x1 e x2) indo para objetos diferentes (`002` e `003`). Essa operação teria êxito.
- Os objetos `002` e `003` têm um link tipado entre eles. Essa operação falharia porque dois links tipados na mesma direção com os mesmos atributos não podem existir entre objetos.
- Os objetos `001` e `003` têm dois links tipados entre eles com os mesmos atributos. Contudo, como os links vão para direções diferentes, essa operação teria êxito.
- Os objetos `002` e `003` têm links tipados entre eles com o mesmo valor para A1, mas com valores diferentes para A2. A identidade do link tipado considera todos os atributos, portanto, essa operação teria êxito.

Regras de links tipados

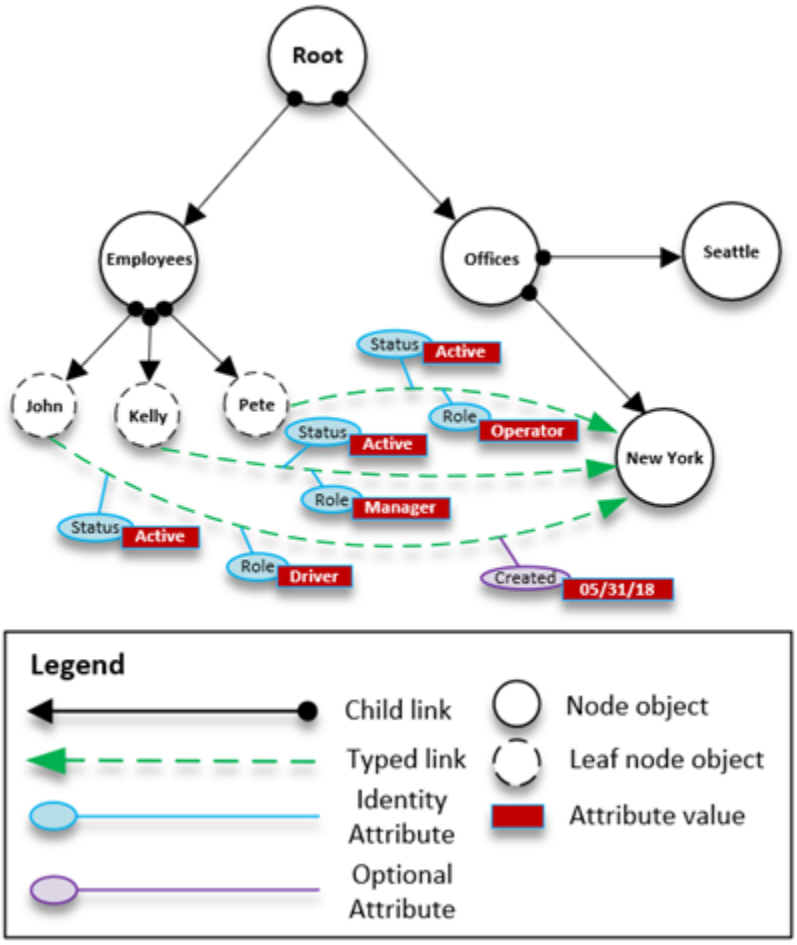
Você pode adicionar regras a atributos de link tipado quando desejar adicionar restrições aos atributos do link. Essas regras são equivalentes às regras em atributos de objeto. Para obter mais informações, consulte Regras para atributos.

Listagem de links tipados

O Cloud Directory fornece operações da API que você pode usar para selecionar links tipados de entrada ou de saída em um objeto. Você pode selecionar um subconjunto específico de links tipados em vez de iterar sobre cada link tipado. Você também pode especificar uma faceta de link tipado específica para filtrar somente os links tipados desse tipo.

Você pode filtrar links tipados na ordem que os atributos estão definidos na faceta do link tipado. Você pode fornecer filtros de intervalo para vários atributos. Ao fornecer intervalos a uma seleção de links tipados, todos os intervalos inexatos devem ser especificados no final. Todos os atributos sem intervalo especificado são supostos corresponderem ao intervalo inteiro. Os filtros são interpretados na ordem dos atributos definidos na faceta do link tipado, não na ordem em que são fornecidos a qualquer chamada da API.

Por exemplo, no diagrama a seguir, considere um Cloud Directory que é usado para armazenar informações sobre funcionários e suas capacidades.



Vamos supor que modelamos as capacidades de nossos funcionários com um link tipado chamado `EmployeeCapability`, que é configurado com três atributos de string: `Status`, `Role` e `Created`. Os seguintes filtros são compatíveis nas operações da API `ListIncomingTypedLinks` e `ListOutgoingTypedLinks`.

- Faceta = `EmployeeCapability`, status = `Active`, função = `Driver`
  - Seleciona funcionários ativos que são motoristas. Esse filtro inclui duas correspondências exatas.
- Faceta = `EmployeeCapability`, status = `Active`, função = `Driver`, criação = `05/31/18`
  - Seleciona os funcionários ativos que são motoristas e cujas facetas foram criadas em ou depois de 31 de maio de 2018.
- Faceta = `EmployeeCapability`, status = `Active`
  - Seleciona todos os funcionários ativos.
- Faceta = `EmployeeCapability`, status = `Active`, função = `A` a `M`
  - Seleciona funcionários ativos com funções que começam com `A` a `M`.
- Faceta = `EmployeeCapability`
  - Isso seleciona todos os links tipados do tipo `EmployeeCapability`.

Os seguintes filtros NÃO seriam compatíveis:

- Faceta = `EmployeeCapability`, status entre `A` a `C`, função = `Driver`
  - Esse filtro não é permitido porque todos os intervalos devem aparecer no final do filtro.
- Faceta = `EmployeeCapability`, função = `Driver`
  - Esse filtro não é permitido porque o intervalo de status implícito não é uma correspondência exata e não aparece no final da lista de intervalos.
- Status = `Active`
  - Esse filtro não é permitido porque a faceta do link tipado não está especificada.

Esquema de link tipado

Você pode criar facetas de link tipado de duas maneiras. Você pode gerenciar as facetas de links tipados em chamadas individuais da API, incluindo `CreateTypedLinkFacetDeleteTypedLinkFacet` e `UpdateTypedLinkFacet`. Você também pode carregar um documento JSON que represente seu esquema em uma única chamada da API

PutSchemaFromJson. Para obter mais informações, consulte Formato do esquema JSON. Para visualizar um esquema de exemplo de links tipados, consulte Documento de esquema com links digitados.

Os tipos de alterações permitidas em fases diferentes de ciclo de vida de desenvolvimento do esquema são semelhantes às alterações que são permitidas para a manipulação de facetas de objetos. Os esquemas no estado de desenvolvimento são compatíveis com qualquer alteração. Os esquemas no estado publicado são imutáveis e nenhuma alteração é compatível. Somente certas alterações são permitidas em esquemas que são aplicados a um diretório de dados. Depois de definir a ordem e os atributos em uma faceta de link tipado aplicada, essa ordem não pode ser alterada.

Duas outras operações de API listam facetas e seus atributos:

- ListTypedLinkFacetAttributes
- ListTypedLinkFacetNames

Interação de link tipado

Depois que uma faceta de link tipado foi criada, você estará pronto para começar a criar e interagir com links tipados. Para anexar e desanexar links tipados, use as operações da API AttachTypedLink e DetachTypedLink.

O TypedLinkSpecifier é uma estrutura que contém todas as informações para identificar exclusivamente um link tipado. Nessa estrutura você pode localizar TypedLinkFacet, SourceObjectID, DestinationObjectID e IdentityAttributeValues. Esses são usados para especificar exclusivamente o link tipado que está sendo operado. A operação da API AttachTypedLink retorna um especificador de link tipado, enquanto a operação da API DetachTypedLink aceitar um como entrada. De maneira semelhante, as operações da API ListIncomingTypedLinks e ListOutgoingTypedLinks fornecem especificadores de links tipados como saída. Você também pode criar um especificador de link tipado a partir do zero. A lista completa de operações da API relacionadas a links tipados, inclui o seguinte:

- AttachTypedLink
- CreateTypedLinkFacet
- DeleteTypedLinkFacet
- DetachTypedLink

- GetLinkAttributes
- GetTypedLinkFacetInformation
- ListIncomingTypedLinks
- ListOutgoingTypedLinks
- ListTypedLinkFacetNames
- ListTypedLinkFacetAttributes
- UpdateLinkAttributes
- UpdateTypedLinkFacet

> **Note**
>
> As referências de atributos e a atualização de links tipados não são compatíveis. Para atualizar um link tipado, você deve removê-lo e adicionar a versão atualizada.

Filtros de intervalo

Várias APIs de lista do Cloud Directory permitem especificar um filtro na forma de um intervalo. Esses filtros permitem que você selecione subconjuntos dos links anexados ao nó especificado de maneira eficiente.

Os intervalos são fornecidos normalmente como um mapa (matriz de pares de chave-valor) cujas chaves são identificadores de atributos e cujos valores são os intervalos correspondentes. Isso permite filtrar links cujas identidades consistem em um ou mais atributos. Por exemplo, uma configuração de TypedLink para modelar uma relação de função para determinar permissões pode ter os atributos RoleType e Authorizer. Uma chamada ListOutgoingTypedLinks poderia especificar intervalos para filtrar o resultado como RoleType: “Admin” e Authorizer: “Julia”. O mapa de intervalos usado para filtrar uma única solicitação de lista deve conter somente os atributos que definem a identidade do link (o OrderedIndexedAttributeList de um índice ou o IdentityAttributeOrder de um TypedLink), mas não precisa conter intervalos para todos eles. Os intervalos ausentes serão preenchidos automaticamente com os intervalos que abrangem todos os valores possíveis (do PRIMEIRO ao ÚLTIMO).

Se você considerar cada atributo como a definição de um domínio plano independente de valores, as estruturas do intervalo definirão dois pontos lógicos nesse domínio — os pontos de início e de

término — e o intervalo corresponderá a todos os pontos possíveis entre esses pontos. O StartValue e o EndValue de estrutura do intervalo definem a base para esses dois pontos com os “modos” os refinando adicionalmente para indicar se cada próprio ponto deve ser incluído ou excluído do intervalo. No exemplo de RoleType: "Admin" acima, os dois valores do atributo RoleType seriam “Admin”, e os modos seriam “INCLUSIVE” (escritos como [“Admin” to “Admin”]). Um filtro para uma chamada de ListIndex onde o índice é definido no LastName de uma faceta de usuário pode usar StartValue=”D”, StartMode=INCLUSIVE, EndValue: “G”, EndMode:EXCLUSIVE para reduzir a listagem a nomes que começam com D, E ou F.

O ponto de início de um intervalo sempre deve preceder ou ser igual ao ponto de término. O Cloud Directory retornará um erro se EndValue preceder o StartValue. Os valores também devem ser do mesmo tipo primitivo que o atributo que estão filtrando, valores de String para um atributo String, Integer para um atributo Integer e assim por diante. StartValue=“D”, StartMode=EXCLUSIVE, EndValue=“D”, EndMode=INCLUSIVE é inválido, por exemplo, porque o ponto de término inclui o valor enquanto o ponto de início segue o valor.

Existem três modos especiais que podem ser usados por pontos de início ou de término. Os seguintes modos não requerem que o campo de valor correspondente seja especificado, pois implicam uma posição em si mesmos.

- FIRST - precede todos os valores possíveis no domínio. Quando usado para o ponto de início, corresponde a todos os valores possíveis do início do domínio até o ponto de término. Quando usado para o ponto de término, nenhum valor no domínio corresponderá ao intervalo.
- LAST - segue todos os valores possíveis no domínio. Quando usado para o ponto de término, corresponde a todos os valores que seguem o ponto de início, incluindo valores ausentes. Quando usado para o ponto de início, nenhum valor no domínio corresponderá ao intervalo.
- LAST_BEFORE_MISSING_VALUES - este modo só é útil para atributos opcionais em que o valor pode ser omitido (consulte Valores ausentes). Corresponde ao ponto entre os valores ausentes e os valores reais do domínio. Quando usado para o ponto de término, corresponde a todos os valores não ausentes do domínio que seguem o ponto de início. Quando usado para o ponto de início, exclui todos os valores não ausentes do domínio. Se o atributo for exigido, esse modo será equivalente a LAST, pois não pode haver nenhum valor ausente.

Várias limitações de intervalo

O Cloud Directory limita padrões onde há vários atributos para garantir o processamento de solicitações eficiente e de baixa latência. Cada link com vários atributos de identificação especifica-

os em uma ordem bem-definida. Por exemplo, o exemplo de Role acima define o atributo RoleType como o mais significativo, e o atributo Authorizer como o menos significativo. Uma solicitação de lista pode especificar somente um único intervalo “de qualificação” que não seja 1) um único valor ou 2) que abranja todos os valores possíveis (pode haver vários intervalos que correspondem a esses dois requisitos). Todos os intervalos de atributos mais significativos que o atributo do intervalo de qualificação devem especificar um único valor, e qualquer intervalo de intervalos menos significativos deve abranger todos os valores possíveis. No exemplo de Role, os conjuntos de filtros (RoleType:”Admin”, Authorizer:[“J” to “L”]) (valor único + intervalo de qualificação), (RoleType: [”Admin” to “User”]) (intervalo de qualificação + intervalo abrangente implícito) e (RoleType:[FIRST to LAST]) (dois intervalos abrangentes, um implícito) são todos exemplos de conjuntos válidos de filtros. (RoleType:[FIRST to LAST], Authorizer:”Julia”) não é um conjunto válido, pois o intervalo abrangente é mais significativo que o intervalo de valor único.

Alguns padrões úteis ao preencher as estruturas de intervalos, incluem:

Corresponder a um único valor

Especifique o valor para StartValue e EndValue e defina os dois modos como “INCLUSIVE”.

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

Corresponder a um prefixo

Especifique o prefixo como o StartValue com modo INCLUSIVE, e o primeiro valor depois do prefixo como EndValue com um modo EXCLUSIVE.

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

Filtrar por maior que um valor

Especifique o valor para o StartValue com modo EXCLUSIVE, e LAST como o EndMode (ou LAST_BEFORE_MISSING_VALUES para excluir valores ausentes, se aplicável).

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

Filtrar por menor que ou igual a um valor

Especifique o valor para o EndValue com modo INCLUSIVE, e FIRST como o StartMode.

Valores ausentes

Quando um atributo é marcado como opcional no esquema, seu valor pode ser “ausente” uma vez que ele não precisaria ter sido fornecido quando a faceta foi anexada ou o atributo poderia ter sido excluído subsequentemente. Se o objeto com um valor ausente for anexado a um índice, o link do índice ainda estará presente, mas movido para o final do conjunto de links. Uma chamada para `ListIndex` retornará primeiro todos os links onde os atributos indexados estão todos presentes antes de retornar os links onde um ou mais estão ausentes. Isso é aproximadamente semelhante ao valor NULL de um banco de dados relacional, com esses valores ordenados depois dos valores não NULL. Você pode especificar se um intervalo inclui esses valores ausentes ou não escolhendo os modos LAST ou LAST_BEFORE_MISSING_VALUES. Por exemplo, você fornece um filtro para uma chamada de ListIndex para retornar apenas os valores ausentes em um índice filtrando com o intervalo [LAST_BEFORE_MISSING_VALUES to LAST].

Acessar objetos

Os objetos em um diretório podem ser acessados por caminho ou por `objectIdentifier`.

Caminho— cada objeto em uma árvore do Cloud Directory pode ser identificado e localizado pelo nome do caminho que descreve como acessá-lo. O caminho começa no nó raiz do diretório (nó `000` na figura anterior). A notação do caminho começa com o link rotulado com uma barra (/) e segue os links filho separados pelo separador de caminho (também uma barra) até atingir a última parte do caminho. Por exemplo, o objeto `005` na figura anterior pode ser identificado usando o caminho `/group/a/d`. Vários caminhos podem identificar um objeto, uma vez que os objetos que são nós folha podem ter vários pais. O caminho a seguir também pode ser usado para identificar o objeto `005` : /group/b/e

ObjectIdentifier— cada objeto no diretório tem um identificador exclusivo global, que é o`ObjectIdentifier`.`ObjectIdentifier`O é retornado como parte do`CreateObject`Chamada de API. Você também pode buscar o `ObjectIdentifier` usando a chamada da API `GetObjectInformation`. Por exemplo, para buscar o identificador do objeto `005`, você pode chamar `GetObjectInformation` com a referência do objeto como o caminho que resulta no objeto, que é group/b/e ou group/a/d.

```
GetObjectInformationRequest request = new GetObjectInformationRequest()
    .withDirectoryArn(directoryArn)
    .withObjectReference("/group/b/e")
    .withConsistencyLevel(level)
```

```
GetObjectInformationResult result = cdClient.getObjectInformation(request)
String objectIdentifier = result.getObjectIdentifier()
```

Preencher objetos

As facetas novas podem ser adicionadas a um objeto usando a chamada da API `AddFacetToObject`. O tipo do objeto é determinado com base nas facetas anexadas ao objeto. O anexo do objeto em um diretório funciona com base no tipo do objeto. Para anexar um objeto, lembre-se destas regras:

- Um objeto de nó folha não pode ter filhos.
- Um objeto de nó pode ter vários filhos.
- Um objeto do tipo política não pode ter filhos e pode ter zero ou um pai.

Atualização de objetos

Você pode atualizar um objeto de várias maneiras:

1. Usar a operação `UpdateObjectAttributes` para atualizar atributos individuais de facetas em um objeto.
2. Usar a operação `AddFacetToObject` para adicionar novas facetas a um objeto.
3. Usar a operação `RemoveFacetFromObject` para excluir facetas existentes de um objeto.

Excluir objetos

Um objeto anexado deve atender a certas condições para que você possa excluí-lo de um diretório:

1. Você deve desanexar o objeto da árvore. Você pode desanexar um objeto somente quando ele não tiver nenhum filho. Se o objeto tiver filhos, você deve desanexar todos os filhos primeiro.
2. Você pode excluir um objeto desanexado somente se todos os atributos desse objeto estiverem excluídos. Você pode excluir atributos em um objeto excluindo cada faceta anexada a esse objeto. Você pode buscar uma lista de facetas anexadas a um objeto chamando `GetObjectInformation`.
3. Um objeto também não deve ter nenhum pai, nenhum anexo de política e nenhum anexo de índice.

Como um objeto deve estar totalmente desanexado da árvore a ser excluída, você deve usar o identificador do objeto para excluí-lo.

Consultar objetos

Esta seção aborda os vários elementos relevantes para consultar objetos em um diretório.

Travessia do diretório

Como o Cloud Directory é uma árvore, você pode consultar objetos da parte superior para baixo usando o`ListObjectChildren`operação da API ou de baixo para cima usando o`ListObjectParents`Operação da API.

Pesquisa de política

Dada uma referência de objeto, a operação da API `LookupPolicy` retorna todas as políticas que estão anexadas ao longo de seu caminho (ou caminhos) à raiz de uma maneira de cima para baixo. Todos os caminhos que não estiverem levando a raiz são ignorados. Todos os objetos de tipo de política são retornados.

Se o objeto for um nó folha, ele poderá ter vários caminhos até a raiz. Essa chamada retorna somente um caminho para cada chamada. Para buscar caminhos adicionais, use o token de paginação.

Consulta de índice

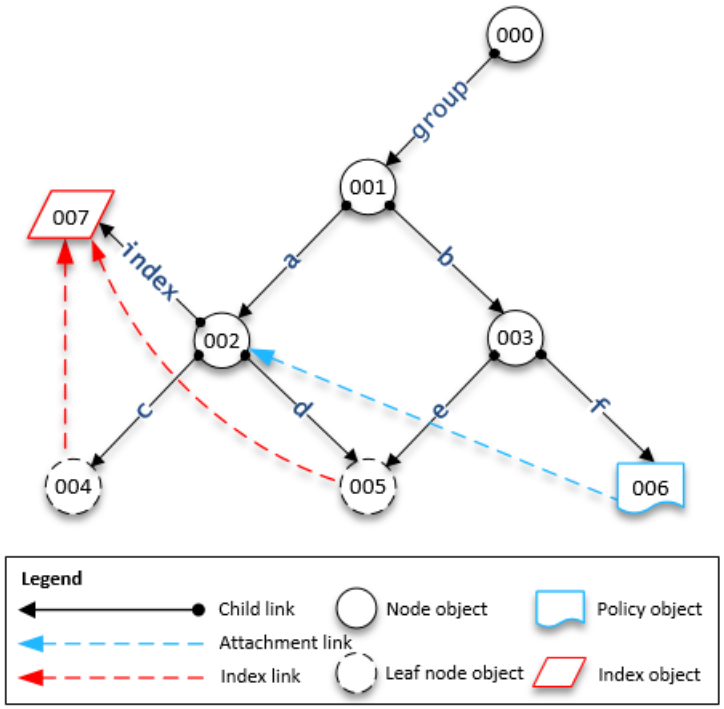
O Cloud Directory oferece suporte à funcionalidade sofisticada de consulta de índice com o uso dos seguintes intervalos:

- FIRST - começa no primeiro valor do atributo indexado. O valor inicial do atributo é opcional.
- LAST - retorna os valores do atributo até o final do índice, incluindo os valores ausentes. O valor final do atributo é opcional.
- LAST_BEFORE_MISSING_VALUES - retorna os valores do atributo até o final do índice, excluindo os valores ausentes.
- INCLUSIVE - inclui o valor do atributo que está sendo especificado.
- EXCLUSIVE - exclui o valor do atributo que está sendo especificado.

Listagem de caminhos pai

Usando a chamada da API `ListObjectParentPaths`, você pode recuperar todos caminhos pai disponíveis para qualquer tipo de objeto (nó, nó folha, nó de política, nó de índice). Essa operação da API pode ser útil quando você precisar avaliar todos os pais de um objeto. A chamada retorna todos os objetos da raiz do diretório até o objeto solicitado. Também retorna o número de caminhos com base no `MaxResults` definido pelo usuário, no caso de vários caminhos até o pai. A ordem dos caminhos e dos nós retornados é consistente entre várias chamadas da API a menos que os objetos sejam excluídos ou movidos. Os caminhos que não levam para a raiz do diretório são ignorados do objeto de destino.

Para um exemplo de como isso funciona, digamos que um diretório tem uma hierarquia de objeto semelhante à ilustração mostrada a seguir.



As formas numeradas representam os diferentes objetos. O número de setas entre o objeto e a raiz do diretório (`000`) representa o caminho completo e seria expresso na saída. As tabela a seguir mostra as solicitações e as respostas às consultas feitas a objetos de nós folha específicos na hierarquia.

Consultas de exemplo nos objetos

Solicitação	Resposta
`004, PageToken : null, MaxResults: 1`	`[{/group/a/c], [000, 001, 002, 004]}], PageToken: null`
`005, PageToken : null, MaxResults: 2`	`[{/group/a/d, [000, 001, 002, 005]}, { /group/b/e, [000, 001, 003, 005]}], PageToken: null` **Note** Neste exemplo, o objeto `005` tem dois nós `002` e `003` como pais. Além disso, como `MaxResults` é 2, os dois caminhos exibem objetos em uma 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** Neste exemplo, o objeto `005` tem dois nós `002` e `003` como pais. Além disso, como `MaxResults` é 1, várias chamadas paginadas com tokens de página serão feitas para obter todos os caminhos com uma lista de objetos.
`006, PageToken : null,`	`[{/group/b/f, [000, 001, 003, 006]}], PageToken: null`

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

Níveis de consistência

O Amazon Cloud Directory é um armazenamento de diretório distribuído. Os dados são distribuídos para vários servidores em diferentes zonas de disponibilidade. Uma solicitação de gravação bem-sucedida atualiza os dados em todos os servidores. Os dados estarão disponíveis eventualmente em todos os servidores, geralmente em um segundo. Para ajudar os usuários do serviço, o Cloud Directory oferece dois níveis de consistência para operações de leitura. Esta seção descreve os diferentes níveis de consistência e a natureza eventualmente consistente do Cloud Directory.

Níveis de isolamento de leitura

Ao ler dados do Cloud Directory, você deve especificar o nível de isolamento no qual deseja ler. Diferentes níveis de isolamento têm compensações entre latência e atualização dos dados.

- EVENTUAL— o nível de isolamento de snapshot lê qualquer dado que esteja disponível imediatamente.Fornece a latência mais baixa de qualquer nível de isolamento. Também fornece uma visualização potencialmente antiga dos dados no diretório. O isolamento EVENTUAL não fornece consistência de leitura depois da gravação. Isso significa que não há garantia de que você possa ler os dados imediatamente depois de gravá-los.
- SERIALIZÁVELO nível de isolamento serializável fornece o nível mais alto de consistência oferecido pelo Cloud Directory. As leituras feitas no nível de isolamento SERIALIZABLE garantem que você receba dados de todas as gravações bem-sucedidas. Se foi feita uma alteração nos dados que você solicitou, e essa alteração ainda não estiver disponível, o sistema rejeitará sua solicitação com `RetryableConflictException`. Recomendamos que você tente novamente essas exceções (consulte a seção a seguir). Quando você tenta novamente, as leituras de SERIALIZABLE oferecem consistência de leitura após gravação.

Solicitações de gravação

O Cloud Directory garante que várias solicitações de gravação não atualizem simultaneamente o mesmo objeto ou objetos. Se duas solicitações de gravação forem descobertas operando nos mesmos objetos, uma das operações falhará com uma `RetryableConflictException`. Recomendamos que você tente novamente essas exceções (consulte a seção a seguir).

> **Note**
>
> `RetryableConflictException` as respostas recebidas durante operações de gravação não podem ser usadas para detectar condições de corrida. Considerando um caso de uso que mostrou precipitar essa situação, não há nenhuma garantia de que uma exceção sempre ocorrerá. Se uma exceção ocorrerá ou não depende da ordem em que cada solicitação será processada internamente.

RetryableConflictExceptions

Ao executar operações de gravação ou operações de leitura com um nível de isolamento SERIALIZABLE após uma gravação no mesmo objeto, o Cloud Directory pode responder com uma`RetryableConflictException`. Essa exceção indica que os servidores do Cloud Directory ainda não processaram o conteúdo da gravação anterior. Essas situações são transitórias e são autocorrigidas rapidamente. É importante observar que o `RetryableConflictException` não pode ser usado para detectar nenhum tipo de consistência de leitura-após-gravação. Não há nenhuma garantia de que um caso de uso específico causará essa exceção.

Recomendamos que você configure seus clientes do Cloud Directory para tentar novamente a`RetryableConflictException`. Essa configuração fornece comportamento livre de erros durante a operação. O código de exemplo a seguir demonstra como essa configuração pode ser feita em 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);
 }
}
```

Indexação e pesquisa

O Amazon Cloud Directory suporta dois métodos de indexação: Com base em valor e baseado em tipo. A indexação com base em valor é a forma mais comum. Com ela você poderá indexar e pesquisar objetos no diretório com base nos valores dos atributos do objeto. Com indexação baseada em tipo, você pode indexar e pesquisar por objetos no diretório com base em tipos de objeto. Facetas ajudam a definir tipos de objeto. Para obter mais informações sobre esquemas e facetas, consulte Schemas e Facets.

Os índices no Cloud Directory permitem a simples listagem de outros objetos pelos valores de atributo ou faceta deles. Cada índice é definido na criação para funcionar com um atributo ou faceta de nome específico. Por exemplo, um índice pode ser definido no atributo “email” da faceta “Person”. Os índices são objetos de primeira classe, o que significa que os clientes podem criá-los, modificá-los, listá-los e excluí-los com flexibilidade, de acordo com as necessidades de lógica do aplicativo.

Conceitualmente, os índices são semelhantes a nós com filhos, nos quais os links para os nós indexados são rotulados de acordo com os atributos indexados, em vez de receberem um rótulo quando o filho estiver anexado. No entanto, os links de índices não são bordas pai-filho e têm seu próprio conjunto de operações da API de enumeração.

É importante compreender que os índices no Cloud Directory não são preenchidos automaticamente, pois podem estar em outros sistemas. Em vez disso, você usa chamadas de API para anexar e separar diretamente objetos de ou para o índice. Embora dê um pouco mais de trabalho, isso lhe dá a flexibilidade de definir escopos de índice variados. Por exemplo, você pode definir um índice que rastreie somente filhos diretos de um nó específico. Ou você pode definir um índice que acompanhe todos os objetos em uma determinada ramificação em uma raiz local, como todos os nós em um departamento. Você pode também fazer os dois ao mesmo tempo.

Tópicos

- Ciclo de vida do índice
- Indexação baseada em facetas
- Índices exclusivos vs. não exclusivos

Ciclo de vida do índice

Você pode usar as seguintes chamadas de API para ajudar com o ciclo de vida de desenvolvimento dos índices.

1. Você cria índices com a chamada de API `CreateIndex`. Você fornece uma estrutura de definição de índice que descreve os atributos sobre objetos anexados que o índice rastreará. A definição também indica mesmo se o índice deve ou não aplicar exclusividade. O resultado é um ID de objeto para o novo índice, que deve imediatamente ser conectado à sua hierarquia como qualquer outro objeto. Por exemplo, isso pode ser uma ramificação dedicada a manter índices.
2. Você anexa objetos ao índice manualmente com a chamada de API `AttachToIndex`. O índice então rastreia automaticamente os valores de seus atributos definidos em cada objeto anexado.
3. Para usar os índices de pesquisa para objetos com enumeração mais eficiente, chamar `ListIndex` e especifique um intervalo de valores nos quais você está interessado.
4. Use a chamada de API `ListAttachedIndices` para enumerar os índices anexados a determinado objeto.
5. Use a chamada de API `DetachFromIndex` para remover manualmente objetos do índice.
6. Depois de separar todos os objetos do índice, você pode excluir o índice com a chamada de API `DeleteObject`.

Não há limite quanto ao número de índices em um diretório, além do limite sobre o espaço usado por todos os objetos. Os índices e seus anexos consomem espaço, mas são semelhantes ao consumidos por nós e links pai—filho. Há um limite quanto ao número de índices que podem ser anexados a determinado objeto. Para obter mais informações, consulte Entre os limites do Amazon Cloud Directory.

Indexação baseada em facetas

Com indexação e pesquisa baseadas em facetas, você pode otimizar suas pesquisas de diretório procurando somente um subconjunto do diretório. Para isso, você deve usar uma faceta do esquema. Por exemplo, em vez de pesquisar em todos os objetos do usuário no seu diretório, você pode pesquisar somente os objetos do usuário que contêm uma faceta do funcionário. Essa eficiência ajuda a reduzir o tempo de latência e a quantidade de dados recuperados para consulta.

Com a indexação baseada em facetas, você pode usar as operações de API do índice do Cloud Directory para criar e anexar um índice às facetas dos objetos. Você também pode listar os resultados do índice e filtrá-los com base em determinadas facetas. Isso pode reduzir efetivamente o tempo de consulta e a quantidade de dados, estreitando o escopo de pesquisa somente a objetos contendo determinado tipo de facetas.

O atributo “`facets`” que é usado com as chamadas de API `CreateIndex` e `ListIndex` faz surgir a coleção de facetas aplicadas a um objeto. Este atributo está disponível para uso somente com as chamadas de API `CreateIndex` e `ListIndex`. Conforme exibido no código de exemplo a seguir, o ARN do esquema usa a região, a conta do proprietário e o ID do diretório para fazer referência ao esquema do Cloud Directory. Esse esquema fornecido por serviço não aparece nas listagens.

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

Por exemplo, o código de exemplo a seguir cria um índice baseado em faceta específico da sua conta da AWS e do diretório, onde você pode enumerar todos os objetos criados com a faceta `SalesDepartmentFacet`.

> **Note**
>
> Use o valor das “facetas” nos parâmetros, conforme exibido abaixo. As instâncias das “facetas” mostradas no código de exemplo referem-se a um valor fornecido e controlado pelo serviço do Cloud Directory. Você pode usá-las para indexação, mas pode ter acesso de somente leitura.

```
// 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 exclusivos vs. não exclusivos

Os índices exclusivos diferem dos índices não exclusivos na aplicação da exclusividade dos valores de atributo indexados para objetos anexados ao índice. Por exemplo, você pode querer preencher objetos Person em dois índices: um exclusivo em um atributo “email” e um não exclusivo no atributo “lastname”. O índice lastname permite que vários objetos Person com o mesmo sobrenome estejam vinculados. Por outro lado, a chamada `AttachToIndex` destinada ao índice do e-mail apresenta o erro `LinkNameAlreadyInUseException` se uma pessoa com os mesmos atributos do e-mail já estiver anexada. Observe que o erro não remove o objeto Person em si. Portanto, um aplicativo pode criar a Person, anexá-la à hierarquia e anexá-la a índices, tudo em uma única solicitação em lote. Isso garante que, se a exclusividade for violada em algum de índices, o objeto e todos seus anexos serão implementados de volta automaticamente.

Como administrCloud Directory

Esta seção lista todos os procedimentos para operar e manter um ambiente do Cloud Directory.

Tópicos

- Gerenciar diretórios do
- Gerenciar seu esquema

Gerenciar diretórios do

Esta seção descreve como manter tarefas comuns de diretório para o seu ambiente do Cloud Directory.

Tópicos

- Crie seu diretório
- Excluir seu diretório
- Desativar o diretório do
- Habilitar seu diretório

Crie seu diretório

Antes de criar um diretório no Amazon Cloud Directory, o AWS Directory Service requer que você aplique primeiro um esquema a ele. Um diretório não pode ser criado sem um esquema e, normalmente, tem um esquema aplicado a ele. Entretanto, você usa as Operações da API do Cloud Directory para aplicar esquemas adicionais a um diretório. Para obter mais informações, consulteApplySchemanoGuia de referência da API do Amazon Cloud Directory.

Para criar um Cloud Directory

1. NoAWS Directory Servicepainel de navegação, emDiretório na nuvem, escolhaDiretórios.
2. SelecioneConfigurar o Cloud Directory.
3. UnderEscolha um esquema para aplicar ao seu novo diretório, digite o nome amigável do diretório, como`User Repositorye`, em seguida, escolha uma das seguintes opções:
    - Esquema gerenciado

- Exemplo de esquema
- Esquema personalizado

Esquemas de amostra e esquemas personalizados são colocados no arquivoDesenvolvimentoestado, por padrão. Para obter mais informações sobre estados de esquema, consulte Ciclo de vida do esquema. Antes que um esquema seja aplicado a um diretório, ele deve ser convertido ao estado Published. Para publicar com êxito um esquema de exemplo usando o console, você deve ter permissões para as ações a seguir:

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

Visto que os esquemas de exemplo são modelos somente leitura fornecidos pela AWS, eles não podem ser publicados diretamente. Em vez disso, quando você optar por criar um diretório com base em um esquema de exemplo, o console criará uma cópia temporária do esquema de exemplo que você tiver selecionado e a colocará na caixaDesenvolvimentoEstado. Em seguida, ele criará uma cópia daquele esquema de desenvolvimento e a colocará no estado Published. Depois de publicado, o esquema de desenvolvimento será excluído, o que explica o motivo da ação `DeleteSchema` ser necessária quando se publica um esquema de exemplo.

4. Escolha Next (Próximo).
5. Revise as informações do diretório e faça as alterações necessárias. Quando as informações estiverem corretas, selecione Create (Criar).

Excluir seu diretório

Use o procedimento a seguir para excluir um diretório no Cloud Directory.

> **Note**
>
> Antes de excluir um diretório, você deve primeiro desativá-lo. Para obter instruções, consulte Desativar o diretório do.

Como excluir um diretório

1. NoAWS Directory Servicepainel de navegação, emDiretório na nuvem, selecioneDiretórios.
2. Selecione a opção na tabela ao lado da ID do diretório que você deseja excluir.
3. Escolha Actions.
4. SelecioneExcluir
5. NoExcluir diretório, confirme a operação digitando o nome do seu diretório e escolhaExcluir.

Desativar o diretório do

Use o procedimento a seguir para desabilitar um diretório no Cloud Directory do.

Para desativar um diretório

1. NoAWS Directory Servicepainel de navegação, emDiretório na nuvem, selecioneDiretórios.
2. Selecione a opção na tabela ao lado da ID do diretório que você deseja desativar.
3. Escolha Actions.
4. SelecioneDesabilitar o

Habilitar seu diretório

Use o procedimento a seguir para habilitar um diretório desabilitado anteriormente no Cloud Directory.

Para habilitar um diretório

1. NoAWS Directory Servicepainel de navegação, emDiretório na nuvem, selecioneDiretórios.
2. Selecione a opção na tabela ao lado da ID do diretório que você deseja ativar.
3. Escolha Actions.
4. SelecioneHabilitar

Gerenciar seu esquema

Esta seção descreve como manter tarefas de esquemas comuns para o seu ambiente do Cloud Directory.

Tópicos

- Criar seu esquema
- Excluir um esquema
- Fazer download de um esquema
- Publicar um esquema
- Atualize seu esquema
- Atualize o esquema

Criar seu esquema

O Amazon Cloud Directory oferece suporte ao carregamento de arquivos JSON que sejam compatíveis para a criação de esquemas. Para criar um esquema novo, você pode criar seu próprio arquivo JSON do zero ou fazer download de um dos esquemas existentes listados no console. Em seguida, faça upload dele como esquema personalizado. Para obter mais informações, consulte Esquemas personalizados.

Você também pode criar, excluir, baixar, listar, publicar, atualizar e atualizar esquemas usando as APIs do Cloud Directory. Para obter mais informações sobre as operações da API do esquema, consulte oAmazon Cloud Directory Service Guia de referência.

Escolha qualquer pessoa procedimentos abaixo, dependendo de seu método preferido.

Para criar um esquema personalizado

1. NoAWS Directory Service Consolepainel de navegação, emDiretório na nuvem, escolhaSchemas.
2. Crie um arquivo JSON com todas as definições novas do esquema. Para obter mais informações sobre como formatar um arquivo JSON, consulte Formato do esquema JSON.
3. No console, escolhaUpload new esquema.
4. NoUpload new esquema, digite um nome para o esquema.

5. SelectEscolher arquivo, selecione o novo arquivo JSON que você acabou de criar e escolhaAberto.
6. Escolha Upload (Fazer upload). Isso adiciona um novo esquema à sua biblioteca de esquemas e o coloca noDesenvolvimentoEstado. Para obter mais informações sobre estados de esquema, consulte Ciclo de vida do esquema.

Para criar um esquema personalizado com base em um existente no console

1. NoAWS Directory Service Consolepainel de navegação, emDiretório na nuvem, escolhaSchemas.
2. Na lista de lista os esquemas, selecione a opção perto do esquema que você deseja copiar.
3. Escolha Actions.
4. SelecioneFazer download do esquema.
5. Renomeie o arquivo JSON, edite-o conforme necessário e salve o arquivo. Para obter mais informações sobre como formatar um arquivo JSON, consulte Formato do esquema JSON.
6. No console, escolhaUpload new esquema, selecione o arquivo JSON que você acabou de editar e escolhaAberto.

   Isso adiciona um novo esquema à sua biblioteca de esquemas e o coloca noDesenvolvimentoEstado. Para obter mais informações sobre estados de esquema, consulte Ciclo de vida do esquema.

Excluir um esquema

Use o procedimento a seguir para excluir um esquema no Cloud Directory (Diretório do Cloud Directory).

Para excluir um esquema

1. NoAWS Directory Service Consolepainel de navegação, emDiretório na nuvem, selecioneSchemas.
2. Selecione a opção na tabela ao lado do nome do esquema que você deseja excluir.
3. Escolha Actions.
4. SelecioneExcluir
5. NoExcluir esquema, confirme a operação escolhendoExcluir.

Fazer download de um esquema

Use o procedimento a seguir para fazer download de um esquema.

Para fazer download de um esquema

1. NoAWS Directory Service Consolepainel de navegação, emDiretório na nuvem, selecioneSchemas.
2. Selecione a opção na tabela ao lado do nome do esquema que você deseja baixar.
3. Escolha Actions.
4. SelecioneFazer download do esquema

Publicar um esquema

Use o procedimento a seguir para publicar um esquema no Cloud Directory (Diretório do Cloud Directory).

Para publicar um esquema

1. NoAWS Directory Service Consolepainel de navegação, emDiretório na nuvem, selecioneSchemas.
2. Selecione a opção na tabela ao lado do nome do esquema que deseja publicar.
3. Escolha Actions.
4. SelecionePublicar
5. NoPublicar esquema, forneça as seguintes informações:
   a. Nome do esquema
   b. Versão principal
   c. Versão secundária
6. Escolha Publish.

Atualize seu esquema

Use o procedimento a seguir para atualizar um esquema no Cloud Directory (Diretório do Cloud Directory).

Para atualizar um esquema

1. NoAWS Directory Service Consolepainel de navegação, emDiretório na nuvem, selecioneSchemas.
2. Selecione a opção na tabela ao lado do nome do esquema que deseja atualizar.
3. Escolha Actions.
4. Escolha Update (Atualizar)
5. NoAtualizar esquema, modifique opcionalmente a caixa de diálogoNome do esquemaou selecioneEscolher arquivoPara aplicar ou remover facetas e atributos.
6. Escolha Update.

Atualize o esquema

Atualizar um esquema adicionará as facetas e atributos escolhidos ao esquema publicado selecionado. Use o procedimento a seguir para atualizar um esquema publicado.

Para atualizar um esquema

1. NoAWS Directory Service Consolepainel de navegação, emDiretório na nuvem, selecioneSchemas.
2. Selecione a opção na tabela ao lado do nome do esquema que você deseja atualizar.
3. Escolha Actions.
4. SelecioneUpgrade
5. NoAtualizar o esquema publicadoEscolha uma das seguintes opções e escolhaUpgrade:
   - Escolha a partir da sua lista atual de esquemas de desenvolvimento
   - Upload un novo arquivo de esquema (JSON)
6. SelecioneAtualizar.

Segurança no Amazon Cloud Directory

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você aproveita um datacenter e uma arquitetura de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve a segurança da nuvem e a segurança na nuvem:

- Segurança da nuvem: a AWS é responsável por proteger a infraestrutura que executa serviços da AWS na Nuvem AWS. A AWS também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao Amazon Cloud Directory, consulteServiços da AWS no escopo pelo programa de conformidade.
- Segurança na nuvem: a responsabilidade é determinada pelo serviço da AWS usado. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Cloud Directory. Os tópicos a seguir mostram como configurar o Cloud Directory para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros serviços da AWS que ajudam a monitorar e proteger os recursos do Cloud Directory.

Tópicos

- Identity and Access Management no Amazon Cloud Directory
- Registrar em log e monitorar no Amazon Cloud Directory
- Validação de conformidade para o Amazon Cloud Directory
- Resiliência no Amazon Cloud Directory
- Segurança de infraestrutura no Amazon Cloud Directory

Identity and Access Management no Amazon Cloud Directory

O acesso ao Amazon Cloud Directory exige credenciais que a AWS possa usar para autenticar as solicitações. Essas credenciais devem ter permissões para acessar recursos da AWS. As seguintes

seções fornecem detalhes sobre como você pode usar oAWS Identity and Access Management (IAM)E o Cloud Directory para ajudar a proteger seus recursos controlando quem pode acessá-los:

- Authentication
- Controle de acesso

Authentication

Você pode acessar a AWS como alguns dos seguintes tipos de identidades:

- Usuário raiz da conta da AWS: ao criar uma conta AWS, você começa com uma única identidade de login que tem acesso completo a todos os serviços e recursos da AWS na conta. Essa identidade é denominada usuário raiz da conta da AWS e é acessada pelo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável que você não use o usuário raiz nas tarefas diárias, nem mesmo nas administrativas. Em vez disso, siga as práticas recomendadas para o uso do usuário raiz somente a fim de criar seu primeiro usuário do IAM. Depois, armazene as credenciais do usuário raiz com segurança e use-as para executar somente algumas tarefas de gerenciamento de contas e de serviços.
- Usuário do IAM— UmUsuário do IAMO é uma identidade na sua conta da AWS com permissões personalizadas específicas (por exemplo, permissões para criar um diretório no Cloud Directory). É possível usar um nome de usuário e uma senha do IAM para fazer login em páginas da Web seguras da AWS, como o Console de Gerenciamento da AWS, os Fóruns de discussão da AWS ou o AWS Support Center.

  Além de um nome e senha de usuário, você também pode gerar chaves de acesso para cada usuário. Você pode usar essas chaves ao acessar serviços da AWS de forma programática, seja por meio de um dos vários SDKs ou usando a Interface da linha de comando (CLI) da AWS. As ferramentas de SDK e de CLI usam as chaves de acesso para o cadastramento criptográfico da sua solicitação. Se você não utilizar ferramentas da AWS, assine a solicitação você mesmo. Cloud DirectorySignature versão 4, um protocolo para autenticar solicitações de API de entrada. Para obter mais informações sobre solicitações de autenticação, consulte Processo de assinatura do Signature versão 4 na Referência geral da AWS.

- Função do IAM: uma função do IAM é uma identidade do IAM que você pode criar em sua conta com permissões específicas. Uma função do IAM é semelhante a um usuário do IAM, pois é uma identidade da AWS com políticas de permissão que determinam o que a identidade pode e não pode fazer na AWS. No entanto, em vez de ser exclusivamente associada a uma pessoa, uma função destina-se a ser assumida por qualquer pessoa que precisar dela. Além disso, uma função não tem credenciais de longo prazo padrão, como uma senha ou chaves de acesso, associadas a ela. Em vez disso, quando você assumir uma função, ela fornecerá credenciais de segurança temporárias para sua sessão de função. As funções do IAM com credenciais temporárias são úteis nas seguintes situações:

  - Acesso de usuário federado: em vez de criar um usuário do IAM, é possível usar identidades existentes do AWS Directory Service, o diretório de usuários da sua empresa ou um provedor de identidades da Web. Estes são conhecidos como usuários federados. A AWS atribui uma função a um usuário federado quando o acesso é solicitado por meio de um provedor de identidades. Para obter mais informações sobre usuários federados, consulte Usuários e funções federados no Guia do usuário do IAM.

  - Acesso ao serviço da AWS: uma função de serviço é uma função do IAM que um serviço assume para realizar ações em seu nome. As funções de serviço fornecem acesso apenas dentro de sua conta e não podem ser usadas para conceder acesso a serviços em outras contas. Um administrador do IAM pode criar, modificar e excluir uma função de serviço do IAM. Para obter mais informações, consulte Criar uma função para delegar permissões a um serviço da AWS no Guia do usuário do IAM.

  - Aplicações em execução no Amazon EC2: é possível usar uma função do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 que fazem solicitações de API da AWS ou de CLI da AWS. É preferível fazer isso do que armazenar chaves de acesso na instância do EC2. Para atribuir uma função da AWS a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, crie um perfil de instância para ser anexado à instância. Um perfil de instância contém a função e permite que programas que estão em execução na instância do EC2 obtenham credenciais temporárias. Para obter mais informações, consulte Usar uma função do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Controle de acesso

Você pode ter credenciais válidas para autenticar suas solicitações, mas, a menos que tenha permissões, não pode criar nem acessar os recursos do Cloud Directory. Por exemplo, você deve ter permissões para criar um Amazon Cloud Directory.

As seções a seguir descrevem como gerenciar permissões para o Cloud Directory. Recomendamos que você leia a visão geral primeiro.

- Visão geral do gerenciamento de permissões de acesso aos recursos do Cloud Directory
- Usar políticas baseadas em identidade (políticas do IAM) para o Cloud Directory
- Permissões da Amazon Cloud Directory: Referência de ações, recursos e condições

Visão geral do gerenciamento de permissões de acesso aos recursos do Cloud Directory

Todo recurso da AWS é de propriedade de uma conta da AWS, e as permissões para criar ou acessar os recursos são regidas por políticas de permissões. Um administrador da conta pode anexar políticas de permissões a identidades do IAM; (ou seja, usuários, grupos e funções) e alguns serviços (como o AWS Lambda) também permitem que se atribuam políticas de permissões aos recursos.

> **Note**
>
> Um administrador da conta (ou usuário administrador) é um usuário com privilégios de administrador. Para obter mais informações, consulte Melhores práticas do IAM no Guia do usuário do IAM.

Ao conceder permissões, você decide quem recebe as permissões, os recursos relacionados às permissões concedidas e as ações específicas que deseja permitir nesses recursos.

Tópicos

- Recursos e operações do Cloud Directory
- Entender a propriedade de recursos

- Gerenciamento do acesso aos recursos
- Especificação de elementos de política: Ações, efeitos, recursos e principais
- Especificação de condições em uma política

Recursos e operações do Cloud Directory

No Cloud Directory, os recursos principais são diretórios e esquemas. Esses recursos têm nomes de recurso da Amazon (ARNs) exclusivos associados, conforme mostrado na tabela a seguir.

Tipo de recurso	Formato de Nome de região da Amazon (ARN)
Diretório	`arn:aws:clouddirectory: region:account-id :directory/ directory-id`
Esquema	`arn:aws:clouddirectory: region:account-id :schema/schema-state /schema-name`

Para obter mais informações sobre estados de esquema e ARNs, consulteExemplos de ARN donoReferência da API Amazon Cloud Directory.

O Cloud Directory fornece um conjunto de operações para trabalhar com os recursos adequados. Para obter uma lista das operações disponíveis, consulteAções da Amazon Cloud DirectoryouAções do Directory Service.

Entender a propriedade de recursos

Um proprietário do recurso é a conta da AWS que criou um recurso. Isto é, o proprietário do recurso é a conta da AWS da entidade principal (a conta-raiz, um usuário do IAM ou uma função do IAM) que autentica a solicitação que cria o recurso. Os exemplos a seguir ilustram como isso funciona:

- Se você usar as credenciais da conta raiz de sua conta da AWS para criar um recurso do Cloud Directory, como um diretório, sua conta da AWS será a proprietária desse recurso.
- Se você criar um usuário do IAM em sua conta da AWS e conceder permissões para criar recursos do Cloud Directory a esse usuário, o usuário também poderá criar recursos do Cloud Directory. No entanto, a sua conta da AWS, à qual o usuário pertence, é a proprietária dos recursos do .
- Se você criar uma função do IAM em sua conta da AWS com permissões para criar recursos do Cloud Directory, qualquer pessoa que puder assumir essa função poderá criar recursos do Cloud

Directory. Sua conta da AWS, à qual a função pertence, é a proprietária dos recursos do Cloud Directory.

Gerenciamento do acesso aos recursos

A política de permissões descreve quem tem acesso a quê. A seção a seguir explica as opções disponíveis para a criação das políticas de permissões.

> **Note**
>
> Esta seção discute o uso do IAM no contexto do Cloud Directory. Não são fornecidas informações detalhadas sobre o serviço IAM. Para concluir a documentação do IAM, consulteO que é o IAM?noGuia do usuário do IAM. Para obter mais informações sobre a sintaxe as descrições da política do IAMReferência de política do AWS IAMnoGuia do usuário do IAM.

Políticas anexadas a uma identidade do IAM são chamadas deBaseado em identidade doPolíticas (políticas do IAM) e políticas anexadas a um recurso são chamadas deBaseado em recursos doPolíticas do. Cloud Directory oferece suporte apenas às políticas baseadas em identidade (políticas do IAM).

Tópicos

- Políticas baseadas em identidade (políticas do IAM)
- Políticas com base em recurso

Políticas baseadas em identidade (políticas do IAM)

Você pode anexar as políticas a identidades do IAM. Por exemplo, você pode fazer o seguinte:

- Anexar uma política de permissões a um usuário ou grupo na sua conta do— um administrador da conta pode usar uma política de permissões associada a determinado usuário para conceder permissões para que esse usuário crie um recurso do Cloud Directory, como um novo diretório.
- Anexar uma política de permissões a uma função (conceder permissões entre contas)— você pode associar uma política de permissões baseada em identidade a uma função do IAM para conceder permissões entre contas. Por exemplo, o administrador na Conta A pode criar uma

função para conceder permissões entre contas a outra conta da AWS (por exemplo, Conta B) ou um serviço da AWS da seguinte forma:

1. Um administrador da Conta A cria uma função do IAM e anexa uma política de permissões à função que concede permissões em recursos da Conta A.
2. Um administrador da Conta A anexa uma política de confiança à função identificando a Conta B como a principal, que pode assumir a função.
3. A seguir, o administrador da Conta B pode delegar permissões para assumir a função para todos os usuários na Conta B. Isso permite que os usuários na Conta B criem ou acessem recursos na Conta A. O principal na política de confiança também poderá ser um principal do serviço da AWS, se você quiser conceder a um serviço da AWS permissões para assumir a função.

Para obter mais informações sobre o uso do IAM para delegar permissões, consulteGerenciamento de acessonoGuia do usuário do IAM.

A seguinte política de permissões concede permissões a um usuário para executar todas as ações que começam com `Create`. Essas ações mostram informações sobre um recurso do Cloud Directory, como um diretório ou um esquema. Observe que o caractere curinga (*) no`Resource`O elemento indica que as ações são permitidas para todos os recursos do Cloud Directory que pertencem à conta.

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

Para mais informações sobre como usar políticas baseadas em identidade com o Cloud Directory, consulteUsar políticas baseadas em identidade (políticas do IAM) para o Cloud Directory. Para obter mais informações sobre usuários, grupos, funções e permissões, consulte Identidades (usuários, grupos e funções) no Guia do usuário do IAM.

Políticas com base em recurso

Outros serviços, como Amazon S3, também dão suporte a políticas de permissões baseadas em recursos. Por exemplo: você pode anexar uma política a um bucket do S3 para gerenciar permissões de acesso a esse bucket. Cloud Directory não é compatível com as políticas baseadas em recursos.

Especificação de elementos de política: Ações, efeitos, recursos e principais

Para cada recurso do Cloud Directory (consulteRecursos e operações do Cloud Directory), o serviço define um conjunto de operações da API. Para obter uma lista das operações da API disponíveis, consulteAções da Amazon Cloud DirectoryouAções do Directory Service. Para conceder permissões a essas operações de API, o Cloud Directory define um conjunto de ações que podem ser especificados em uma política. Observe que a execução de uma operação de API pode exigir permissões para mais de uma ação.

Estes são os elementos de política básicos:

- Recurso – Em uma política, você usa um Amazon Resource Name (ARN – Nome de recurso da Amazon) para identificar o recurso a que a política se aplica. Para os recursos do Cloud Directory, você sempre usa o caractere curinga (*) nas políticas do IAM. Para obter mais informações, consulte Recursos e operações do Cloud Directory.
- Ação: você usa palavras-chave de ação para identificar operações de recursos que você deseja permitir ou negar. Por exemplo, as receitas`clouddirectory:GetDirectory`Permissões permitem que as permissões do usuário executem o Cloud Directory`GetDirectory`operação.
- Efeito— Você especifica o efeito quando o usuário solicita a ação específica — que pode ser permitir ou negar. Se você não conceder (permitir) explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar explicitamente o acesso a um recurso, o que pode fazer para ter a certeza de que um usuário não consiga acessá-lo, mesmo que uma política diferente conceda acesso.
- Principal: em políticas baseadas em identidade (políticas do IAM), o usuário ao qual a política é anexada é implicitamente o principal. Para as políticas baseadas em recursos, você especifica quais usuários, contas, serviços ou outras entidades deseja que recebam permissões (aplica-se somente a políticas baseadas em recursos). Cloud Directory não é compatível com as políticas baseadas em recursos.

Para saber mais sobre a sintaxe e as descrições da política do IAM, consulteReferência de política do AWS IAMnoGuia do usuário do IAM.

Para obter uma tabela que mostra todas as ações da API do Amazon Cloud Directory e os recursos a que elas se aplicam, consultePermissões da Amazon Cloud Directory: Referência de ações, recursos e condições.

Especificação de condições em uma política

Ao conceder permissões, você pode usar a linguagem da política de acesso para especificar as condições quando uma política deve entrar em vigor. Por exemplo, convém que uma política só seja aplicada após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte Condição no Guia do usuário do IAM.

Para expressar condições, você usa chaves de condição predefinidas. Não há chaves de condição específicas do Cloud Directory. No entanto, existem chaves de condição em toda a AWS que você pode usar conforme apropriado. Para obter uma lista completa das chaves de toda a AWS, consulteChaves de condição globais disponíveisnoGuia do usuário do IAM.

Usar políticas baseadas em identidade (políticas do IAM) para o Cloud Directory

Este tópico fornece exemplos de políticas baseadas em identidade em que um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções).

> **Important**
>
> Recomendamos analisar primeiro os tópicos introdutórios que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos recursos do Cloud Directory. Para obter mais informações, consulte Visão geral do gerenciamento de permissões de acesso aos recursos do Cloud Directory.

As seções neste tópico abrangem o seguinte:

- Permissões necessárias para usar o AWS Directory Service Console
- Políticas gerenciadas (predefinidas) pela AWS para o Amazon Cloud Directory

Permissões necessárias para usar o AWS Directory Service Console

Para que um usuário trabalhe com o console do AWS Directory Service, esse usuário deve ter as permissões listadas na política acima ou as permissões concedidas pela função de acesso total do Directory Service ou pela função de somente leitura do Directory Service, descritas em.Políticas gerenciadas (predefinidas) pela AWS para o Amazon Cloud Directory.

Se você criar uma política do IAM que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para os usuários com essa política do IAM.

Políticas gerenciadas (predefinidas) pela AWS para o Amazon Cloud Directory

A AWS resolve muitos casos de uso comuns fornecendo políticas do IAM autônomas criadas e administradas pela AWS. As políticas gerenciadas concedem permissões necessárias para casos de uso comuns, de maneira que você possa evitar a necessidade de investigar quais permissões são necessárias. Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

As seguintes políticas gerenciadas pela AWS, que você pode associar a usuários na sua conta, são específicas ao Amazon Cloud Directory:

- AmazonCloudDirectoryReadOnlyAccess— concede acesso de somente leitura a um usuário ou grupo a todos os recursos do Amazon Cloud Directory. Para obter mais informações, consulte a página Policies (Políticas) no Console de Gerenciamento da AWS.
- AmazonCloudDirectoryFullAccess— concede acesso total a um usuário ou grupo ao Amazon Cloud Directory. Para obter mais informações, consulte a página Policies (Políticas) no Console de Gerenciamento da AWS.

Além disso, há outras políticas gerenciadas da AWS que são adequadas para uso com outras funções do IAM. Essas políticas são atribuídas às funções associadas aos usuários em seu Amazon Cloud Directory e são necessárias para que esses usuários tenham acesso a outros recursos da AWS, como o Amazon EC2.

Você também pode criar políticas personalizadas do IAM que permitem que os usuários acessem os recursos e as ações necessários da API do . Você pode anexar essas políticas personalizadas a usuários ou grupos do IAM que exijam essas permissões.

Permissões da Amazon Cloud Directory: Referência de ações, recursos e condições

Ao configurar o Controle de acesso e escrever políticas de permissões que podem ser anexadas a uma identidade do IAM (políticas baseadas em identidade), você pode usar a tabela a seguir como referência. O A lista inclui oCada operação de API do Amazon Cloud Directory, as ações correspondentes às quais você pode conceder permissões para executar a ação, o recurso da AWS ao qual você pode conceder as permissões. Especifique as ações no campo `Action` da política e o valor do recurso no campo `Resource` da política.

Você pode usar as chaves de condição em toda a AWS nas suas políticas de Amazon Cloud Directory para expressar condições. Para obter uma lista completa das chaves de toda a AWS, consulteChaves de condição globais disponíveisnoGuia do usuário do IAM.

> **Note**
>
> Para especificar uma ação, use o prefixo `clouddirectory:` seguido do nome da operação da API (por exemplo, `clouddirectory:CreateDirectory`).

Registrar em log e monitorar no Amazon Cloud Directory

Como uma prática recomendada, você deve monitorar seu diretório para garantir que as alterações sejam registradas. Isso ajuda a garantir que qualquer alteração inesperada possa ser investigada e alterações indesejadas possam ser restabelecidas. Atualmente, o Amazon Cloud Directory oferece suporte ao AWS CloudTrail, que você pode usar para monitorar seu diretório e qualquer atividade associada.

Para obter mais informações, consulteRegistrar chamadas de API Cloud Directory com o CloudTrail.

Validação de conformidade para o Amazon Cloud Directory

Auditores terceiros avaliam a segurança e a conformidade do Amazon Cloud Directory como parte de vários programas de conformidade da AWS. Isso inclui ISO, SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista dos serviços da AWS no escopo de programas de conformidade específicos, consulte Serviços da AWS no escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

É possível fazer download de relatórios de auditoria externa usando o AWS Artifact. Para obter mais informações, consulte Download de relatórios no AWS Artifact.

Sua responsabilidade de conformidade ao usar o Cloud Directory é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias de início rápido de segurança e conformidade— esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para a implantação de ambientes de linha de base concentrados em conformidade e segurança na AWS.
- Whitepaper Arquitetura para segurança e conformidade com HIPAA – esse whitepaper descreve como as empresas podem usar a AWS para criar aplicativos em conformidade com a HIPAA.
- Recursos de conformidade da AWS: esta coleção de guias e pastas de trabalho pode ser aplicada ao seu setor e local.
- AWS Config: este serviço da AWS avalia até que ponto suas configurações de recursos estão em conformidade com as práticas internas, e com as diretrizes e as normas do setor.
- Security Hub da AWS – esse serviço da AWS fornece uma visão abrangente do estado da segurança na AWS que ajuda você a verificar sua conformidade com padrões e melhores práticas de segurança do setor.

Resiliência no Amazon Cloud Directory

A infraestrutura global da AWS é criada com base em regiões e zonas de disponibilidade da AWS. As regiões da AWS fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, altas taxas de transferência e redes altamente redundantes. O Cloud Directory foi criado com base nesses princípios e está disponível em várias regiões da AWS, que estão fisicamente isoladas umas das outras. Em cada região, o serviço é ainda suportado por pelo menos três zonas de disponibilidade, minimizando o tempo de inatividade do serviço devido à indisponibilidade de qualquer zona de disponibilidade única.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Segurança de infraestrutura no Amazon Cloud Directory

Como serviço gerenciado, o Amazon Cloud Directory é protegido pelos procedimentos de segurança da rede global da AWS descritos naAmazon Web Services Visão geral dos processos de segurançaWhitepaper branco.

Você usa as chamadas de API publicadas da AWS para acessar o Cloud Directory por meio da rede. Os clientes devem oferecer suporte ao Transport Layer Security (TLS). Recomendamos TLS 1.2 ou posterior. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos.

Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para obter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Suporte de transação

Com o Amazon Cloud Directory, sempre é necessário adicionar novos objetos ou adicionar relações entre objetos novos e objetos existentes para refletir as alterações em uma hierarquia de mundo real. As operações em lotes podem tornar tarefas de diretório como essas mais fáceis de gerenciar fornecendo os seguintes benefícios:

- As operações em lotes podem minimizar o número de viagens de ida e volta para gravar e ler objetos no diretório e vice-versa, melhorando o desempenho total do aplicativo.
- A gravação em lotes fornece a semântica de transação equivalente ao banco de dados SQL. Todas as operações concluídas com êxito, ou se qualquer operação tiver uma falha, nenhuma delas será aplicada.
- Usando a referência de lote você pode criar um objeto e usar uma referência ao novo objeto para ação adicional, como a adicioná-lo a uma relação, reduzindo a sobrecarga de usar uma operação de leitura antes de uma operação de gravação.

BatchWrite

Use operações BatchWrite para executar várias operações de gravação em um diretório. Todas as operações de gravação em lotes são executadas sequencialmente. Elas funcionam de maneira semelhante às transações de banco de dados SQL. Se uma da operações da gravação em lotes falhar, toda a gravação em lotes não terá nenhum efeito no diretório. Se uma gravação em lotes falhar, ocorrerá uma exceção de gravação em lotes. A exceção contém o índice da operação que falhou junto com o tipo e a mensagem da exceção. Essas informações podem ajudá-lo a identificar a causa raiz da falha.

As seguintes operações da API são compatíveis como parte da gravação em lotes:

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

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

Nome da referência de lote

Os nomes de referência de lote são compatíveis apenas para gravações em lotes quando você precisa fazer referência a um objeto como parte da operação em lotes intermediária. Por exemplo, suponha que, como parte de uma determinada gravação em lotes, 10 objetos diferentes estão sendo desanexados e anexados a uma outra parte do diretório. Sem a referência de lote, você precisaria ler todas as referências aos 10 objetos e fornecê-las como entrada durante a nova anexação, como parte da gravação em lotes. Você pode usar uma referência de lote para identificar o recurso desanexado durante a anexação. Uma referência de lote pode ser qualquer string normal prefixada com o símbolo de jogo da velha/hashtag (#).

Por exemplo, no exemplo de código a seguir, um objeto com o nome de link `"this-is-a-typo"` está sendo desanexado da raiz com um nome de referência de lote `"ref"`. Mais tarde, o mesmo objeto é anexado à raiz com o nome de link como `"correct-link-name"`. O objeto é identificado com o conjunto de referências filho para a referência de lote. Sem a referência de lote, você precisaria inicialmente obter o `objectIdentifier` que está sendo desanexado e fornecê-lo na referência filho durante a anexação. Você pode usar um nome da referência de lote para evitar essa leitura extra.

```
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

Use operações BatchRead para executar várias operações de leitura em um diretório. Por exemplo, no código de exemplo a seguir, os filhos de objetos com a referência "`/managers`" estão sendo lidos junto com os atributos de objetos com a referência "`/managers/bob`" em uma única leitura em 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);
```

A BatchRead é compatível com as seguintes operações da API:

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

Limites de operações em lote

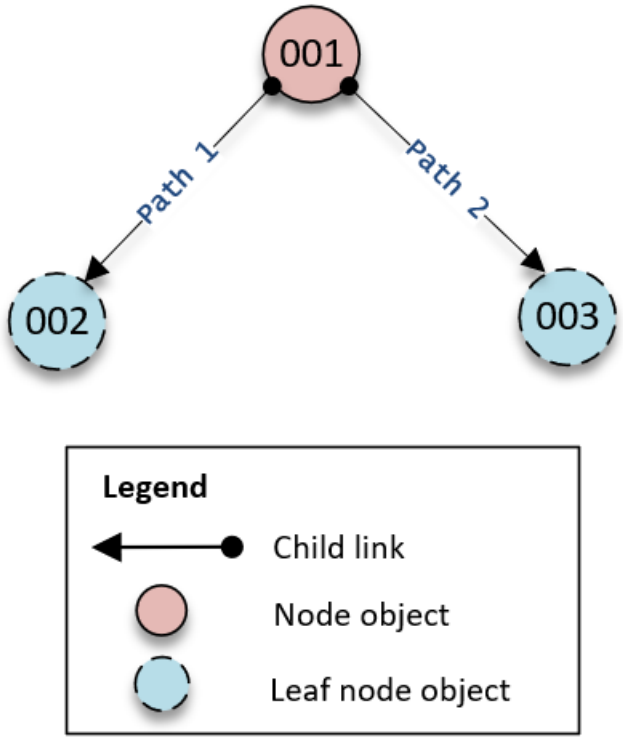
Cada solicitação ao servidor (incluindo solicitações em lotes) tem um número máximo de recursos nos quais é possível operar, independentemente do número de operações na solicitação. Isso

permite que você componha solicitações em lotes com alta flexibilidade desde que se mantenha dentro do número máximo de recursos. Para obter mais informações sobre os números máximos de recursos, consulte Entre os limites do Amazon Cloud Directory.

Os limites são calculados somando as gravações ou leituras de cada única operação no lote. Por exemplo, o limite de operação de leitura atualmente é de 200 objetos por chamada da API. Digamos que você deseja compor um lote que adicione nove chamadas da API ListObjectChildren e que cada chamada exija a leitura de 20 objetos. Como o número total de objetos de leitura (9 x 20 = 180) não excede 200, a operação em lotes teria êxito.

O mesmo conceito se aplica ao cálculo de operações de gravação. Por exemplo, o limite da operação de gravação atualmente é 20. Se você configurar o lote para adicionar duas chamadas da API UpdateObjectAttributes com nove operações de gravação cada uma, isso também teria êxito. Em ambos os casos, se a operação em lotes exceder o limite, a operação falhará e uma `LimitExceededException` será lançada.

A maneira correta de calcular o número de objetos que estão incluídos em um lote é incluir o nó real ou objetos de nó folha (leaf_node). Se estiver usando uma abordagem baseada no caminho para iterar a árvore de diretórios, você também precisa incluir no lote cada caminho que está iterado. Por exemplo, como mostrado a seguir na ilustração de uma árvore de diretórios básica, para ler um valor de atributo para o objeto `003`, a contagem de leitura total de objetos seria três.



O percurso de leituras da árvore funciona da seguinte forma:

1. Leia o objeto `001` para determinar o caminho do objeto `003`

2. Vá para o `Path 2`

3. Leia o objeto `003`

Da mesma forma, para o número de atributos, precisamos contar o número de atributos nos objetos `001` e `003` para garantir que o limite não seja atingido.

Tratamento de exceções

Às vezes, as operações em Batch no Cloud Directory podem falhar. Nesses casos, é importante saber como tratar essas falhas. O método usado para solucionar falhas difere para operações de gravação e para operações de leitura.

Falhas de operações de gravação em lotes

Se uma operação de gravação em lotes falhar, o Cloud Directory falhará a operação em lotes inteira e retornará uma exceção. A exceção contém o índice da operação que falhou junto com o tipo e a mensagem da exceção. Se você vir `RetryableConflictException`, poderá tentar novamente com recuo exponencial. Uma maneira simples de fazer isso é dobrar a quantidade de tempo que você espera a cada vez que obtém uma exceção ou uma falha. Por exemplo, se sua primeira operação de gravação em lotes falhar, aguarde 100 milissegundos e teste a solicitação novamente. Se a segunda solicitação falhar, aguarde 200 milissegundos e tente novamente. Se a terceira solicitação falhar, aguarde 400 milissegundos e tente novamente.

Falhas de operações de leitura em lotes

Se uma operação de leitura em lotes falhar, a resposta conterá uma resposta bem-sucedida ou uma resposta de exceção. Falhas individuais de operações de leitura em lotes não causam falha em uma operação inteira de leitura em lotes — o Cloud Directory retorna êxito ou falha individual para cada operação.

Artigos relacionados do blog do Cloud Directory

- Write and Read Multiple Objects in Amazon Cloud Directory usando operações Batch
- How to Use Batch References in the Amazon Cloud Directory para fazer referência a objetos novos em uma solicitação em Batch

Conformidade da Amazon Cloud Directory

O Amazon Cloud Directory passou por uma auditoria dos padrões a seguir e poderá fazer parte de sua solução quando você precisar obter certificação de conformidade.

FedRAMP	O Amazon Cloud Directory atende aos requisitos de segurança do Federal Risk and Authorization Managemen t Program (FedRAMP) e recebeu uma autoridade provisóri a da Joint Authorization Board (JAB) da FedRAMP para operar (P-ATO) na linha de base moderada da FedRAM Para obter mais informações sobre conformidade com FedRAMP, consulte Conformidade com FedRAMP.
PCI Security Standards Council PARTICIPATING ORGANIZATION™	O Amazon Cloud Directory tem uma Declaração de conformidade com o padrão de segurança de dados (Data Security Standard, DSS) versão 3.2 da Payment Card Industry (PCI) no nível 1 de prestador de serviços. Os clientes que usam os produtos da AWS para armazenar , processar ou transmitir dados de titulares de cartões podem usar o Cloud Directory ao gerenciar sua própria certificação de conformidade com o PCI DSS. Para obter mais informações sobre PCI DSS, incluindo como solicitar uma cópia do pacote de conformidade com PCI da AWS, consulte Nível 1 do PCI DSS.

	A AWS expandiu seu programa de conformidade da lei americana HIPAA (Health Insurance Portability Accountability Act) para incluir o Amazon Cloud Directory como umServiço qualificado pela HIPAA. Se você tiver um acordo de associado comercial (BAA) com a AWS, poderá usar o Cloud Directory para ajudar a criar seus aplicativos compatíveis com a HIPAA. A AWS oferece umaWhitepaper focado em HIPAApara os clientes interessados em saber mais sobre como podem utilizar a AWS para processamento e armazenamento de informações de saúde. Para obter mais informações, consulte Conformidade com a HIPAA.
	O Amazon Cloud Directory concluiu com êxito a certificação de conformidade para a ISO/IEC 27001, a ISO/IEC 27017, a ISO/IEC 27018 e a ISO 9001. Para obter mais informações, consulte ISO 27001, ISO 27017, ISO 27018 e ISO 9001.
	Os relatórios de Controle de Sistema e Organização (System and Organization Control, SOC) são relatórios de exames de terceiros independentes que demonstram como o Amazon Cloud Directory obtém os principais controles e objetivos de conformidade. O objetivo desses relatórios é ajudar você e seus auditores a compreenderem os controles da AWS estabelecidos para oferecer suporte às operações e à conformidade. Para obter mais informações, consulte Conformidade com o SOC.

Responsabilidade compartilhada

A segurança, incluindo a conformidade com HIPAA e PCI, é uma responsabilidade compartilhada. É importante compreender que o status de conformidade do Cloud Directory não se aplica

automaticamente a aplicativos executados na Nuvem AWS. Você deve garantir que seu uso dos serviços da AWS esteja em conformidade com os padrões.

Usando as APIs do Cloud Directory

O Amazon Cloud Directory inclui um conjunto de operações da API que permitem acesso programático aos recursos do Cloud Directory. Você pode usar oGuia de referência da API do Amazon Cloud DirectoryPara saber como fazer solicitações à API do Cloud Directory para criar e gerenciar os vários elementos. Ele também discute os componentes das solicitações, o conteúdo das respostas e como autenticar solicitações.

O Cloud Directory fornece todas as operações da API necessárias que permitem que os desenvolvedores criem novos aplicativos. Fornece as seguintes categorias de chamadas da API:

- Criação, leitura, atualização, exclusão (CRUD) de esquema
- CRUD de faceta
- CRUD de diretórios
- CRUD de objetos (nós, políticas etc.)
- CRUD de definição de índice
- Leitura em lotes, gravação em lotes

Como funciona o faturamento com as APIs do Cloud Directory

O faturamento de chamadas de API varia com base nos tipos específicos de chamadas de API sendo feitas. Há taxas de faturamento específicas para chamadas de API de Leitura eventualmente consistente, de Leitura fortemente consistente e de Gravação. As chamadas de API de metadados são gratuitas.

Operações fortemente consistentes são usadas para consistência de leitura após gravação ao ler um valor. As operações eventualmente consistentes são usadas para recuperar um valor enquanto atualizações são executadas. Com as operações eventualmente consistentes, os resultados recuperados pode não ser os mais precisos, pois o host específico do qual você está lendo o valor ainda está processando atualizações. No entanto, a latência para essas operações de leitura é baixa quando você recupera uma chamada de desempenho.

Ao ler dados do Cloud Directory, você deve especificar uma operação de Leitura eventualmente consistente ou Leitura fortemente consistente. O tipo da leitura tem como base o nível de consistência. Os dois níveis de consistência são EVENTUAL para Leituras eventualmente

consistentes e SERIALIZABLE para Leituras fortemente consistentes. Para obter mais informações, consulte Níveis de consistência.

A tabela a seguir lista todas as APIs do Cloud Directory e como podem afetar o faturamento de sua conta da AWS.

API	Leitura eventualmente consistente [1]	Leitura fortemente consistente [2]	Leitura [3]	Metadados [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

API	Leitura eventualmente consistente [1]	Leitura fortemente consistente [2]	Leitura [3]	Metadados [4]
DeleteFacet				X
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	Leitura eventualmente consistente [1]	Leitura fortemente consistente [2]	Leitura [3]	Metadados [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	Leitura eventualmente consistente [1]	Leitura fortemente consistente [2]	Leitura [3]	Metadados [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	Leitura eventualmente consistente [1]	Leitura fortemente consistente [2]	Leitura [3]	Metadados [4]
PutSchemaFromJson				X
RemoveFacetFromObject			X	
TagResource				X
UntagResource				X
UpdateFacet				X
UpdateLinkAttributes			X	
UpdateObjectAttributes			X	
UpdateSchema				X
UpdateTypedLinkFacet				X
UpgradeAppliedSchema				X
UpgradePublishedSchema				X

[1] As APIs de Leitura eventualmente consistentes são chamadas com o nível de consistência EVENTUAL

[2] As APIs de Leitura fortemente consistentes são chamadas com o nível de consistência SERIALIZABLE

[3] APIs de gravação são faturadas como chamadas da API de gravação

[4] APIs de metadados NÃO são faturadas, mas são categorizadas como chamadas da API de metadados

Para obter informações adicionais sobre faturamento, consulte Definição de preço do Amazon Cloud Directory.

Entre os limites do Amazon Cloud Directory

A seguir estão os limites padrão para o Cloud Directory. Cada limite é por região, a menos que observado em contrário.

Amazon Cloud Directory

Limites do esquema e diretório

Limite/conceito	Quantidade
Número de atributos por faceta (incluindo o necessário)	1000
Número de facetas por objeto	5
Número de índices exclusivos aos quais um objeto é ligado	3
Número de facetas por esquema	30
Número de regras por atributo	5
Número de atributos com valores padrão por faceta	10
Número de atributos necessários por faceta	30
Número de esquemas de desenvolvimento	20
Número de esquemas publicados	20
Número de esquemas aplicados	5
Número de diretórios	100
Número máximo de elementos na página	30
Tamanho máximo de entrada (todas as entradas combinadas)	200 KB

Limite/conceito	Quantidade
Tamanho máximo de resposta (todas as saída combinadas)	1 MB
Limite de tamanho de arquivo JSON do esquema	200 KB
Tamanho do nome da faceta	64 bytes codificados por UTF-8
Tamanho do nome do diretório	64 bytes codificados por UTF-8
Tamanho do nome do esquema	64 bytes codificados por UTF-8

Limites de objeto

Limite/conceito	Quantidade
Número de objetos gravados	20 por chamada de API
Número de objetos lidos	200 por chamada de API
Número de valores de atributos gravados	1000 por chamada de API
Número de valores de atributos lidos	1000 por chamada de API
Profundidade do caminho	15
Tamanho máximo de entrada (todas as entradas combinadas)	200 KB
Tamanho máximo de resposta (todas as saída combinadas)	1 MB
Limite de tamanho da política	10 KB
Número de atributos que podem ser excluídos durante uma exclusão de objeto	30

Limite/conceito	Quantidade
Comprimento do valor agregado para atributos de identidade do link tipado	64 bytes codificados por UTF-8
Tamanho do nome da borda ou do link	64 bytes codificados por UTF-8
Tamanho do valor para atributos indexados	512 bytes codificados por UTF-8
Tamanho do valor para atributos não indexados	2 KB
Número de políticas anexadas a um objeto	4

Limites de operações em lote

Não há limite sobre o número de operações que você chama dentro de um lote. Para obter mais informações, consulte Limites de operações em lote.

Limites que não podem ser modificados

Entre os limites do Amazon Cloud Directory que não podem ser alterados nem aumentados estão:

- Tamanho do nome da faceta
- Tamanho do nome do diretório
- Tamanho do nome do esquema
- Número máximo de elementos na página
- Tamanho do nome da borda ou do link
- Tamanho do valor para atributos indexados

Directory Recursos da nuvem

A tabela a seguir lista os recursos relacionados que serão úteis à medida que você utilizar este serviço.

Cloud Directory Conceitos básicos do	Link
Directory Webinar da nuvem	https://www.youtube.com/watch?v=UANm3DC_IxE
Código Java de exemplo do Directory	https://github.com/aws-samples/AmazonCloudDirectory-sample

Publicações no blog do Amazon Directory	Descrição
How to rapidly develop applications on Amazon Cloud Directory with Managed Schema (Como desenvolver de forma rápida os aplicativos no Amazon Cloud Directory com o esquema gerenciado)	Essa postagem do blog explica a rápida criação de protótipos e desenvolvimento no diretório na nuvem usando o esquema gerenciado. Também inclui código Java de exemplo.
Como pesquisar com mais eficiência no Amazon Cloud Directory	Essa postagem de blog explica sobre a pesquisa mais eficiente com o uso de indexação baseada em facetas. Também inclui código Java de exemplo.
Como aplicar com facilidade alterações de esquema de diretório na nuvem da Amazon com atualizações de esquemas no local	Essa postagem de blog explica como executar uma atualização de esquema no local para qualquer Cloud Directory (em execução) operacional. Também inclui código Java de exemplo.
Gravar e ler vários objetos no Amazon Cloud Directory usando operações Batch	Explica sobre o uso de lotes de leitura e gravação. Também inclui código Java de exemplo.

Publicações no blog do Amazon Directory	Descrição
Como usar referências de Batch no Amazon Cloud Directory para fazer referência a objetos novos em uma solicitação em Batch	Explica sobre o uso de referência de lote. Também inclui código Java de exemplo.
Atualização Cloud Directory — Support para links digitados	Explica sobre a criação e a pesquisa de relacionamentos em hierarquias no Diretório na nuvem usando links digitados. Também inclui código Java de exemplo.
Nova API do diretório da nuvem facilita a consulta a dados ao longo de várias dimensões	Explica como consultar dados em várias dimensões com uma única chamada usando a API `ListObjectParentPaths` .
How to Create an Organizational Chart with Separate Hierarchies by Using Directory	Explica como criar um esquema e um diretório com código Java de exemplo.
Amazon Cloud Directory — Um diretório nativo da nuvem para dados hierárquicos	Descreve o lançamento do Cloud Directory como um novo serviço da AWS.

Directory Documentação da nuvem	Link
Guia do desenvolvedor do Directory	https://docs.aws.amazon.com/clouddirectory/latest/developerguide/what_is_cloud_directory.html
Cloud Directory Reference	https://docs.aws.amazon.com/clouddirectory/latest/APIReference/welcome.html
Diretório Limites da nuvem	https://docs.aws.amazon.com/clouddirectory/latest/developerguide/limits.html

Cloud Directory Outro	Link
Directory Info do produto da nuvem	https://aws.amazon.com/cloud-directory/
Directory Preços da nuvem	https://aws.amazon.com/cloud-directory/pricing/

Histórico do documento

A tabela a seguir descreve as alterações na documentação desde a última versão doGuia do desenvolvedor do Amazon Cloud Directory.

- Última atualização de documentação: 21 de junho de 2018

update-history-change	update-history-description	update-history-date
Novo esquema gerenciado	Adição de conteúdo para opção de esquema gerenciado.	21 de junho de 2018
Conteúdo migrado para este guia	Todo o conteúdo existente do Cloud Directory foi transferido do Guia do administrador do AWS Directory Service para esse novo Guia do desenvolvedor do Amazon Cloud Directory para mapear diretamente às necessidades do cliente.	20 de junho de 2018
Atualizações de esquema no local	Conteúdo adicionado para aplicar alterações de esquema em diretórios do Amazon Cloud Directory com atualizações de esquema no local.	6 de dezembro de 2017
Indexação baseada em facetas	Adicionada uma seção ao índice baseado em facetas.	9 de agosto de 2017
Lotes	Informações atualizadas sobre lotes para o Amazon Cloud Directory.	26 de julho de 2017

Conformidade	Informações adicionadas sobre conformidade com HIPAA e PCI.	14 de julho de 2017
Links digitados	Adicionado novo conteúdo de links digitados para o Amazon Cloud Directory.	31 de maio de 2017
Serviço da Amazon Cloud Directory	Novo tipo de diretório apresentado.	26 de janeiro de 2017

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o Glossário da AWS na Referência geral da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.