aws

開發人員指南

Amazon Cloud Directory

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

Amazon Cloud Directory: 開發人員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 Amazon Cloud Directory？

Amazon Cloud Directory 是 AWS 中具高可用性的多租戶目錄架構存放區。這些目錄可視應用程式需要，自動擴展至數億個物件。這可讓操作人員專注於開發及部署推動業務的應用程式，而不是管理目錄基礎設施。不同於傳統目錄系統，Cloud Directory 不限於將目錄物件組織成單一固定階層。

有了 Cloud Directory，您可以將目錄物件組織成多個階層，來支援目錄資訊之間的多個組織樞紐和關係。例如，使用者目錄可提供以報告結構、位置和專案關係為基礎的階層檢視。同樣地，裝置目錄可有以其製造商、目前擁有者和實體位置為基礎的多個階層檢視。

Cloud Directory 是以圖表架構目錄存放區為核心，為開發人員提供重要的建置區塊基礎。透過 Cloud Directory，開發人員可以執行下列作業：

- 輕鬆地建立目錄式的應用程式，無須擔心部署、全球規模、可用性和效能
- 建立應用程式，提供使用者和群組管理、許可或政策管理、裝置登錄、客戶管理、通訊錄，以及應用程式或產品目錄
- 定義新的目錄物件或擴展現有的類型，以滿足其應用程式需求，並減少需要撰寫的程式碼
- 降低在 Cloud Directory 上應用程式分層的複雜度
- 管理結構描述資訊隨時間的演變，以確保消費者的未來相容性

Cloud Directory 包含一組 API 操作，可存取存放在您 Cloud Directory 目錄中的各種物件和政策。如需可用操作的清單，請參閱Amazon Cloud Directory API 操作。如需執行每個 API 動作所需之操作和許可的清單，請參閱「Amazon Cloud Directory 許可：動作、資源和條件參考」。

如需支援的 Cloud Directory 區域清單，請參閱AWS 區域與端點文件中)。如需其他資源，請參閱「Cloud Directory 資源」。

什麼 Cloud Directory 不是

Cloud Directory 不適用於要管理或遷移目錄基礎設施的 IT 管理員。

入門

在此入門練習中，您會建立一個結構描述。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. 雖然沒有要求，但我們建議您在開始使用主控台之前檢閱了解重要的 Cloud Directory 概念，以便熟悉核心功能和術語。

主題

- 建立結構描述
- 建立 Amazon Cloud Directory
- 使用 Cloud Directory 介面 VPC 端點

建立結構描述

Amazon Cloud Directory 支援上傳相容的 JSON 檔案以建立結構描述。若要建立新的結構描述，您可以從頭開始建立自己的 JSON 檔案，或下載列於主控台中的其中一個現有結構描述。然後將它上傳做為自訂結構描述。如需詳細資訊，請參閱 自訂結構描述。

您也可以使用 Cloud Directory API 建立、刪除、下載、列出、發佈、更新和升級結構描述。如需結構描述 API 操作的詳細資訊，請參閱Amazon Cloud Directory API 參考指南。

根據您慣用的方法，選擇下列其中一個程序。

建立自訂結構描述

1. 在 中AWS Directory Service 主控台導覽窗格, 在Cloud Directory中，選擇Schemas。
2. 建立一個 JSON 檔案，其中包含您所有的新結構描述定義。如需如何格式化 JSON 檔案的詳細資訊，請參閱「JSON 結構描述格式」。
3. 在主控台，選擇上傳新的結構描述。
4. 在 中上傳新的結構描述對話方塊中，輸入結構描述的名稱。
5. 選擇選擇檔案，選取您剛建立的新 JSON 檔案，然後選擇開啟。
6. 選擇 Upload (上傳)。這會在您的結構描述程式庫中新增一個結構描述，並將其設為開發狀態。如需結構描述狀態的詳細資訊，請參閱「結構描述生命週期」。

在主控台中根據現有的結構描述建立自訂結構描述

1. 在 中AWS Directory Service 主控台導覽窗格, 在Cloud Directory中，選擇Schemas。
2. 在列出結構描述的表格中，選取您要複製的結構描述附近的選項。
3. 選擇 Actions (動作)。
4. 選擇下載結構描述。
5. 將 JSON 檔案重新命名並視需要進行編輯，然後儲存檔案。如需如何格式化 JSON 檔案的詳細資訊，請參閱「JSON 結構描述格式」。
6. 在主控台，選擇上傳新的結構描述，選取您剛編輯的 JSON 檔案，然後選擇開啟。

   這會在您的結構描述程式庫中新增一個結構描述，並將其設為開發狀態。如需結構描述狀態的詳細資訊，請參閱「結構描述生命週期」。

建立 Amazon Cloud Directory

在 Amazon Cloud Directory 中建立目錄之前，AWS Directory Service 需要您先對其套用結構描述。您無法建立沒有結構描述的目錄，而且一個目錄通常會套用一個結構描述。不過，您可以使用 Cloud Directory API 操作將其他結構描述套用至目錄。如需詳細資訊，請參閱「」ApplySchema中的Amazon Cloud Directory API 參考指南。

建立雲端 Directory

1. 在 中AWS Directory Service 主控台導覽窗格, 在Cloud Directory中，選擇目錄。
2. 選擇設定 Cloud Directory。
3. 根據選擇要套用至新目錄的綱要，輸入您目錄的易記名稱，例如`User Repository`，然後選擇下列其中一種選項：

   - 受管結構描述
   - 範例結構描述
   - 自訂結構描述

   範例結構描述和自訂結構描述放置在開發狀態，依預設。如需結構描述狀態的詳細資訊，請參閱「結構描述生命週期」。您必須將結構描述轉換成 Published (已發佈) 狀態，才能套用至目錄。若要使用主控台成功發佈範例結構描述，您必須具有下列動作的許可：

   - `clouddirectory:Get*`

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

  由於範例結構描述是由 AWS 提供的唯讀範本，因此無法直接發佈。反之，當您選擇根據範例結構描述建立目錄時，主控台會建立所選範例結構描述的暫存副本，並將它設定為開發狀態。接著會建立開發結構描述的副本，並將它設定為 Published (已發佈) 狀態。一旦發佈，即會刪除開發結構描述，因此 `DeleteSchema` 動作對發佈範例結構描述而言是必要的。

4. 選擇下一步。
5. 檢閱目錄資訊，並進行必要的變更。若資訊無誤，請選擇 Create (建立)。

使用 Cloud Directory 介面 VPC 端點

如果您使用 Amazon Virtual Private Cloud (Amazon VPC) 來託管 AWS 資源，您可以在 VPC 和 Cloud Directory 之間建立私有連線。您可以使用此連線來啟用 Cloud Directory 不用透過公有網際網路在 VPC 與您的資源進行通訊。

Amazon VPC 是一項 AWS 服務，您可用來在自己定義的虛擬網路中啟動 AWS 資源。您可利用 VPC 來控制您的網路設定，例如 IP 地址範圍、子網路、路由表和網路閘道。若要將您的 VPC 連接到 Cloud Directory，請定義界面 VPC 端點(位於 Cloud Directory))。端點能為 Cloud Directory 提供可靠、可擴展性的連線，無須使用網際網路閘道、網路位址轉譯 (NAT) 執行個體或 VPN 連線。如需詳細資訊，請參閱「」什麼是 Amazon VPC？中的Amazon VPC 使用者指南。

界面 VPC 端點由 AWS PrivateLink 提供，一種 AWS 技術可使用 elastic network interface 搭配私有 IP 地址，來在 AWS 服務之間進行私有通訊。如需詳細資訊，請參閱「」AWS 服務的 AWS PrivateLink。

以下步驟適用於 Amazon VPC 的使用者。如需詳細資訊，請參閱「」Amazon VPC 入門中的Amazon VPC 使用者指南。

Availability

Cloud Directory 目前在下列區域支援 VPC 端點：

- 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 (美國西部)

為 Cloud Directory 建立 VPC

若要搭配您的 VPC 開始使用 Cloud Directory，請使用 Amazon VPC 主控台建立 Cloud Directory 的界面 VPC 端點。如需詳細資訊，請參閱建立界面端點。

- 適用於服務目錄中，選擇AWS 服務。
- 在 Service Name (服務名稱) 中，選擇 **`com.amazonaws.region.clouddirectory`**。這會為 Cloud Directory 作業建立 VPC 端點。

如需一般資訊，請參閱「什麼是 Amazon VPC？」中的Amazon VPC 使用者指南。

控制對 Cloud Directory VPC 端點的存取

當您建立或修改端點時，VPC 端點原則是您連線至端點的 IAM 資源原則。如果您未在建立端點時連接政策，我們會以預設政策連接以允許完整存取服務。端點政策不會覆寫或取代 IAM 使用者政策或服務特定的政策。這個另行區分的政策會控制從端點到所指定之服務的存取。

端點政策必須以 JSON 格式撰寫。如需詳細資訊，請參閱「」使用 VPC 端點控制對服務的存取中的Amazon VPC 使用者指南。

以下是 Cloud Directory 端點政策的範例。此政策可讓使用者透過 VPC 連接到 Cloud Directory 來列出目錄，而且會防止使用者執行其他 Cloud Directory 動作。

```
{
  "Statement": [
```

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

修改 Cloud Directory 的 VPC 端點政策

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中選擇 Endpoints (端點)。
3. 如果您尚未建立 Cloud Directory 的端點，請選擇建立端點。然後選取**com.amazonaws.region.clouddirectory**，然後選擇建立端點。
4. 選取**com.amazonaws.region.clouddirectory**端點，然後選擇政策索引標籤 (位於螢幕下半部)。
5. 選擇 Edit Policy (編輯政策)，並對政策做出變更。

如需詳細資訊，請參閱「」使用 VPC 端點控制對服務的存取中的Amazon VPC 使用者指南。

了解重要的 Cloud Directory 概念

Amazon Cloud Directory 是目錄架構資料存放區，可建立結構描述導向格式的多種物件類型。

主題

Schema

結構描述是面向集合，用來定義可在目錄中建立的物件及其組織方式。結構描述也會行使資料完整性和互通性。單一結構描述一次可套用至多個目錄。如需詳細資訊，請參閱 Schemas。

Facets

面向是結構描述中定義的屬性、限制條件和連結集合。綜合來看，面向會定義目錄中的物件。例如，Person 和 Device 可以是定義企業員工與多部裝置之關聯的面向。如需詳細資訊，請參閱 Facets。

受管結構描述

提供的結構描述可讓您更輕鬆快速開發和維護應用程式。如需詳細資訊，請參閱 受管結構描述。

範例結構描述

AWS Directory Service 主控台中預設會提供一組範例結構描述。例如，Person、Organization 和 Device 都是範例結構描述。如需詳細資訊，請參閱 範例結構描述。

自訂結構描述

由使用者定義的一或多個結構描述，可能是從「結構描述」區段或在 AWS Directory Service 主控台的 Cloud Directory 建立過程中上傳，或是透過 API 呼叫建立。

Directory

目錄是結構描述架構的資料存放區，其中包含由多階層結構組織的特定物件類型 (如需詳細資訊，請參閱「目錄結構」)。例如，使用者目錄可提供以報告結構、位置和專案關係為基礎的階層檢視。同樣地，裝置目錄可有以其製造商、目前擁有者和實體位置為基礎的多個階層檢視。

目錄定義資料存放區的邏輯邊界，藉此與服務中的其他目錄完全隔離。此外也定義個別請求的邊界。單一交易或查詢會在單一目錄內容中執行。您無法建立沒有結構描述的目錄，而且一個目錄通常會套用一個結構描述。不過，您可以使用 Cloud Directory API 操作將其他結構描述套用至目錄。如需詳細資訊，請參閱「」ApplySchema中的Amazon Cloud Directory API 參考指南。

Objects

物件是目錄中的結構化資料實體。目錄中的物件旨在擷取實體 (Physical) 或邏輯實體 (Entity) 相關的中繼資料 (或屬性)，通常用於資訊探索及政策行使等目的。例如，使用者、裝置、應用程式、AWS 帳戶、EC2 執行個體和 Amazon S3 儲存貯體在目錄中都會以不同物件類型表示。

物件的結構和類型資訊會以面向集合表示。您可以使用 `Path` 或 `ObjectIdentifier` 來存取物件。物件也可以包含屬性，即使用者定義的中繼資料單位。例如，使用者物件可以包含名為 email-address 的屬性。屬性一律會與物件產生關聯。

Policies

政策是專門用來存放許可或功能的物件類型。政策提供 `LookupPolicy` API 動作。查詢政策動作可參考任何物件做為其起始輸入。接著會在目錄中一路向上到根目錄。此動作會收集在到達根目錄的每個路徑上，所遇到的任何政策物件。Cloud Directory 完全不會解譯任何政策。反之，會是由 Cloud Directory 使用者透過其專屬的商業邏輯來解譯政策。

例如，假設有一個存放員工資訊的系統。而員工會依工作職能分組。我們想要為人力資源群組成員與會計群組成員建立不同的許可。人力資源群組成員將能夠存取薪資資訊，而會計群組成員將能夠存取總帳資訊。為了建立這些許可，我們將政策物件連接到每個群組。需要評估使用者的許可時，我們可以對該使用者的物件使用 `LookupPolicy` API 動作。所以此`LookupPolicy`API 動作會從指定政策的物件一路向上到根目錄。途中會在每個節點停下並檢查是否有任何連接政策，然後傳回這些政策。

政策連接

您可以透過兩種方式將政策連接到其他物件：一般父子連接與特殊政策連接。使用一般父子連接，政策可以連接到父節點。這提供了一個簡單的機制，經常有助於尋找資料目錄中的政策。政策不能有子項。`LookupPolicy` API 呼叫期間不會傳回透過父子連接所連接的政策。

政策物件也可透過政策連接來連接到其他物件。您可以使用 AttachPolicy 和 DetachPolicy API 動作來管理這些政策連接。政策連接可讓您在使用 LookupPolicy API 時找到政策節點。

政策結構描述規格

若要開始使用政策，您必須先將面向新增至結構描述以支援建立政策。為了達成此目標，請建立面向，並將面向的 objectType 設定為 POLICY。建立使用 POLICY 類型面向的物件可確保物件具有政策功能。

政策面向會繼承兩個屬性，以及您新增至定義的任何屬性：

- policy_type (字串、必要) - 這是您可以提供來區分不同政策使用的識別符。如果您的政策有清楚的分類邏輯，建議您適當地設定政策類型屬性。LookupPolicy API 會傳回連接政策的政策類型 (請參閱「PolicyAttachment」)。這可讓您輕鬆篩選想要尋找的特定政策類型。您也可以使用 policy_type 來決定文件的處理或解譯方式。
- policy_document (二進位、必要) - 您可以在此屬性中存放應用程式特定資料，例如與政策相關聯的許可授予。如果您想要，也可以將應用程式相關資料存放在面向的一般屬性中。

政策 API 概觀

有各種不同的專用 API 動作可搭配政策使用。如需可用操作的清單，請參閱Amazon Cloud Directory 操作。

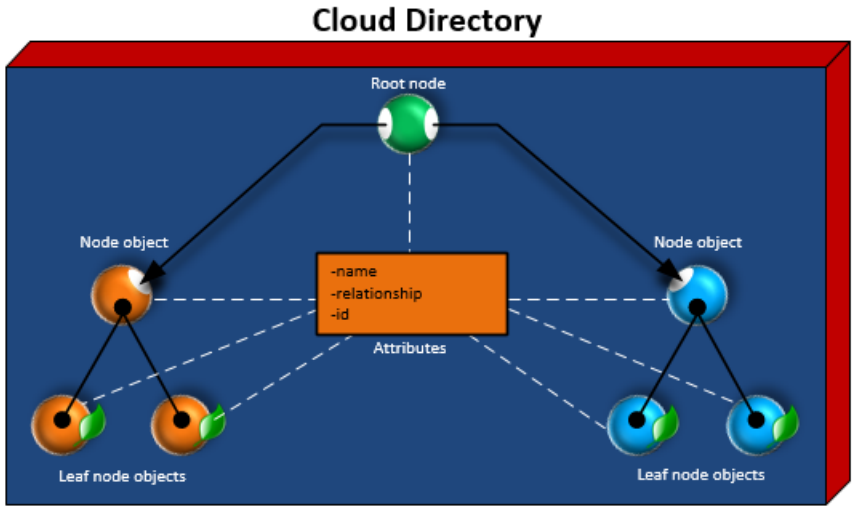
若要建立政策物件，請使用 CreateObject API 動作並搭配適當的面向：

- 若要將政策連接到物件或從物件分離，請分別使用 AttachPolicy 和 DetachPolicy 動作。
- 若要在樹狀目錄中一路向上尋找連接到物件的政策，請使用 LookupPolicy API 動作。
- 若要列出連接到特定物件的政策，請使用 ListObjectPolicies API 動作。

如需執行每個 API 動作所需之操作和許可的清單，請參閱「Amazon Cloud Directory 許可：動作、資源和條件參考」。

目錄結構

目錄中的資料是透過樹狀目錄模式以階層方式結構化，該樹狀目錄包含節點、分葉節點和節點之間的連結，如下圖所示。這有助於進行應用程式開發，以型塑、存放及快速周遊階層資料。



根節點

根節點是目錄中的頂端節點，可用來組織階層中的父子節點。就如同檔案系統中的資料夾可包含子資料夾和檔案。

Node

節點即代表物件，可擁有子物件。例如，一個節點在邏輯上可代表一組管理員，而各種使用者物件為子項或分葉節點。節點物件只能有一個父項。

分葉節點

分葉節點代表沒有子項的物件，不一定要直接連線到父節點。例如，使用者或裝置物件。分葉節點物件可有多個父項。雖然分葉節點物件不一定要連線到父節點，但強烈建議您這麼做，因為若沒有根節點路徑，就只能從物件的 `NodeId` 來存取物件。如果您誤置物件 ID，就再也無法找到該物件。

節點連結

節點彼此之間的連線。Cloud Directory 支援節點之間的各種連結類型，包括父子連結、政策連結和索引屬性連結。

Schemas

使用 Amazon Cloud Directory，結構描述定義在目錄中可以建立哪些類型的物件 (使用者、裝置和組織)、強制驗證每個物件類別的資料以及處理結構描述與時而進的變更。具體而言，結構描述可定義下列項目：

- 目錄中可映射到物件的一或多個面向類型 (例如 Person、Organization_Person)
- 目錄中可映射到物件的屬性 (例如 Name、Description)。各種類型面向上可為必要或選用的屬性，在面向的內容中定義。
- 會在物件屬性上強制執行的限制 (如必要、整數、字串)

當結構描述套用到目錄後，該目錄中的所有資料即必須符合此套用的結構描述。利用這種方式，結構描述定義基本上就是可利用已套用的結構描述用來建構多個目錄的藍圖。建立之後，這些套用的結構描述，每一個與原始藍圖在某方面都會有所不同。

您以後可以使用版本控制更新套用的結構描述，再將它重新套用到使用它的所有目錄。如需詳細資訊，請參閱 就地升級結構描述。

Cloud Directory 提供 API 操作來建立、讀取、更新和刪除結構描述。這可讓程式設計代理更輕鬆使用結構描述的內容。這種代理程式會存取目錄，以探索適用於目錄中資料的完整面向、屬性和限制集合。如需結構描述 API 的詳細資訊，請參閱Amazon Cloud Directory 參考指南。

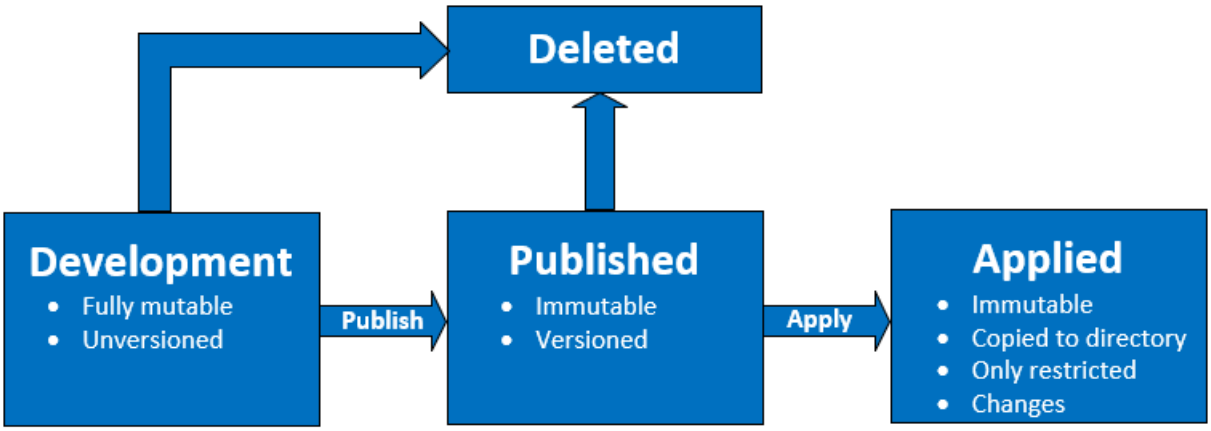
Cloud Directory 支援上傳相容的 JSON 檔案以建立結構描述。您還可以使用 AWS Directory Service 主控台建立和管理結構描述。如需詳細資訊，請參閱 建立 Amazon Cloud Directory。

主題

結構描述生命週期

Cloud Directory 提供結構描述生命週期，協助開發結構描述。這個生命週期包含三個狀態：開發、發佈和 Applied (已套用)。這些狀態旨在方便您建構和分發結構描述。這些狀態各有不同的功能，為此作業提供幫助。

下圖說明可能的轉換和用語。所有的結構描述轉換都是寫入時複製。例如，發佈開發結構描述不會改變或移除開發結構描述。



您可以刪除狀態為 Development (開發) 或 Published (已發佈) 的結構描述。刪除結構描述無法復原，一經刪除也無法還原。

Development (開發)、Published (已發佈) 和 Applied (已套用) 狀態的結構描述都有代表它們的 ARN。這些 ARN 是用於 API 操作，以描述 API 操作所在的結構描述。只要查看結構描述 ARN，很容易分辨結構描述的狀態。

- 開發：`arn:aws:clouddirectory:us-east-1:1234567890:schema/development/SchemaName`
- 已發佈：`arn:aws:clouddirectory:us-east-1:1234567890:schema/published/SchemaName/Version`
- 已套用: `arn:aws:clouddirectory:us-east-1:1234567890:directory/directoryid/schema/SchemaName/Version`

Development (開發) 狀態

結構描述初始建立的狀態為開發狀態。此狀態下的結構描述是完全可變的。您可以自由新增或移除面向和屬性。絕大多數的結構描述設計都是出現在這個狀態中。此狀態下的結構描述有名稱，但沒有版本。

Published (已發佈) 狀態

已發佈的結構描述狀態存放準備套用到資料目錄的結構描述。結構描述是從 Development (開發) 狀態發佈到 Published (已發佈) 狀態。您無法變更 Published (已發佈) 狀態的結構描述。您可以將已發佈的結構描述套用到任意數量的資料目錄。

已發佈和已套用的結構描述都必須有與其相關聯的版本。如需有關版本的詳細資訊，請參閱結構描述版本控制。

Applied (已套用) 狀態

已發佈的結構描述可以套用到資料目錄。已套用到資料目錄的結構描述即為 Applied (已套用)。一旦您將結構描述套用到資料目錄，您就可以在建立物件時使用結構描述的面向。您可以將多個結構描述套用到同一個資料目錄。已套用的結構描述只允許下列變更。

- 新增面向到已套用的結構描述
- 新增非必要屬性到已套用的結構描述

Facets

面向是結構描述內最基本的抽象概念。它們代表一組可在目錄中與物件相關聯的屬性，類似 LDAP 物件類別的概念。每個目錄物件最多可有特定數目的相關聯面向。如需詳細資訊，請參閱 Amazon Cloud Directory 限制。

每個面向會維護自己獨立的一組屬性。每個面向包含基本的中繼資料，例如面向名稱、版本資訊和行為。結構描述 ARN、面向和屬性的組合定義物件的唯一性。

一組物件面向、其限制及彼此之間的關係，構成抽象的結構描述定義。結構描述面向用於定義下列項目的限制：

1. 物件允許的屬性
2. 允許套用到物件的政策類型

您一旦在結構描述中新增了必要的面向，就可以將結構描述套用到您的目錄，並建立適用的物件。例如，您可以透過新增電腦、手機以及平板電腦等面向，定義裝置結構描述。然後，您可以使用這些面向在套用結構描述的目錄中，建立電腦物件、手機物件以及平板電腦物件。

Cloud Directory's schema support makes it easy to add or modify facets and attributes without worrying about breaking applications. 如需詳細資訊，請參閱 就地升級結構描述。

就地升級結構描述

Cloud Directory 讓您更新現有的結構描述屬性和面向，協助您整合應用程式和 AWS 提供的服務。Published (已發佈) 或 Applied (已套用) 狀態下的結構描述擁有版本，且無法變更。如需詳細資訊，請參閱 結構描述生命週期。

結構描述版本控制

結構描述版本會指出結構描述的唯一識別符，開發人員可在程式設計其應用程式時為符合特定規則和格式化資料而指定。版本控制搭配 Cloud Directory 的方法中有兩項重大差異，開發人員一定要了解。這些差異化因素 (主要版本和次要版本) 可以決定未來的結構描述對您應用程式的影響。

主要版本

主要版本是用於追蹤結構描述主要版本變更的版本識別符。其長度最多可有 10 個字元。同一個結構描述的不同版本都是完全獨立的。例如，名稱相同但版本不同的兩個結構描述，會被視為完全不同的結構描述，各有各的命名空間。

回溯不相容的變更

我們建議只有當結構描述不相容時才變更主要版本。例如，變更現有屬性的資料類型時 (例如從 `string` 變更為 `integer`)，或從您的結構描述捨棄必要屬性時。回溯不相容的變更需要將目錄資料從上一版的結構描述遷移到新版的結構描述。

次要版本

次要版本是用於就地升級結構描述的版本識別符，或當您想要回溯相容升級的版本識別符，例如新增其他屬性或新增面向。使用次要版本升級的結構描述，可以就地套用到使用它不會損及任何執行中應用程式的所有目錄。這包括使用在生產環境中的目錄。如需使用案例範例，請參閱「如何透過就地結構描述升級，輕鬆套用 Amazon Cloud Directory 結構描述變更」在 Cloud Directory 部落格中。

次要版本資訊和歷史記錄會與其他的結構描述資訊一起儲存在結構描述中繼資料儲存庫中。物件中不保留任何次要版本資訊。推出次要版本的優點是，只要主要版本不變更，就可以無縫使用用戶端程式碼。

次要版本限制

Cloud Directory 會保留並因此限制最多五個次要版本。不過，次要版本限制會以下列方式針對已發佈和套用的結構描述強制執行：

- 已套用結構描述： 一旦超過次要版本限制，Cloud Directory 就會自動刪除最舊的次要版本。

- 已發佈結構描述： 一旦超過次要版本限制，Cloud Directory 就不會刪除任何次要版本，但會透過`LimitExceededException`該限制已超出。一旦超過次要版本限制，您可以使用DeleteSchemaAPI 或請求提高限制。

使用結構描述升級 API 操作

您可以使用 `UpgradePublishedSchema` API 呼叫來升級已發佈的結構描述。結構描述升級使用 `UpgradeAppliedSchema` API 呼叫就地套用到倚賴它的目錄。此外，您也可以呼叫 `GetAppliedSchemaVersion`，藉此擷取已套用結構描述的主要及次要版本。或者，您也能呼叫，進而檢視相關聯的結構描述 ARN 及結構描述修訂歷史記錄`ListAppliedSchemaArns`。Cloud Directory 會保留已套用結構描述變更的 5 個最新版本。

如需說明範例，請參閱「如何透過就地結構描述升級，輕鬆套用 Amazon Cloud Directory 結構描述變更」在 Cloud Directory 部落格中。部落格文章會示範如何執行結構描述就地升級及在 Cloud Directory 結構描述版本。內容涵蓋如何在現有的面向中新增其他屬性、在結構描述中新增新的面向、發佈新的結構描述，以及將它套用到執行中的目錄來完成結構描述的就地升級。並說明如何檢視目錄結構描述的版本歷史記錄，這有助於確保目錄叢集執行相同的結構描述版本，並套用正確的結構描述變更歷史記錄。

受管結構描述

Cloud Directory 讓快速開發應用程式變得更加容易，藉此使用受管結構描述。透過受管結構描述，您就能建立目錄，然後更快速地從該目錄開始建立和擷取物件。如需詳細資訊，請參閱 建立您的目錄。

本服務目前擁有一個受管結構描述，稱為 `QuickStartSchema`。您可以利用類型連結等架構，藉此建置豐富的階層式資料模型，並建立各物件間的關係。接著，您便能周遊這些階層，進而在資料中查詢任何資訊。

`QuickStartSchema` 受管結構描述會以下列 JSON 格式呈現：

```
QuickStartSchema: {
    "facets": {
        "DynamicObjectFacet": {
            "facetStyle": "DYNAMIC"
        },
        "DynamicTypedLinkFacet": {
            "facetAttributes": {
                "DynamicTypedLinkAttribute": {
                    "attributeDefinition": {
                        "attributeRules": {},
                        "attributeType": "VARIANT",
```

```
                        "isImmutable": false
                    },
                    "requiredBehavior": "REQUIRED_ALWAYS"
                }
            },
            "identityAttributeOrder": [
                "DynamicAttribute"
            ]
        }
    }
}
```

QuickStartSchema ARN

QuickStartSchema 受管結構描述會採用下列 ARN：

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

舉例來說，您可以使用此 ARN 來建立名為 ExampleDirectory 的目錄，如下所示：

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

面向樣式

本服務擁有兩種不同的樣式 (Static 和 Dynamic)，可供您定義任何指定面向。

靜態面向

靜態面向是最佳選擇，當您擁有資料模型的所有細節 (如屬性及其資料類型的清單)，並且您還想定義屬性的限制條件 (如必要或唯一欄位)。Cloud Directory 會在您的物件建立或變更期間強制執行資料約束和規則檢查。

動態面向

若您選擇使用動態面向，即可視需要彈性地變更屬性數量，或是變更屬性內存放的資料值。在建立或變更物件期間，Cloud Directory 不會強制執行任何資料限制和規則檢查。

一旦建立具有動態面向的結構描述，您就能定義建立物件時所需的任何屬性。Cloud Directory 會接受索引鍵/值組形式的屬性，並將其存放至提供的物件上。

您可以將動態面向新增至新的或現有的結構描述。此外，您也可以在單一結構描述內結合使用靜態和動態面向，進而在目錄內善用每種面向樣式的優點，從中獲益。

當您使用動態面向建立任何屬性時，系統即會將這些屬性建立為 `Variant` 資料類型。若要儲存定義為`Variant`資料類型，您可以使用 Cloud Directory 支援的任何基本資料類型值 (如`String`或`Binary`。日後，您也能將屬性值變更為另一種資料類型；系統並不會強制驗證資料。

您可以透過動態面向來定義下列類型的物件：

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

如需有關受管理的結構描述、動態 Facet 或變體資料類型的其他詳細資料，以及查看範例使用案例，請參閱如何使用 AWS 受管結構描述，快速開發應 Amazon Cloud Directory 程式在 Amazon Cloud Directory 部落格中。

範例結構描述

具有適用於 Organizations、個人及裝置的範例結構描述。下節會列出各種範例結構描述，並列出每個的不同之處。

Organizations

下表列出包含在 Organizations 範例結構描述中的面向。

"Organization" 面向	資料類型	長度	必要行為？	描述
account_id	字串	1024	N	組織的唯一 ID
account_name	字串	1024	N	組織名稱
organization_status	字串	1024	N	「作用中」、「暫停」、「非作用中」、「關閉」等狀態

"Organization" 面向	資料類型	長度	必要行為？	描述
mailing_address (street1)	字串	1024	N	此公司/實體的實體郵寄地址
mailing_address (street2)	字串	1024	N	此公司/實體的實體郵寄地址
mailing_address (city)	字串	1024	N	此公司/實體的實體郵寄地址
mailing_address (state)	字串	1024	N	此公司/實體的實體郵寄地址
mailing_address (country)	字串	1024	N	此公司/實體的實體郵寄地址
mailing_address (postal_code)	字串	1024	N	此公司/實體的實體郵寄地址
電子郵件	字串	1024	N	組織的電子郵件 ID
web_site	字串	1024	N	網站 URL
telephone_number	字串	1024	N	組織的電話號碼
描述	字串	1024	N	組織的說明

"Legal_Entity" 面向	資料類型	長度	必要行為？	描述
registered_company_name	字串	1024	N	合法實體名稱
mailing_address (street1)	字串	1024	N	此公司/實體的實體註冊地址

"Legal_Entity" 面向	資料類型	長度	必要行為？	描述
mailing_address (street2)	字串	1024	N	此公司/實體的實體註冊地址
mailing_address (city)	字串	1024	N	此公司/實體的實體註冊地址
mailing_address (state)	字串	1024	N	此公司/實體的實體註冊地址
mailing_address (country)	字串	1024	N	此公司/實體的實體註冊地址
mailing_address (postal_code)	字串	1024	N	此公司/實體的實體註冊地址
industry_vertical	字串	1024	N	產業區隔
billing_currency	字串	1024	N	帳單貨幣
tax_id	字串	1024	N	稅務識別碼

Person

下表列出包含在 Person 範例結構描述中的面向。

"Person" 面向	資料類型	長度	必要行為？	描述
display_name	字串	1024	N	使用者名稱，適用於向使用者顯示。
first_name	字串	1024	N	使用者的指定名稱，或大部分西方語言中的名字
last_name	字串	1024	N	使用者的姓氏，或大部分西方語言中的姓氏

"Person" 面向	資料類型	長度	必要行為？	描述
middle_name	字串	1024	N	使用者的中間名
nickname	字串	1024	N	使用者在現實生活中的暱稱，例如，"Bob" 或 "Bobby"，而不是 "Robert"
電子郵件	字串	1024	N	使用者的電子郵件地址
mobile_phone_number	字串	1024	N	使用者的手機電話號碼
home_phone_number	字串	1024	N	使用者的手機電話號碼
username	字串	1024	Y	使用者的唯一識別符
profile	字串	1024	N	URI，亦即統一資源定位器，指向表示使用者線上設定檔 (如網頁) 的位置
picture	字串	1024	N	URI，亦即統一資源定位器，指向表示使用者影像的資源位置。
website	字串	1024	N	URL
timezone	字串	1024	N	使用者的時區
locale	字串	1024	N	用於指出使用者的預設位置，以本地化貨幣、日期時間格式或數字表示等項目。
address (street1)	字串	1024	N	此使用者的實體郵寄地址。
address (street2)	字串	1024	N	此使用者的實體郵寄地址。
address (city)	字串	1024	N	此使用者的實體郵寄地址。
address (state)	字串	1024	N	此使用者的實體郵寄地址。

"Person" 面向	資料類型	長度	必要行為？	描述
address (country)	字串	1024	N	此使用者的實體郵寄地址。
address (postal_code)	字串	1024	N	此使用者的實體郵寄地址。
user_status	字串	1024	N	指出使用者管理狀態的值

"Organization_Person" 面向	資料類型	長度	必要行為？	描述
title	字串	1024	N	組織中的職銜
preferred_language	字串	1024	N	指出使用者偏好的書寫或口語語言，通常用於選取本地化的使用者界面。
employee_id	字串	1024	N	字串識別符，通常是數字或英數字元，指派給個人
cost_center	整數	1024	N	識別成本中心
department	字串	1024	N	識別部門的名稱
manager	字串	1024	N	使用者的經理
company_name	字串	1024	N	識別組織的名稱
company_address (street1)	字串	1024	N	組織的實體郵寄地址
company_address (street2)	字串	1024	N	組織的實體郵寄地址
company_address (city)	字串	1024	N	組織的實體郵寄地址
company_address (state)	字串	1024	N	組織的實體郵寄地址

"Organization_Person" 面向	資料類型	長度	必要行為？	描述
company_address (country)	字串	1024	N	組織的實體郵寄地址
company_address (postalCode)	字串	1024	N	組織的實體郵寄地址

Device

下表列出包含在 Device 範例結構描述中的面向。

"Device" 面向	資料類型	長度	必要行為？	描述
device_id	字串	1024	N	唯一的英數字元裝置 ID
name	字串	1024	N	裝置的易記名稱
描述	字串	1024	N	裝置的說明
X.509_certificates	字串	1024	N	X.509 憑證
device_version	字串	1024	N	裝置版本
device_os_type	字串	1024	N	裝置的作業系統
device_os_version	字串	1024	N	裝置的作業系統版本號碼
serial_number	字串	1024	N	裝置的序號
device_status	字串	1024	N	裝置的狀態 (例如作用中、非作用中、暫停、關閉、結束)

自訂結構描述

建立自訂結構描述的第一步，是確實定義您必須建立索引的欄位。這些必要的欄位構成您結構描述的架構元素，在其中新增您自己的欄位。將每個欄位的名稱和類型 (例如字串、整數、布林值) 映射到您物件的結構。您可以使用類型和限制定義結構描述，然後將它們套用到目錄。Once defined, Cloud Directory performs validation for attributes.

如需詳細資訊，請參閱 建立結構描述。

屬性參考

Amazon Cloud Directory 的面向包含屬性。屬性可以是屬性定義或屬性參考。屬性定義是宣告其名稱和基本類型 (string、binary、Boolean、DateTime 或 number) 的屬性。該屬性也可選擇性地宣告面向的必要行為、預設值、不可變標記和屬性規則 (例如長度下限/上限)。

屬性參考則是從其他既有的屬性定義衍生其基本類型、預設值、不可變標記和屬性規則的屬性。屬性參考本身不具基本類型、預設值、不可變標記或規則，因為這些屬性皆衍生自目標屬性定義。

屬性參考可覆寫目標定義的必要行為 (以下會詳細說明)。

當您建立屬性參考時，您只需提供屬性名稱和目標屬性定義 (包含目標屬性定義的面向名稱和屬性名稱)。屬性參考不可參考其他屬性參考。另外，此時屬性參考也不可以其他結構描述的屬性定義為目標。

當您希望物件的兩個或多個屬性都參考同一個儲存位置時，您可以使用屬性參考。例如，假設有個套用了 User 面向和 EnterpriseUser 面向的物件。User 面向具有 FirstName 屬性定義，而 EnterpriseUser 面向具有指向 User.FirstName 的屬性參考。由於這兩種 FirstName 屬性都參考物件的同一個儲存位置，因此變更 User.FirstName 或 EnterpriseUser.FirstName 的效果皆相同。

API 範例

下列範例示範如何透過 Cloud Directory API 使用屬性參考。在這個範例中，基礎面向包含屬性定義，而另一個面向則包含參考該基礎面向中屬性的屬性。請注意，雖然基礎面向為 Not Required，參考屬性仍可標示為 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)
```

JSON 範例：

下列範例示範如何在 JSON 模型中使用屬性參考。此模型呈現的結構描述與上述模型相同。

```
{
 "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"
    }
}
```

屬性參考考量

屬性參考必須以同一個結構描述中的既有屬性定義為目標。

- 屬性參考可以同一個面向或不同面向中的既有屬性定義為目標。
- 屬性參考不可以其他屬性參考為目標。
- 若面向包含的屬性定義為另一個面向的屬性參考目標，則在所有參考刪除前，該面向不可刪除。

您可以像使用傳統屬性定義一樣，透過建立物件或將面向套用到現有物件來使用屬性參考。

> **Note**
>
> 您可以套用參考其他面向的面向，而不必直接套用目標面向。若未套用目標面向，屬性參考的行為就不會變更。(只有當您想要物件上存在目標面向的其他屬性時，才要套用該目標面向。)

設定屬性參考值

當您想要變更屬性值時，可以呼叫 UpdateObjectAttributes API 動作。更新 (或刪除) 該物件之定義或對同一個定義的任何其他參考皆具有相同效果。

取得屬性參考值

您可以呼叫 ListObjectAttributes API 動作來擷取儲存別名。此呼叫會傳回元組清單，每個清單都包含屬性鍵及其關聯的值。屬性鍵會對應到該物件上存在的儲存別名清單。

> **Note**
>
> 若面向未明確套用到物件，屬性鍵仍可能傳回。若屬性參考以未套用到物件的面向為目標，就可能發生這種情況。

例如，假設您具有 User 面向和 EnterpriseUser 面向。EnterpriseUser.FirstName 屬性參考 User.FirstName。接著您將 User 面向和 EnterpriseUser 面向皆套用到物件，並將 User.FirstName 設定為 Robert，再將 EnterpriseUser.FirstName 設定為 Bob。當您呼叫 ListObjectAttributes 時，您只會看到 "User.FirstName = Bob"，因為這兩個 FirstName 屬性只有一個儲存別名。

搭配屬性參考使用索引

您只可用屬性定義建立索引，而不可用參考。列出索引並不會傳回屬性參考的屬性鍵。但會傳回索引物件上存在的參考作為目標之任何屬性定義的屬性鍵。換言之，在索引層，屬性參考僅作為屬性的替代識別符，其會在執行時間解析為正確的屬性定義識別符。

例如，假設您具有面向為 User 且屬性為 FirstName 的索引。您先連接僅套用 EnterpriseUser 面向的物件。再將該物件的 EnterpriseUser.FirstName 屬性值設定為 Bob。最後呼叫 ListIndex 動作。結果僅會包含 "User.FirstName = Bob"。

屬性參考的必要行為

屬性參考可以具有與其目標屬性定義不同的必要行為。這讓基礎定義可為選用項，而對同一個定義的參考則可為必要項。當物件具有基礎定義及對該相同基礎定義的一或多個參考時，該基礎定義和所有參考皆必須遵循所有相關屬性存在的最強大必要行為。

- 與屬性定義一樣，當您建立物件或將面向新增至現有物件時，必須為任何必要屬性定義提供值。
- 為方便起見，當物件的多個屬性參考相同儲存位置時，您只需為該儲存位置提供一個屬性的值。
- 同樣地，如果您為相同儲存位置提供多個值，這些值也都必須相等。

屬性規則

規則描述屬性類型的允許值，並限制任何特定屬性的允許值。當您建立面向時，您必須將規則指定為屬性定義的一部分。Cloud Directory 支援下列規則類型：

- 字串長度
- 二進位長度
- 字串來源集
- 數值比較

字串長度

限制字串屬性值的長度。

允許的規則參數金鑰：下限、上限

允許的規則參數值：數值

二進位長度

限制二元屬性值的位元組陣列長度。

允許的規則參數金鑰：下限、上限

允許的規則參數值：數值

字串來源集

將字串屬性的值限制在允許的指定字串集合。

允許的規則參數金鑰：allowedValues

允許的規則參數值：每個字串都是 UTF-8 編碼的字串集

允許的值以逗號分隔，且可用引號括住。當允許的值包含逗號時，這相當實用。例如：

- One,two,three = 符合 One、two 或 three
- “with,comma”,”withoutcomma” = 符合 “with,comma” 或 “withoutcomma”
- with”quote,withoutquote 符合 ‘with”quote’ 或 ‘withoutquote’

數值比較

限制數值屬性允許的數值。

允許的規則參數金鑰：下限、上限

允許的規則參數值：數值

格式化規格

Cloud Directory 結構描述會將結構新增至您資料目錄的資料中。Cloud Directory 為您提供兩種定義結構描述的機制。開發人員可以使用特定的 API 操作來建構結構描述，或者他們可以使用結構描述上傳功能來完整上傳結構描述。結構描述文件可以透過 API 呼叫或主控台上傳。本節說明當您上傳整份結構描述文件時要使用的格式。

JSON 結構描述格式

結構描述文件是使用以下整體格式的 JSON 文件。

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

結構描述文件包含面向的面向名稱對應圖。因此每個面向又包含一個屬性對應圖。結構描述內的所有面向名稱必須是唯一的。面向內的所有屬性名稱必須是唯一的。

屬性 JSON 子區塊

面向包含屬性。每個屬性定義可存放在屬性的值類型。以下 JSON 格式說明一個屬性。

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

您必須提供屬性定義或屬性參考。如需各自的詳細資訊，請參閱相關子區塊。

必要的行為欄位會指出此屬性是否為必要。您必須提供此欄位。可能的值如下：

- REQUIRED_ALWAYS：建立物件或將面向新增到物件時，必須提供此屬性。您不能移除此屬性。
- NOT_REQUIRED：此屬性可能出現，也可能不出現。

屬性定義子區塊

屬性會定義與屬性值相關聯的類型和規則。以下 JSON 配置說明格式。

```
{
    "attributeType": One of "STRING", "NUMBER", "BINARY", "BOOLEAN" or "DATETIME",
    "defaultValue": Default Value Subsection,
```

```
    "isImmutable": true or false,
    "attributeRules": "Attribute Rules Subsection"
}
```

預設值子區塊

確實指定下列預設值的其中之一。長值和布林值應位在引號外 (為其各自的 Javascript 類型而不是字串)。使用 URL 安全的 Base64 編碼字串提供二元值 (如 RFC 4648 中所述)。日期時間以自 epoch (1970 年 1 月 1 日 00:00:00 UTC) 起算的毫秒數表示。

```
{
 "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
}
```

屬性規則子區塊

屬性規則定義屬性值限制。您可以為每個屬性定義多個規則。屬性規則包含規則的規則類型和一組參數。您可以在「屬性規則」一節找到詳細資訊。

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

屬性參考子區塊

屬性參考是進階功能。它們允許多個面向共用屬性定義和存放的值。如需詳細資訊，請參閱屬性參考一節。您可以使用以下範本在 JSON 結構描述中定義屬性參考。

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

```
}
```

結構描述文件範例

下列結構描述文件範例顯示有效的 JSON 格式。

> **Note**
>
> 以 `allowedValues` 字串表示的所有值都必須以逗號分隔，且不含空格。例如，`"SENSITIVE,CONFIDENTIAL,PUBLIC"`。

基本結構描述文件

```
{
    "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"
        }
    }
}
```

使用類型連結的結構描述文件

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

目錄物件

開發人員使用可擴展的結構描述建立目錄物件模型，以自動強制執行資料正確性限制，讓程式設計變得更容易。Amazon Cloud Directory 根據您已定義的已建立索引之屬性提供豐富的資訊查詢，以快速在目錄樹狀目錄內執行樹狀目錄周遊及搜尋。Cloud Directory 資料於靜態及傳輸中加密。

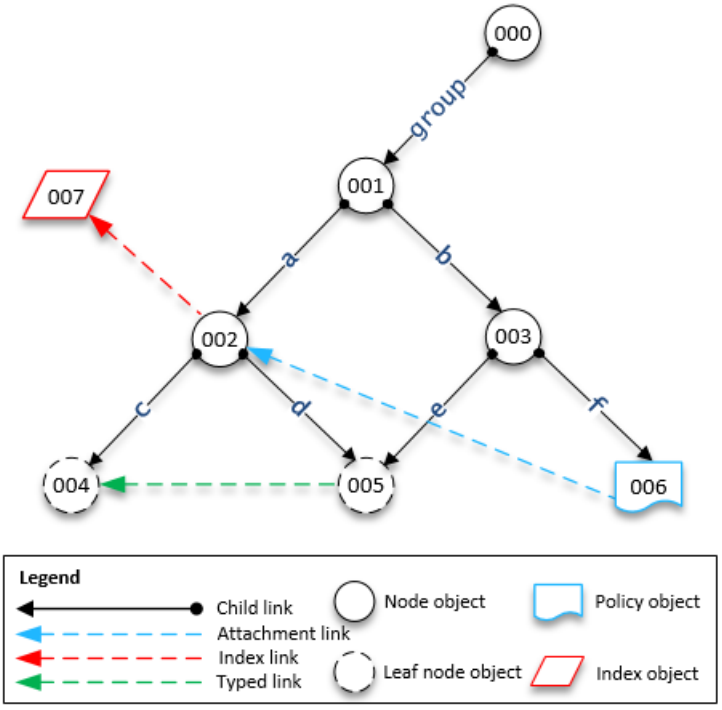
物件是基本的雲目錄元素。每個物件都有全域唯一的識別符，由物件識別符所指定。物件是零或多個面向及其屬性金鑰和值的集合。物件可從單一已套用之結構描述中的一或多個面向建立，或從多個已套用之結構描述中的多個面向建立。在物件建立期間，您必須指定所有必要的屬性值。物件可具的面向數目有限。如需詳細資訊，請參閱 Amazon Cloud Directory 限制。

物件可以是一般物件、政策物件或索引物件。物件也可以是節點物件或分葉節點物件。物件類型可自它所連接的面向物件類型推斷。

主題

Links

連結是定義關係之兩個物件間的導向邊緣。Cloud Directory 目前支援下列連結類型。



子連結

子連結會建立其所連線之物件間的父子關係。例如，上圖中的子連結 b 連線物件 001 和 003。子連結定義 Cloud Directory 中的階層。子連結參與定義連結指向的物件路徑時會有名稱。

附件連結

附件連結會將分葉節點政策物件套用到另一個分葉節點或節點物件。附件連結不會定義 Cloud Directory 的階層結構。例如，在上圖中，附件連結會將存放在政策分葉節點物件 006 的政策套用到節點物件 002。每個物件都可以連接多項政策，但只能連接一項任何指定政策類型的政策。

索引連結

索引連結根據索引物件和您已定義的已建立索引之屬性提供豐富的資訊查詢，以快速在目錄樹狀目錄內執行樹狀目錄周遊及搜尋。從概念上講，索引類似於具有子節點：連接子項時，會根據已建立索引之屬性標示，而不是指定標籤。不過，索引連結不是父子邊緣，各有各的列舉 API 操作集。如需詳細資訊，請參閱 索引和搜尋。

類型連結

類型連結可讓您在雲端目錄中的階層內或跨階層的物件之間建立關係。接著，您就能利用這些關係來查詢資訊，例如哪些使用者具備「xyz」裝置，或是使用者「abc」擁有哪些裝置。

您可以使用類型連結在您的目錄中建立不同物件的關係模型。例如，在上圖中，請考慮代表使用者的物件 004 和代表裝置的物件 005 之間的關係。

我們可能會使用類型連結建立這兩個物件之間的擁有權關係模型。我們可以在類型連結中新增屬性，來表示購買的費用或者裝置為租賃或買斷。與類型連結相關的屬性有兩種：

- 以身分為基礎的屬性 – 該屬性可區分類型連結與其他連結 (例如，子連結、附件連結、索引連結)。每個類型連結面向都會定義一組排序的身分屬性。類型連結的身分是來源物件 ID、面向識別碼 (類型)，其身分屬性的值 (由其面向定義) 以及目標物件 ID。單一目錄中的識別碼必須是唯一的。
- 選用屬性 – 該屬性可針對與連結身分無關的類型連結，存放其追蹤特性。例如，選用屬性可識別類型連結首次建立或上次修改的日期。

與物件一樣，您必須使用 `CreateTypedLinkFacet` API 定義類型連結架構及其屬性，以建立類型連結面向。類型連結面向需要唯一的面向名稱及與連結相關聯的屬性集。當設計類型連結架構時，您可以在類型連結面向上定義一組排序的屬性。若要檢視類型連結範本結構描述，請參閱「使用類型連結的結構描述文件」。

當您需要執行下列任一項操作時，可以使用類型連結屬性：

- 允許篩選傳入或傳出的類型連結。如需詳細資訊，請參閱 類型連結清單。
- 代表兩個物件之間的關係。
- 追蹤類錫連結的相關管理資料，例如建立連結的日期。

決定類型連結是否適合您的使用案例時，請考慮以下事項：

- 類型連結不能用於路徑型物件規格。您必須改用 `ListOutgoingTypedLinks` 或 `ListIncomingTypedLinks` API 操作選取類型連結。
- 類型連結不參與 `LookupPolicy` 或 `ListObjectParentPaths` API 操作。
- 同一方向的兩個相同物件之間的類型連結，可能不會有相同的屬性值。這有助於避免相同物件之間重複類型連結。
- 當您想要新增選用資訊時，可以使用其他屬性。
- 所有身分屬性值的合併大小皆限於 64 位元組。如需詳細資訊，請參閱 Amazon Cloud Directory 限制。

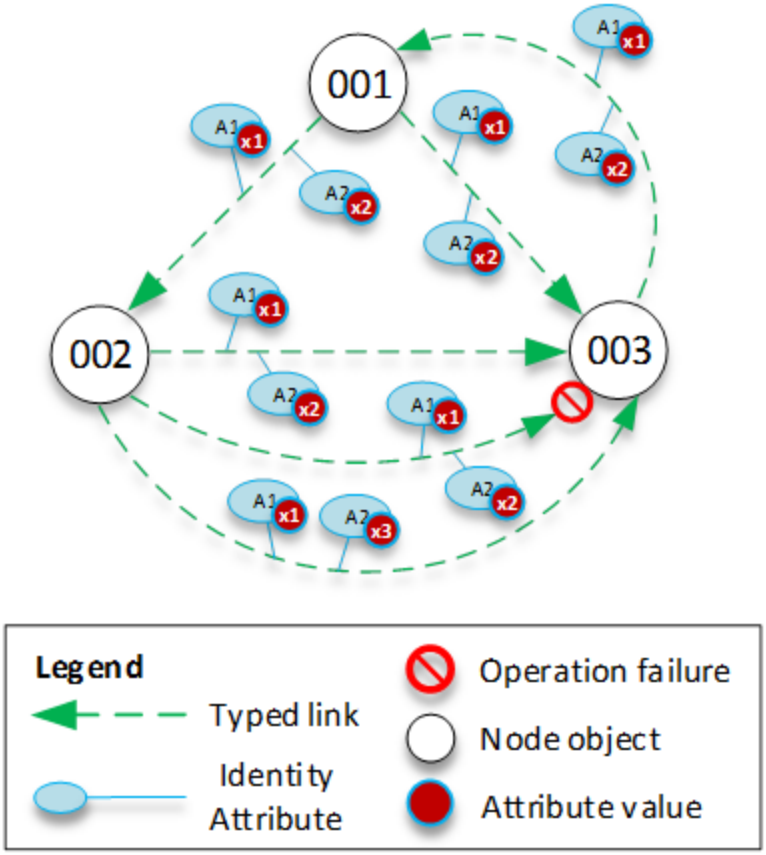
相關 Cloud Directory 部落格文章

- 使用 Amazon Cloud Directory 類型連結建立並搜尋各階層間的關係

類型連結身分

身分是唯一可定義兩個物件之間是否存在類型連結的項目。例外狀況是當您使用完全相同的屬性值連接同一方向的兩個物件時。屬性必須設定為 REQUIRED_ALWAYS。

從不同類型連結面向建立的類型連結彼此之間永遠不會發生衝突。例如，請考量下圖：



- 物件 001 讓具有相同屬性值 (x1 和 x2) 的類型連結和屬性 (A1 與 A2) 到不同的物件 (002 和 003)。這個操作會成功。
- 物件 002 和 003 之間有類型連結。這個操作會失敗，因為物件之間不能存在方向相同、屬性相同的兩個類型連結。
- 物件 001 和 003 之間有兩個類型連結具有相同的屬性。不過，因為連結的方向不同，所以這個操作會成功。
- 物件 002 和 003 之間有類型連結，A1 的值相同，但 A2 的值不同。類型連結身分考量所有屬性，所以這項操作會成功。

類型連結規則

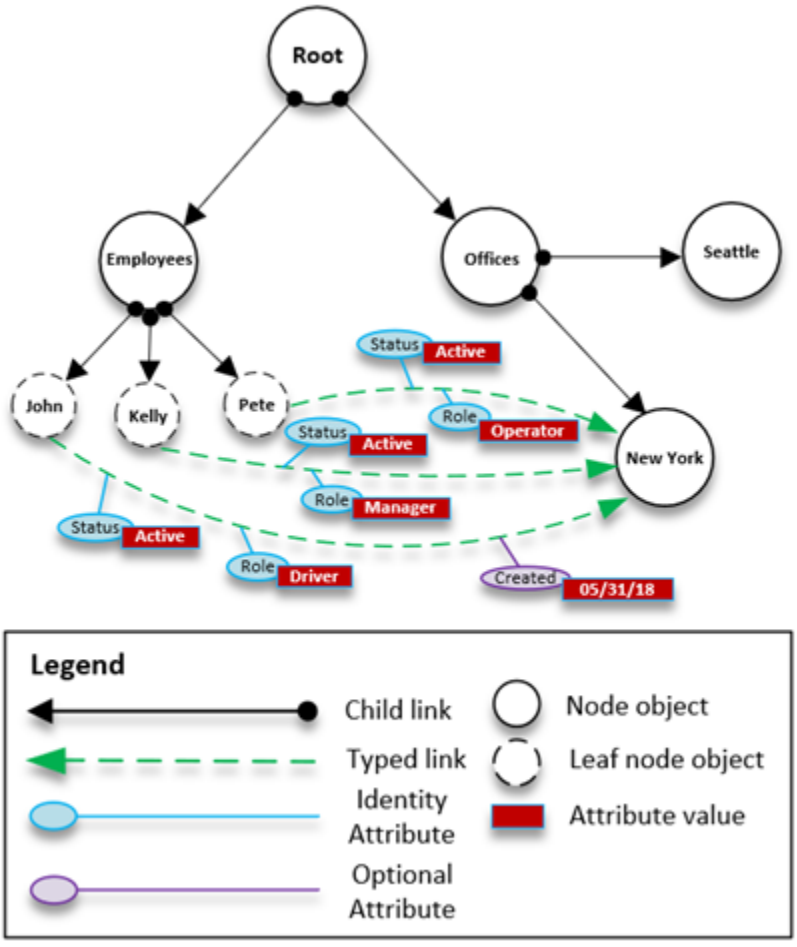
當您希望在連結屬性中新增限制時，您可以在類型連結屬性中新增規則。這些規則等同於物件屬性的規則。如需詳細資訊，請參閱 屬性規則。

類型連結清單

Cloud Directory 提供的 API 操作可讓您用來從物件選取傳入或傳出類型連結。您可以選取特定的類型連結部分，不用重複每一個類型連結。您也可以指定特定的類型連結面向，只篩選該類型的類型連結。

您可以根據類型連結面向上定義的屬性順序篩選類型連結。您可以為多個屬性提供範圍篩選條件。提供類型連結選取範圍時，必須在結尾處指定任何不精確的範圍。任何未指定範圍的屬性都假定符合整個範圍。篩選條件依類型連結面向上定義的屬性順序執行，不是向任何 API 呼叫提供的順序。

例如，在下圖中，請假設有存放員工及其技能相關資訊的 Cloud Directory。



例如，我們以名為 EmployeeCapability 的類型連結建立員工技能的模型，它由三個字串屬性設定：Status、Role 和 Created。ListIncomingTypedLinks 和 ListOutgoingTypedLinks API 操作支援下列篩選條件。

- 面向 = EmployeeCapability、狀態 = Active、角色 = Driver
  - 選取擔任司機的作用中員工。此篩選條件包含兩個完全相符的項目。
- 面向 = EmployeeCapability，狀態 = Active，角色 = Driver 建立日期 = 05/31/18
  - 選取身為駕駛並且其面向是在 2018 年 5 月 31 日當天或之後建立的作用中員工。
- 面向 = EmployeeCapability、狀態 = Active
  - 選取所有作用中的員工。
- 面向 = EmployeeCapability、狀態 = Active、角色 = A 到 M
  - 選取角色由 A 到 M 的作用中員工。
- 面向 = EmployeeCapability
  - 這會選取 EmployeeCapability 類型的所有類型連結。

不支援下列篩選條件：

- 面向 = EmployeeCapability、狀態介於 A 和 C 之間、角色 = Driver
  - 這個篩選條件不成立，因為任何範圍都必須出現在篩選條件的結尾處。
- 面向 = EmployeeCapability、角色 = Driver
  - 這個篩選條件不成立，因為隱含的狀態範圍不是完全相符的項目，也不會出現在範圍清單的結尾處。
- 狀態 = Active
  - 這個篩選條件不成立，因為未指定類型連結面向。

類型連結結構描述

您有兩種方式可以建立類型連結面向。您可以從個別的 API 呼叫管理您的類型連結面向，包括 CreateTypedLinkFacet、DeleteTypedLinkFacet 和 UpdateTypedLinkFacet。您也可以上傳代表您在單一 PutSchemaFromJson API 呼叫中之結構描述的 JSON 文件。如需詳細資訊，請參閱 JSON 結構描述格式。若要檢視類型連結範本結構描述，請參閱「使用類型連結的結構描述文件」。

結構描述開發生命週期不同階段中允許的變更類型，類似物件面向操作允許的變更。開發狀態下的結構描述支援任何變更。Published (已發佈) 狀態下的結構描述為不可變，也不支援任何變更。結構描述只允許已套用到資料目錄的特定變更。一旦您在已套用的類型連結面向上設定了順序和屬性，該順序即無法變更。

其他兩項 API 操作會列出面向及其屬性：

- ListTypedLinkFacetAttributes
- ListTypedLinkFacetNames

類型連結互動

建立類型連結面向後，您就可以開始建立類型連結並與其互動。若要連接和分離類型連結，請使用 AttachTypedLink 和 DetachTypedLink API 操作。

TypedLinkSpecifier 是包含可唯一識別類型連結所有資訊的結構。在此結構中，您可以找到 TypedLinkFacet、SourceObjectID、DestinationObjectID 和 IdentityAttributeValues。只有它們可用來指定要操作的類型連結。AttachTypedLink API 操作傳回類型連結指標，而 DetachTypedLink API 操作則接受指標做為輸入。同樣地，ListIncomingTypedLinks 和 ListOutgoingTypedLinks API 操作提供類型連結指標做為輸出。您也可以從頭開始建構類型連結指標。類型連結相關 API 操作的完整清單包括下列項目：

- AttachTypedLink
- CreateTypedLinkFacet
- DeleteTypedLinkFacet
- DetachTypedLink
- GetLinkAttributes
- GetTypedLinkFacetInformation
- ListIncomingTypedLinks
- ListOutgoingTypedLinks
- ListTypedLinkFacetNames
- ListTypedLinkFacetAttributes
- UpdateLinkAttributes
- UpdateTypedLinkFacet

> **Note**
>
> 不支援屬性參考和更新類型連結。若要更新類型連結，您必須將其移除並新增更新的版本。

範圍篩選條件

數個 Cloud Directory 清單 API 可讓您指定範圍形式的篩選條件。這些篩選條件可讓您有效率地選取已連接到指定節點的部分連結。

一般提供對應範圍 (鍵/值對陣列)，其金鑰是屬性識別符且其值為對應的範圍。這可篩選由一或多個屬性構成身分的連結。例如，為建立角色關係模型以決定許可而設定的 TypedLink，可能同時擁有 RoleType 和 Authorizer 屬性。然後，`ListOutgoingTypedLinks` 呼叫可以指定範圍來篩選 RoleType:"Admin" 和 Authorizer:"Julia" 的結果。用來篩選單一清單請求的對應範圍必須只包含定義連結身分的屬性 (索引的 OrderedIndexedAttributeList 或 TypedLink 的 IdentityAttributeOrder)，但不需要包含它們所有的範圍。缺漏的範圍會自動填入跨所有可能值的範圍 (從 FIRST 到 LAST)。

如果您將每個屬性想像成定義這些值的獨立一般網域，範圍結構會在該網域中定義兩個邏輯點，起點和終點，範圍即是這些點之間所有符合可能的點。範圍結構的 StartValue 和 EndValue 使用「模式」定義這兩個點的基礎，並進一步細化到指出各點本身是包含在範圍內或排除在範圍外。在上述 RoleType:"Admin" 範例中，RoleType 屬性的兩個值都是 "Admin"，兩種模式也都是 "INCLUSIVE" (寫作 ["Admin" to "Admin"])。在 User 面向的 LastName 中定義索引之 ListIndex 呼叫的篩選條件，可能使用 StartValue="D"、StartMode=INCLUSIVE、EndValue:"G"、EndMode:EXCLUSIVE 將清單縮小至開頭為 D、E 或 F 的名稱。

範圍的起點必須一律先於或等於終點。如果 EndValue 先於 StartValue，Cloud Directory 就會傳回錯誤。這些值也必須和它們篩選的屬性是同一個基本類型，String 屬性是字串值、Integer 屬性是整數，以此類推。例如，StartValue="D"、StartMode=EXCLUSIVE、EndValue="D"、EndMode=INCLUSIVE 無效，因為起點在終點所包含值的後面。

起點或終點有三種特殊模式可用。下列模式不需要指定對應的值欄位，因為它們自行暗示位置。

- FIRST - 先於網域中所有可能的值。用於起點時，這符合從網域最開始到終點的所有可能值。用於終點時，網域內沒有任何值符合此範圍。
- LAST - 接續在網域中所有可能值的後面。用於終點時，這符合接續在起點後所有可能的值，包括缺漏值。用於起點時，網域內沒有任何值符合此範圍。
- LAST_BEFORE_MISSING_VALUES - 此模式只對可省略值的選用屬性有用 (請參閱「缺少的值」)。它對應到缺漏值和實際網域值之間的點。用於終點時，這符合接續在起點後所有非缺漏網域的值。用於起點時，它會排除所有非缺漏網域值。如為必要屬性，此模式等同於 LAST，因為不會有任何缺漏值。

多範圍限制

Cloud Directory 限制有多個屬性的模式，以保證有效率、低延遲的請求處理。每個有多識別屬性的連結會依定義良好的順序指定它們。例如，上述的角色範例定義 RoleType 屬性為最重要，而 Authorizer 屬性最不重要。List 請求只能指定不可為 1) 單一值或 2) 跨所有可能值的單一「合格」範圍 (可能有多個範圍符合這兩項請求)。重要屬性多過合格範圍屬性的任何範圍都必須指定單一值，而不重要範圍的任何範圍則必須跨所有可能的值。在角色範例中，篩選條件集 (RoleType:”Admin”、Authorizer:[“J” to “L”]) (單一值 + 合格範圍)、(RoleType:[”Admin” to “User”]) (合格範圍 + 隱含跨越範圍) 和 (RoleType:[FIRST to LAST]) (兩個跨越範圍、一個隱含) 都是有效的篩選條件集範例。(RoleType:[FIRST to LAST]、Authorizer:”Julia”) 不是有效的集合，因為跨越範圍比單一值範圍更重要。

填寫範圍結構的一些實用模式包括：

比對單一值

指定 StartValue 和 EndValue 的值，並將這兩種模式設為 “INCLUSIVE”。

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

比對字首

指定字首為使用 INCLUSIVE 模式的 StartValue，前綴後第一個值為使用 EXCLUSIVE 模式的 EndValue。

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

大於某值的篩選條件

指定使用 EXCLUSIVE 模式的 StartValue 值，以及 LAST 做為 EndMode (或 LAST_BEFORE_MISSING_VALUES 排除遺漏的值，如果適用)。

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

小於或等於某值的篩選條件

指定使用 INCLUSIVE 模式的 EndValue 值，且 FIRST 做為 StartMode。

缺少的值

當屬性在結構描述中標示為選用時，它的值可能會「缺漏」，因為連接面向時不需要提供它，或可能在後來刪除了屬性。如果有缺漏值的物件連接到索引，索引連結仍然存在，但會移至連結集的結

尾處。ListIndex 呼叫會先傳回顯示所有已建立索引之屬性的任何連結，再傳回缺漏一或多個屬性的連結。這大概類似關聯式資料庫的 NULL 值，但這些值會排列在非 NULL 值的後面。您可以選擇 LAST 或 LAST_BEFORE_MISSING_VALUES 模式，指定某個範圍是否包含這些缺漏值。例如，您向 ListIndex 呼叫提供篩選條件，篩選範圍為 [LAST_BEFORE_MISSING_VALUES to LAST]，只傳回索引中的缺漏值。

存取物件

您可透過路徑或 objectIdentifier 存取目錄中的物件。

路徑— 透過說明到達方法的路徑名稱，可在 Cloud Directory 樹狀目錄中識別及找到每個物件。路徑從根目錄開始 (上圖中的節點 000)。路徑的表示法為以斜線 (/) 標記的連結開始，後面接著以路徑分隔符號 (也是斜線) 分隔的子連結，一直到路徑的最後一個部分。例如，使用路徑 005 可以找到上圖中的物件 /group/a/d。多個路徑可找到一個物件，因為分葉節點物件可有多個父項。使用以下路徑也可以找到物件 005：/group/b/e

物件識別器— 目錄中的每個物件都有唯一的全域識別符，即ObjectIdentifier。ObjectIdentifier傳回為CreateObjectAPI 呼叫。您也可以ObjectIdentifier透過使用GetObjectInformationAPI 呼叫。例如，若要擷取物件 005 的物件識別符，您可以指向物件的物件參考為路徑呼叫 GetObjectInformation 來導致物件，即 group/b/e 或 group/a/d。

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

填入物件

您可以使用 AddFacetToObject API 呼叫將新的面向新增到物件。物件類型是由連接到物件的面向所決定。目錄中的物件附件是根據物件類型運作。連接物件時，請記住這些規則：

- 分葉節點物件不能有子項。
- 節點物件可有多個子項。
- 政策類型的物件不能有子項，但可有零或一個父項。

更新物件

您有多種方式可以更新物件：

1. 使用 UpdateObjectAttributes 操作更新物件上的個別面向屬性。
2. 使用 AddFacetToObject 操作將新的面向新增到物件。
3. 使用 RemoveFacetFromObject 操作從物件中刪除現有的面向。

刪除物件

連接的物件必須符合特定條件，您才可以從目錄中刪除它：

1. 您必須將物件從樹狀目錄分離。只有當物件沒有任何子項時，您才可以分離物件。如果物件有子項，您必須先分離所有子項。
2. 您只有在刪除該物件中的所有屬性後，才可以刪除分離的物件。您可以透過刪除連接到該物件的每個面向來刪除物件上的屬性。您可以呼叫 GetObjectInformation，以擷取連接到物件的面向清單。
3. 物件也必須沒有任何父項、沒有任何政策附件、沒有任何索引附件。

因為物件必須完全與樹狀目錄分離才能刪除，所以您必須使用物件識別符將其刪除。

查詢物件

本節討論與在目錄中查詢物件相關的各種元素。

目錄周遊

因為 Cloud Directory 是樹狀目錄，所以您可以使用ListObjectChildrenAPI 操作，或從下而上使用ListObjectParentsAPI 操作。

政策查閱

考慮到物件參考，LookupPolicy API 操作會以由上而下的方式傳回所有連接的政策及其一或多個根路徑。任何無法指向根目錄的路徑都予以忽略。會傳回所有政策類型物件。

如果此物件是分葉節點，它可以有到達根目錄的多個路徑。這個呼叫每個呼叫只傳回一個路徑。若要擷取額外的路徑，請使用分頁字符。

索引查詢

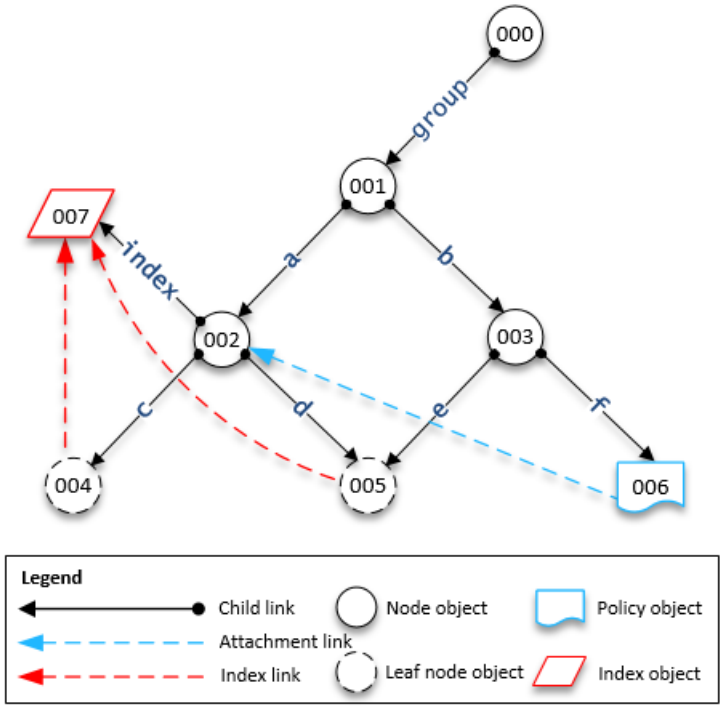
Cloud Directory 使用以下範圍支援豐富的索引查詢功能：

- FIRST - 從第一個已建立索引的屬性值開始。開始屬性值為選用。
- LAST - 傳回直到索引結尾的屬性值，包括缺漏值。結束屬性值為選用。
- LAST_BEFORE_MISSING_VALUES - 傳回直到索引結尾的屬性值，排除缺漏值。
- INCLUSIVE - 包括要指定的屬性值。
- EXCLUSIVE - 排除要指定的屬性值。

父項路徑清單

使用 `ListObjectParentPaths` API 呼叫，您可以擷取任何物件類型的所有可用父項路徑 (節點、分葉節點、政策節點、索引節點)。當您需要評估物件的所有父項時，這個 API 操作很有用。這個呼叫會傳回根目錄中所請求物件前的所有物件。它也會根據使用者定義的 `MaxResults` 傳回路徑數，如果父項有多個路徑。多個 API 呼叫中傳回的路徑和節點順序是一致的，除非物件已遭刪除或移動。目標物件中不指向根目錄的路徑都予以忽略。

如需此運作方式的範例，假設目錄的物件階層類似下圖。



有編號的圖形代表不同的物件。該物件和根目錄 (000) 之間的箭頭數代表完整的路徑，可能會在輸出中表示。下表顯示對階層中特定分葉節點物件所執行之查詢的請求和回應。

物件的範例查詢

要求	回應
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** 在本例中，物件 005 有兩個父節點 002 和 003。而且，因為 MaxResults 是 2，所以兩個路徑都在清單中顯示物件。
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** 在本例中，物件 005 有兩個父節點 002 和 003。而且，因為 MaxResults 是 1，所以會建立多個有頁面字符的編頁呼叫，以取得有物件清單的所有路徑。
006, PageToken: null, MaxResults: 1	[{/group/b/f, [000, 001, 003, 006]}], PageToken: null

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

一致性層級

Amazon Cloud Directory 是分散式目錄存放區。資料會分發到不同可用區域的多部伺服器。成功的寫入請求會更新所有伺服器上的資料。通常在一秒內，最終所有伺服器都可取得資料。為了協助服務的使用者，Cloud Directory 提供兩種一致性層級的服務處理讀取操作。本節說明不同的一致性層級和 Cloud Directory 的最終一致性本質。

閱讀隔離層級

從 Cloud Directory 讀取資料時，您必須指定您想要讀取的隔離層級。不同的隔離層級對延遲和資料更新有不同的權衡。

- 最終— 快照隔離層級會讀取任何立即可用的資料。在所有隔離等級中，此等級的延遲最低。 而且，其也讓您有機會檢視目錄中的舊資料。EVENTUAL 隔離不提供先寫後讀一致性。這表示無法保證您能在寫入後，立即讀取資料。
- 可序列化— 可序列化的隔離層級提供 Cloud Directory 所能提供的最高層級一致性。在 SERIALIZABLE 隔離層級完成讀取可確保您收到所有成功寫入的資料。您請求的資料如已變更，但此變更尚未提供，系統會拒絕您的 `RetryableConflictException` 請求。我們建議您重試這些例外狀況 (請參閱下節)。重試成功後，SERIALIZABLE 讀取會提供先寫後讀一致性。

寫入請求

Cloud Directory 可確保多個寫入請求不會同時更新相同的一或多個物件。如果發現同一個物件上正同時進行兩項寫入請求，其中一項操作會因為 `RetryableConflictException` 而失敗。我們建議您重試這些例外狀況 (請參閱下節)。

> **Note**
>
> 在寫入操作期間收到的 RetryableConflictException 回應收到無法用於偵測競爭條件。如果已出現促成這種狀況的使用案例，不保證一定會發生例外狀況。是否發生例外狀況，取決於內部處理每個請求的順序。

RetryableConflictExceptions

在同一個物件上寫入後，當使用 SERIZABLE 隔離層級執行寫入操作或讀取操作時，Cloud Directory 可能回應RetryableConflictException。這個例外狀況表示 Cloud Directory 伺服器還沒有處理之前寫入的內容。這些情況是暫時的，會迅速自行修復。請務必注意，RetryableConflictException 不能用於偵測任何類型的先寫後讀一致性。不保證特定的使用案例會造成此例外狀況。

建議您設定 Cloud Directory 用戶端重試RetryableConflictException。此組態在操作期間提供無錯行為。下列範本程式碼示範如何以 Java 設定此組態。

```
     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);
  }
}
```

索引和搜尋

Amazon Cloud Directory 支援兩種索引方法：以值為基礎及以類型為基礎。以值為基礎的索引是最常見的形式。透過此方法，您可以根據物件屬性值來編製目錄中的物件索引並進行搜尋。透過以類型為基礎的索引，您可以根據物件類型來編製目錄中的物件索引並進行搜尋。面向可協助定義物件類型。如需結構描述和面向的詳細資訊，請參閱「Schemas」和「Facets」。

Cloud Directory 中的索引可讓您依其他物件的屬性和面向值，輕鬆地列出這些物件。每個索引會在建立時定義，以搭配特定具名屬性或面向使用。例如，“Person” 面向的 “email” 屬性上可能會定義一個索引。索引是第一級物件，這表示用戶端可以根據應用程式邏輯的需求，彈性地建立、修改、列出及刪除索引。

在概念上，索引類似有子項的節點：連接子項時，索引節點的連結會根據已建立索引之屬性標示，而不是指定標籤。不過，索引連結不是父子邊緣，各有各的列舉 API 操作集。

請務必了解 Cloud Directory 中的索引不會自動填入，因為索引可能位於其他系統中。反之，您可以使用 API 呼叫直接將物件連接到索引，以及分離物件與索引。雖然此舉需要多費些工夫，但可讓您彈性地定義不同的索引範圍。例如，您可以定義索引，只追蹤特定節點的直系子項。或者，您可以定義索引，追蹤本機根目錄下指定分支中的所有物件，例如部門中的所有節點。您也可以同時執行這兩項作業。

主題

- 索引生命週期
- 以面向為基礎的索引
- 唯一與非唯一索引

索引生命週期

您可以使用下列 API 呼叫，協助進行索引的開發生命週期。

1. 您可以使用 `CreateIndex` API 呼叫建立索引。您可以提供索引定義結構，來描述索引將要追蹤之連接物件上的屬性。此定義也會指出索引是否應該強制執行唯一性。結果會是新索引的物件 ID，您應該立即將之連接到階層，如同其他物件一樣。例如，這可以是專門用來保留索引的分支。
2. 您可使用 `AttachToIndex` API 呼叫手動將物件連接到索引。此索引會接著在每個連接的物件上自動追蹤其所定義的屬性值。
3. 若要使用索引透過更有效率的列舉來搜尋物件，請呼叫 `ListIndex` 並視您所需指定值的範圍。

4. 使用 ListAttachedIndices API 呼叫，列舉連接到指定物件的索引。
5. 使用 DetachFromIndex API 呼叫，從索引中手動移除物件。
6. 一旦您分離所有物件與索引，您可以使用 DeleteObject API 呼叫刪除索引。

除了所有物件使用的空間限制以外，每個目錄中的索引數量沒有任何限制。索引及其連接會耗用空間，但類似於節點及父子連結所耗用的空間。可連接到指定物件的索引數量則有限。如需詳細資訊，請參閱 Amazon Cloud Directory 限制。

以面向為基礎的索引

透過以面向為基礎的索引和搜尋，您可以只搜尋部分的目錄，藉此最佳化您的目錄搜尋。若要執行此作業，您可以使用結構描述 面向。例如，與其搜尋目錄中的所有使用者物件，您可以改為搜尋只含有員工面向的使用者物件。此效率有助於降低查詢的延遲時間與所擷取的資料量。

透過以面向為基礎的索引，您可以使用 Cloud Directory 索引 API 操作來建立索引，並將之連接到物件的面向。您也可以列出索引結果，然後根據特定面向來篩選這些結果。由於這會將搜尋範圍縮小為只包含特定面向類型的物件，因此可有效地降低查詢次數與資料量。

用於 “facets” 與 CreateIndex API 呼叫的 ListIndex 屬性會呈現套用至物件的面向集合。此屬性僅適用於 CreateIndex 和 ListIndex API 呼叫。如下列範本程式碼所示，結構描述 ARN 使用目錄的區域、擁有者帳戶和目錄 ID 來參考 Cloud Directory 結構描述。此服務提供的結構描述不會出現在清單中。

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

例如，下列範本程式碼會針對您的 AWS 帳戶和目錄建立特定的以面向為基礎的索引，您可以在其中列舉使用面向 SalesDepartmentFacet 建立的所有物件。

> **Note**
>
> 請務必在參數中使用 “facets” 值，如下所示。範本程式碼中所示的「facets」執行個體，是指由 Cloud Directory 服務提供及控制的值。您可以使用這些值來編製索引，但只能進行唯讀存取。

```
// 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")
    )))
```

唯一與非唯一索引

唯一索引與非唯一索引的不同之處，在於會對連接到索引的物件強制執行已建立索引之屬性值的唯一性。例如，您可能想要將 Person 物件填入兩個索引：“email” 屬性上的唯一索引，以及 “lastname” 屬性上的非唯一索引。姓氏索引可連接到具有相同姓氏的多個 Person 物件。另一方面，以電子郵件索引為目標的 `AttachToIndex` 呼叫，會在已連接到具有相同 email 屬性的 Person 時，傳回 `LinkNameAlreadyInUseException` 錯誤。請注意，此錯誤不會移除 Person 物件本身。因此，應用程式可能會建立 Person、將之連接到階層，再連接到索引，全部都在單一批次請求中。如此可確保若任何索引違反唯一性，則會自動復原物件及其所有連接。

如何 Cloud Directory

本節列出所有操作和維護 Cloud Directory 環境的程序。

主題

管理您的目錄

本節說明如何維護 Cloud Directory 環境的常見目錄工作。

主題

建立您的目錄

在 Amazon Cloud Directory 中建立目錄之前，AWS Directory Service 需要您先對其套用結構描述。您無法建立沒有結構描述的目錄，而且一個目錄通常會套用一個結構描述。不過，您可以使用 Cloud Directory API 操作將其他結構描述套用至目錄。如需詳細資訊，請參閱「」ApplySchema中的Amazon Cloud Directory API 參考指南。

建立 Cloud Directory

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory中，選擇目錄。
2. 選擇設定 Cloud Directory。
3. 在選擇要套用至新目錄的綱要，請輸入您目錄的易記名稱，例如`User Repository`，然後選擇以下其中一個選項：

    - 受管結構描述
    - 範例結構描述
    - 自訂結構描述

範例結構描述和自訂結構描述放置在開發狀態。如需結構描述狀態的詳細資訊，請參閱「結構描述生命週期」。您必須將結構描述轉換成 Published (已發佈) 狀態，才能套用至目錄。若要使用主控台成功發佈範例結構描述，您必須具有下列動作的許可：

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

由於範例結構描述是由 AWS 提供的唯讀範本，因此無法直接發佈。反之，當您選擇根據範例結構描述建立目錄時，主控台會建立所選範例結構描述的暫存副本，並將它設定為開發狀態。接著會建立開發結構描述的副本，並將它設定為 Published (已發佈) 狀態。一旦發佈，即會刪除開發結構描述，因此 `DeleteSchema` 動作對發佈範例結構描述而言是必要的。

4. 選擇下一步。
5. 檢閱目錄資訊，並進行必要的變更。若資訊無誤，請選擇 Create (建立)。

刪除您的目錄

使用以下程序刪除 Cloud Directory 中的目錄。

> **Note**
>
> 您必須先停用它，之後您才能刪除目錄。如需說明，請參閱「停用您的目錄」。

刪除目錄

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取目錄。
2. 在表格中選取要刪除的目錄 ID 旁的選項。
3. 選擇 Actions (動作)。
4. 選擇刪除

5. 在 中刪除目錄對話方塊中，輸入目錄名稱來確認作業，然後選擇刪除。

停用您的目錄

使用以下程序停用 Cloud Directory 中的目錄。

停用目錄

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取目錄。
2. 在表格中選取要停用的目錄 ID 旁的選項。
3. 選擇 Actions (動作)。
4. 選擇Disable

啟用您的目錄

使用下列程序，在 Cloud Directory 中啟用先前停用的目錄。

啟用目錄

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取目錄。
2. 在表格中選取要啟用的目錄 ID 旁的選項。
3. 選擇 Actions (動作)。
4. 選擇啟用

管理您的結構描述

本節說明如何維護 Cloud Directory 環境的常見結構描述工作。

主題

- 建立您的結構描述
- 刪除結構描述
- 下載結構描述
- 發佈結構描述
- 更新您的結構描述

- 升級您的結構描述

建立您的結構描述

Amazon Cloud Directory 支援上傳相容的 JSON 檔案以建立結構描述。若要建立新的結構描述，您可以從頭開始建立自己的 JSON 檔案，或下載列於主控台中的其中一個現有結構描述。然後將它上傳做為自訂結構描述。如需詳細資訊，請參閱 自訂結構描述。

您也可以使用 Cloud Directory API 建立、刪除、下載、列出、發佈、更新和升級結構描述。如需結構描述 API 操作的詳細資訊，請參閱Amazon Cloud Directory API 參考指南。

根據您慣用的方法，選擇下列其中一個程序。

建立自訂結構描述

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory中，選擇Schemas。
2. 建立一個 JSON 檔案，其中包含您所有的新結構描述定義。如需如何格式化 JSON 檔案的詳細資訊，請參閱「JSON 結構描述格式」。
3. 在主控台，選擇Upload new schema。
4. 在 中Upload new schema對話方塊中，輸入結構描述的名稱。
5. 選擇選擇檔案，選取您剛建立的新 JSON 檔案，然後選擇開啟。
6. 選擇 Upload (上傳)。這會在您的結構描述程式庫中新增一個結構描述，並將其設為開發狀態。如需結構描述狀態的詳細資訊，請參閱「結構描述生命週期」。

在主控台中根據現有的結構描述建立自訂結構描述

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory中，選擇Schemas。
2. 在列出結構描述的表格中，選取您要複製的結構描述附的選項。
3. 選擇 Actions (動作)。
4. 選擇下載結構描述。
5. 將 JSON 檔案重新命名並視需要進行編輯，然後儲存檔案。如需如何格式化 JSON 檔案的詳細資訊，請參閱「JSON 結構描述格式」。
6. 在主控台，選擇Upload new schema，選取您剛編輯的 JSON 檔案，然後選擇開啟。

   這會在您的結構描述程式庫中新增一個結構描述，並將其設為開發狀態。如需結構描述狀態的詳細資訊，請參閱「結構描述生命週期」。

刪除結構描述

使用下列程序來刪除 Cloud Directory 中的結構描述。

刪除結構描述

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取Schemas。
2. 選取要刪除的結構描述名稱旁的選項。
3. 選擇 Actions (動作)。
4. 選擇刪除
5. 在 中刪除結構描述對話方塊中，選擇刪除。

下載結構描述

使用下列程序來下載結構描述。

下載結構描述

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取Schemas。
2. 在您要下載的結構描述名稱旁邊的表格中選取選項。
3. 選擇 Actions (動作)。
4. 選擇下載結構描述

發佈結構描述

使用下列程序在 Cloud Directory 中發佈結構描述。

發佈資料架構的步驟

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取Schemas。
2. 在您要發佈的綱要名稱旁邊的表格中選取選項。
3. 選擇 Actions (動作)。
4. 選擇發布
5. 在 中發佈結構描述對話方塊中，提供下列資訊：

    a. 結構描述名稱

b. 主要版本

c. 次要版本

6. 選擇 Publish (發佈)。

更新您的結構描述

使用下列程序來更新 Cloud Directory 中的結構描述。

更新資料架構的步驟

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取Schemas。
2. 在您要更新的架構名稱旁邊的表格中選取選項。
3. 選擇 Actions (動作)。
4. 選擇 Update (更新)
5. 在 中更新結構描述對話方塊中，選擇性地修改結構描述名稱，或選取選擇檔案來套用或移除面向和屬性。
6. 選擇 Update (更新)。

升級您的結構描述

升級架構會將您選擇的 Facet 和屬性新增到您選取的已發佈架構中。請使用下列程序來升級已發佈的結構描述。

升級結構描述

1. 在 中AWS Directory Service導覽窗格, 在Cloud Directory，選取Schemas。
2. 在要升級的綱要名稱旁邊的表格中選取選項。
3. 選擇 Actions (動作)。
4. 選擇升級
5. 在 中升級發佈結構描述對話方塊中，選擇下列其中一個選項，然後選擇升級：

   - 從目前的開發結構描述清單中選擇
   - 上傳新的結構描述檔案 (JSON)

6. 選擇升級。

Amazon Cloud Directory 中的安全

雲端安全是 AWS 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足對安全最為敏感的組織需求。

安全是 AWS 與您共同肩負的責任。共同的責任模型 將此描述為雲端 本身 的安全和雲端 內部 的安全：

- 雲端安全性 – AWS 負責保護在 AWS 雲端中執行 AWS 服務的基礎設施。AWS 也會提供您可以安全使用的服務。在 AWS 合規計劃中，第三方稽核員會定期測試並驗證我們的安全功效。若要進一步了解適用於 Amazon Cloud Directory 的合規計劃，請參閱合規計劃的 AWS 服務範圍。
- 雲端內部安全 – 您的責任取決於您所使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件有助於您了解如何在使用 Cloud Directory 時套用共同責任模型。下列主題顯示如何設定 Cloud Directory 以達到您的安全性和合規目標。您也將了解如何使用其他 AWS 服務，幫助您監控並保護 Cloud Directory 資源。

主題

- Amazon Cloud Directory 中的 Identity and Access Management
- 在 Amazon Cloud Directory 中進行記錄和監控
- Amazon Cloud Directory 的合規驗證
- Amazon Cloud Directory 中的復原功能
- Amazon Cloud Directory 中的基礎設施安全

Amazon Cloud Directory 中的 Identity and Access Management

存取 Amazon Cloud Directory 需要登入資料，以供 AWS 驗證您的請求。這些登入資料必須具備許可，才能存取 AWS 資源。下列區段提供了詳細資訊，說明您可如何使用AWS Identity and Access Management (IAM)和 Cloud Directory Directory，藉由控制誰能夠存取各項資源，協助保護您的資源：

- Authentication
- 存取控制

Authentication

您可以使用下列任一種身分類型存取 AWS：

- AWS 帳戶根使用者 - 當您初建立 AWS 帳戶時，您一開始具有單一的登入身分，可以完整存取帳戶的所有 AWS 服務與資源。此身分稱為 AWS 帳戶「根使用者」，是藉由您用來建立帳戶的電子郵件地址和密碼以登入並存取。強烈建議您不要以根使用者處理日常作業，即使是管理作業。反之，請遵循僅以根使用者建立您第一個 &IAM; 使用者的最佳實務。接著請妥善鎖定根使用者登入資料，只用來執行少數的帳戶與服務管理作業。
- IAM 使用者— 一個IAM 使用者是您 AWS 帳戶中的一種身分，擁有特定的自訂許可 (例如，在 Cloud Directory 中建立目錄的許可)。您可以使用 IAM 使用者名稱和密碼登入安全的 AWS 網頁，例如 AWS 管理主控台、AWS 開發論壇或 AWS 支援中心。

  除了使用者名稱和密碼之外，您也可以為每個使用者產生存取金鑰。無論是透過數個軟體開發套件中的一個或使用 AWS 命令列界面 (CLI)，以程式設計方式存取 AWS 服務時，您都可以使用這些金鑰。此開發套件和 CLI 工具使用存取金鑰，以加密方式簽署您的請求。如果您未使用 AWS 工具，則必須自行簽署請求。Cloud Directory 支援簽章版本 4，這是用來驗證傳入 API 請求的協定。如需驗證請求的詳細資訊，請參閱簽章版本 4 簽署程序中的AWS 一般參考資料。

- IAM 角色 - IAM 角色是您可以在帳戶中建立的另一種 IAM 身分，具有特定的許可。IAM 角色類似於 IAM 使用者，因為同樣是 AWS 身分，也有許可政策可決定該身分在 AWS 中可執行和不可執行的操作。但是，角色的目的是讓需要它的任何人可代入，而不是單獨地與某個人員關聯。此外，角色沒有與之關聯的標準長期憑證，例如密碼或存取金鑰。反之，當您擔任角色時，其會為您的角色工作階段提供臨時安全性登入資料。使用臨時登入資料的 IAM 角色在下列情況中非常有用：

  - 聯合身分使用者存取 - 非建立 IAM 使用者，而是使用來自 AWS Directory Service、您的企業使用者目錄或 Web 身分供應商的現有身分。這些稱為「聯合身分使用者」。透過身份供應商請求存取時，AWS 會將角色指派給聯合身份使用者。如需聯合身分使用者的詳細資訊，請參閱聯合身分使用者和角色中的IAM 使用者指南。

  - AWS 服務存取 - 服務角色是服務擔任的 IAM 角色，可代您執行動作。服務角色提供的存取權僅限在您的帳戶內，不能用來授予存取其他帳戶中的服務。IAM 管理員可以從 IAM 內建立、修改和刪

除服務角色。如需詳細資訊，請參閱「」建立角色以將許可委派給 AWS 服務中的IAM 使用者指南。

- 在 Amazon EC2 上執行的應用程式 - 針對在 EC2 執行個體上執行並提出 AWS CLI 和 AWS API 請求的應用程式，您可以使用 IAM 角色來管理臨時登入資料。這是在 EC2 執行個體內存放存取金鑰的較好方式。若要指派 AWS 角色給 EC2 執行個體並提供其所有應用程式使用，您可以建立連接到執行個體的執行個體描述檔。執行個體描述檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時登入資料。如需詳細資訊，請參閱「」使用 IAM 角色為在 Amazon EC2 執行個體上執行的應用程式授予許可中的IAM 使用者指南。

存取控制

您可以持有效登入資料為自己的要求進行身份驗證，但還須具備許可才能建立或存取 Cloud Directory 資源。例如，您必須具有許可才能建立 Amazon Cloud Directory。

下節說明如何管理 Cloud Directory 的許可。我們建議您先閱讀概觀。

- 管理您的 Cloud Directory 資源存取許可概觀
- 在 Cloud Directory 使用以身分為基礎的政策 (IAM 政策)
- Amazon Cloud Directory 許可：動作、資源和條件參考

管理您的 Cloud Directory 資源存取許可概觀

每項 AWS 資源均由某個 AWS 帳戶所擁有，而建立或存取資源的許可則由許可政策管理。帳戶管理員可以將許可政策連接到 IAM 身分 (即使用者、群組與角色) 以及某些服務 (例如 AWS Lambda)，也支援將許可政策連接到資源。

> **Note**
>
> 帳戶管理員 (或管理員使用者) 是具有管理員權限的使用者。如需詳細資訊，請參閱 IAM 最佳實務 (在 IAM 使用者指南 中)。

當您授予許可時，能夠決定取得許可的對象、這些對象取得許可的資源，以及可對上述資源進行的特定動作。

主題

Cloud Directory 資源與作業

在 Cloud Directory 中，主要資源是目錄和模式。這些資源各與唯一的 Amazon Resource Name (ARN) 相關聯，如下表所示。

資源類型	ARN 格式
目錄	`arn:aws:clouddirectory:` *`region`*`:`*`account-id`* `:directory/` *`directory-id`*
結構描述	`arn:aws:clouddirectory:` *`region`*`:`*`account-id`* `:schema/`*`schema-state`* `/`*`schema-name`*

如需結構描述狀態和 ARN 的詳細資訊，請參閱ARN 範例中的Amazon Cloud Directory API 參考。

Cloud Directory 提供一組操作，供您使用適當的資源。如需可用操作的清單，請參閱Amazon Cloud Directory 動作或Directory Service 動作。

了解資源所有權

資源擁有者即建立資源的 AWS 帳戶。換言之，資源擁有者就是驗證建立資源請求之 委託人實體(根帳戶、IAM 使用者或 IAM 角色) 的 AWS 帳戶。下列範例說明其如何運作：

- 如果您使用 AWS 帳戶的根帳戶登入資料建立 Cloud Directory 資源 (如目錄)，您的 AWS 帳戶就是該資源的擁有者。

- 如果您在自己的 AWS 帳戶中建立 IAM 使用者，並將建立 Cloud Directory 資源的許可授予該使用者，則該使用者也可建立 Cloud Directory 資源。但是您的 AWS 帳戶 (即該使用者所屬帳戶) 為該 資源的擁有者。
- 如果您在自己的 AWS 帳戶中建立 IAM 角色，並授予該角色建立 Cloud Directory 資源的許可，則任何可擔任該角色的人都能建立 Cloud Directory 資源。您的 AWS 帳戶 (即該角色所屬帳戶) 擁有 Cloud Directory 資源。

管理資源存取

許可政策描述誰可以存取哪些資源。下一節說明可用來建立許可政策的選項。

> **Note**
>
> 本節將著重討論如何在 Cloud Directory 中使用 IAM。它不提供 IAM 服務的詳細資訊。如需完整的 IAM 文件，請參閱什麼是 IAM？中的IAM 使用者指南。如需 IAM 政策語法和說明的詳細資訊，請參閱AWS IAM 參考中的IAM 使用者指南。

連接到 IAM 身分的政策稱為身分類型政策 (IAM 政策) 和連接到資源的政策稱為以資源為基礎的政策。Cloud Directory 僅支援以身分為基礎的政策 (IAM 政策)。

主題

- 身分類型政策 (IAM 政策)
- 資源類型政策

身分類型政策 (IAM 政策)

您可以將政策連接到 IAM 身分。例如，您可以執行下列操作：

- 將許可政策連接至您帳戶中的使用者或群組-帳戶管理員可使用與特定使用者相關聯的許可政策，來授予該使用者建立 Cloud Directory 資源 (例如新目錄) 的許可。
- 將許可政策連接至角色 (授予跨帳戶許可)-您可以將以身分為基礎的許可政策連接至 IAM 角色，以授予跨帳戶許可。例如，帳戶 A 中的管理員可以建立角色，將跨帳戶許可授與其他 AWS 帳戶 (例如，帳戶 B) 或下列 AWS 服務：
  1. 帳戶 A 管理員建立 IAM 角色，並將許可政策連接到可授與帳戶 A 中資源許可的角色。
  2. 帳戶 A 管理員將信任政策連接至該角色，識別帳戶 B 做為可擔任該角的委託人。

3. 帳戶 B 管理員即可將擔任該角色的許可委派給帳戶 B 中的任何使用者。這麼做可讓帳戶 B 的使用者建立或存取帳戶 A 的資源。如果您想要授與 AWS 服務許可以擔任該角色，則信任政策的主體可以是 AWS 服務主體。

如需使用 IAM 委派許可的詳細資訊，請參閱存取管理中的IAM 使用者指南。

下列許可政策會授予使用者執行開頭為 `Create` 之所有動作的許可。這些動作會顯示 Cloud Directory 資源 (如目錄或綱要) 的相關資訊。請注意，萬用字元 (*)`Resource`元素表示可對帳戶擁有的所有 Cloud Directory 資源執行動作。

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

如需搭配 Cloud Directory 使用以身分為基礎的政策的詳細資訊，請參閱在 Cloud Directory 使用以身分為基礎的政策 (IAM 政策)。如需使用者、群組、角色和許可的詳細資訊，請參閱身分 (使用者、群組和角色)中的IAM 使用者指南。

資源類型政策

其他服務 (例如 Amazon S3) 也支援以資源為基礎的許可政策。例如，您可以將政策連接至 S3 儲存貯體，以管理該儲存貯體的存取許可。Cloud Directory 不支援以資源為基礎的政策。

指定政策元素：動作、效果、資源和委託人

針對每個 Cloud Directory 資源 (請參閱Cloud Directory 資源與作業)，該服務會定義一組 API 操作。如需可用的 API 操作清單，請參閱Amazon Cloud Directory 動作或Directory Service 動作。為了授予這些 API 操作的許可，Cloud Directory 會定義一組您可以在政策中指定的動作。請注意，執行 API 操作可能需要多個動作的許可。

以下是基本的政策元素：

- 資源 – 在政策中，您可以使用 Amazon Resource Name (ARN) 來識別要套用政策的資源。對於 Cloud Directory 資源，則一律在 IAM 政策中使用萬用字元 (*)。如需詳細資訊，請參閱 Cloud Directory 資源與作業。
- 動作 - 您使用動作關鍵字識別您要允許或拒絕的資源操作。例如，`clouddirectory:GetDirectory`許可允許使用者執行 Cloud Directory`GetDirectory`operation.
- 效果— 您可指定當使用者要求特定動作時會有什麼效果 — 這可以是允許或拒絕。如果您未明確授予存取 (允許) 資源，則隱含地拒絕存取。您也可以明確拒絕資源存取，這樣做可確保使用者無法存取資源，即使不同政策授予存取也是一樣。
- 委託人 - 在以身分為基礎的政策 (IAM 政策) 中，政策所連接的使用者就是隱含委託人。對於資源類型政策，您可以指定想要收到許可的使用者、帳戶、服務或其他實體 (僅適用於資源類型政策)。Cloud Directory 不支援以資源為基礎的政策。

如需進一步了解 IAM 政策語法和說明，請參閱AWS IAM 參考中的IAM 使用者指南。

如需詳列所有 Amazon Cloud Directory API 動作及適用資源的資料表，請參閱Amazon Cloud Directory 許可：動作、資源和條件參考。

在政策中指定條件

當您授予許可時，可以使用存取原則語言來指定政策應該何時生效的條件。例如，建議只在特定日期之後套用政策。如需使用政策語言指定條件的詳細資訊，請參閱Condition中的IAM 使用者指南。

欲表示條件，您可以使用預先定義的條件金鑰。Cloud Directory 沒有專屬的條件金鑰。不過，您可以使用適合的全 AWS 條件鍵。如需全 AWS 鍵的完整清單，請參閱可用的全球條件金鑰中的IAM 使用者指南。

在 Cloud Directory 使用以身分為基礎的政策 (IAM 政策)

這個主題提供以身分為基礎的政策範例，在該政策中帳戶管理員可以將許可政策連接至 IAM 身分 (即使用者、群組和角色)。

> **Important**
>
> 建議您先檢閱可供您管理 Cloud Directory 資源存取之基本念與選項的說明介紹主題。如需詳細資訊，請參閱 管理您的 Cloud Directory 資源存取許可概觀。

本主題中的各節涵蓋下列內容：

- 使用 AWS Directory Service 主控台所需的許可
- 適用於 Amazon Cloud Directory 的 AWS 受管 (預先定義) 政策

使用 AWS Directory Service 主控台所需的許可

若使用者要使用 AWS Directory Service 主控台，該使用者必須具有上述政策所列的許可，或「」中說明之 Directory Service 完整存取角色或 Directory Service 唯讀角色授予的許可。適用於 Amazon Cloud Directory 的 AWS 受管 (預先定義) 政策。

如果您建立比最基本必要許可更嚴格的 IAM 政策，則對於採取該 IAM 政策的使用者而言，主控台就無法如預期運作。

適用於 Amazon Cloud Directory 的 AWS 受管 (預先定義) 政策

AWS 獨立的 IAM 政策由 AWS 所建立與管理，可用來解決許多常用案例。受管政策授與常見使用案例中必要的許可，讓您免於查詢需要哪些許可。如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS Managed Policies (AWS 受管政策)。

下列 AWS 受管政策 (您可以將這些政策連接到您帳戶中的使用者) 專用於 Amazon Cloud Directory：

- 亞馬遜雲端目錄僅限讀取-授予使用者或群組所有 Amazon Cloud Directory 資源的唯讀存取權。如需詳細資訊，請參閱 AWS 管理主控台中的政策頁面。
- 亞馬遜雲端目錄完整存取— 授予使用者或群組 Amazon Cloud Directory 的完整存取權。如需詳細資訊，請參閱 AWS 管理主控台中的政策頁面。

此外，還有適合與其他 IAM 角色搭配使用的其他 AWS 受管政策。這些政策必須指派給與您 Amazon Cloud Directory 相關聯的角色，才能讓這些使用者存取其他 AWS 資源，如 Amazon EC2。

您也可以建立自訂 IAM 政策，讓使用者可存取所需的 API 動作和資源。您可以將這些自訂政策連接至需要這些許可的 IAM 使用者或群組。

Amazon Cloud Directory 許可：動作、資源和條件參考

當您在設定 存取控制 並撰寫可連接到 IAM 身分 (以身分為基礎的政策) 的許可政策時，可以使用下列資料表做為參考。所以此 清單包括每個 Amazon Cloud Directory Directory API 操作、您可以授予執行動作許可的相應動作，以及您可以授予許可的 AWS 資源。您要在政策的 `Action` 欄位中指定動作，並在政策的 `Resource` 欄位中指定資源值。

您可以在 Amazon Cloud Directory Directory 政策中使用全 AWS 條件金鑰來表達條件。如需全 AWS 鍵的完整清單，請參閱可用的全球條件金鑰中的IAM 使用者指南。

> **Note**
>
> 若要指定動作，請使用後接 API 操作名稱的 `clouddirectory:` 字首 (例如，`clouddirectory:CreateDirectory`)。

在 Amazon Cloud Directory 中進行記錄和監控

最佳實務是應該監控您的目錄，以確保所做的變更都會記錄。這可協助您確保可對任何未預期的變更進行調查，或可還原不想要的變更。Amazon Cloud Directory 前支援 AWS CloudTrail，您可以使用它來監控您的目錄和任何相關的活動。

如需詳細資訊，請參閱「」使用 CloudTrail 記錄 Cloud Directory API 呼叫。

Amazon Cloud Directory 的合規驗證

在多個 AWS 合規計畫中，第三方稽核人員會評估 Amazon Cloud Directory 的安全與合規。這些計劃包括 ISO、SOC、PCI、FedRAMP、HIPAA 等。

如需特定合規計劃範圍內的 AWS 服務清單，請參閱合規計劃的 AWS 服務範圍。如需一般資訊，請參閱 AWS 合規計劃。

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱在 AWS Artifact 中下載報告。

您使用 Cloud Directory 的合規責任，取決於資料的機密性、您公司的合規目標及適用法律和法規。AWS 提供下列資源，以協助合規：

- 安全與合規快速入門指南— 這些部署指南討論在 AWS 上部署以安全及合規為重心基準環境的架構考量和步驟。
- HIPAA 安全與合規架構白皮書 – 本白皮書說明公司可如何運用 AWS 來建立 HIPAA 合規的應用程式。
- AWS 合規資源 - 此系列工作手冊和指南可能適用於您的產業和地點。
- AWS Config - 此 AWS 服務評定您的資源組態符合內部實務、產業準則和法規的程度。
- AWS 安全中樞 – 此 AWS 服務可供您檢視 AWS 中的安全狀態，可助您檢查是否符合安全產業標準和最佳實務。

Amazon Cloud Directory 中的復原功能

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。AWS 區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援網路連線相互連結。Cloud Directory 建立在這些原則上，可在多個 AWS 區域中使用，這些區域實際上彼此隔離。在每個區域內，透過至少三個可用區域進一步支援服務，將任何單一可用區域無法使用而造成的服務停機時間降至最低。

如需 AWS 區域與可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

Amazon Cloud Directory 中的基礎設施安全

Amazon Cloud Directory 是受到 AWS 全球網路安全程序所保護的受管服務，如Amazon Web Services：安全程序概觀白皮書。

您可使用 AWS 發佈的 API 呼叫，透過網路存取 Cloud Directory。用戶端必須支援 Transport Layer Security (TLS)。建議使用 TLS 1.2 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需 FIPS 和 FIPS 端點的詳細資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

此外，請求必須使用存取金鑰 ID 和與 IAM 委託人相關聯的私密存取金鑰來簽署。或者，您可以使用 AWS Security Token Service (AWS STS) 來產生暫時安全登入資料來簽署請求。

交易支援

使用 Amazon Cloud Directory，您通常需要新增物件或新增新物件與現有物件之間的關係，來反映實際階層的變更。批次操作可提供下列好處，讓這類目錄任務更易於管理：

- 批次操作可降低對您目錄寫入和讀取物件所需的往返數量，而提升應用程式的整體效能。
- 批次寫入提供與 SQL 資料庫同等的交易語意。所有操作皆會成功完成，如果任一項操作失敗，就不會套用這些操作。
- 使用批次參考，您可以建立物件，並使用對新物件的參考執行進一步的動作，例如將其新增至關係，以及先降低使用讀取操作的成本再執行寫入操作。

BatchWrite

使用 BatchWrite 操作可對目錄執行多個寫入操作。所有批次寫入操作皆循序執行。其運作方式與 SQL 資料庫交易類似。如果批次寫入內其中一個操作失敗，對目錄執行的整個批次寫入就無效。如果批次寫入失敗，會發生批次寫入例外狀況。例外狀況包含失敗操作的索引，以及例外狀況類型和訊息。此資訊可協助您找出失敗的根本原因。

批次寫入支援下列 API 操作：

- AddFacetToObject
- AttachObject
- AttachPolicy
- AttachToIndex
- AttachTypedLink
- CreateIndex
- CreateObject
- DeleteObject
- DetachFromIndex
- DetachObject
- DetachTypedLink
- RemoveFacetFromObject

- UpdateObjectAttributes

批次參考名稱

當您需要在中介批次操作中參考物件時，僅批次寫入支援批次參考名稱。例如，假設在指定的批次寫入中，先分離 10 個不同的物件，再將其連接到目錄的不同部分。如果沒有批次參考，您就必須讀取所有 10 個物件參考，再於批次寫入中提供該參考作為重新連接過程中的輸入。您可以在連接的過程中使用批次參考識別分離的資源。批次參考可以是任一種開頭為井字號 (#) 的一般字串。

例如，在下列程式碼範本中，連結名稱為 "this-is-a-typo" 的物件會與根分離，其批次參考名稱為 "ref"。接著同一個物件會連接到連結名稱為 "correct-link-name" 的根。並將子參考設定為批次參考來識別該物件。如果沒有批次參考，您需要先取得分離的 objectIdentifier，然後在連接過程中提供給子參考。您可以使用批次參考名稱避免這種額外讀取的情況。

```
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

使用 BatchRead 操作可對目錄執行多個讀取操作。例如，在下列程式碼範本中，我們讀取了參考為 “/managers” 之物件的子系，同時在單一批次讀取中讀取了參考為 “/managers/bob” 之物件的屬性。

```
BatchListObjectChildren listObjectChildrenRequest = new BatchListObjectChildren()
        .withObjectReference(new ObjectReference().withSelector("/managers"));
BatchListObjectAttributes listObjectAttributesRequest = new BatchListObjectAttributes()
        .withObjectReference(new ObjectReference().withSelector("/managers/bob"));
BatchReadRequest batchRead = new BatchReadRequest()
        .withConsistencyLevel(ConsistencyLevel.SERIALIZABLE)
        .withDirectoryArn(directoryArn)
```

```
        .withOperations(new ArrayList(Arrays.asList(listObjectChildrenRequest,
 listObjectAttributesRequest)));
BatchReadResult result = cloudDirectoryClient.batchRead(batchRead);
```

BatchRead 支援下列 API 操作：

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

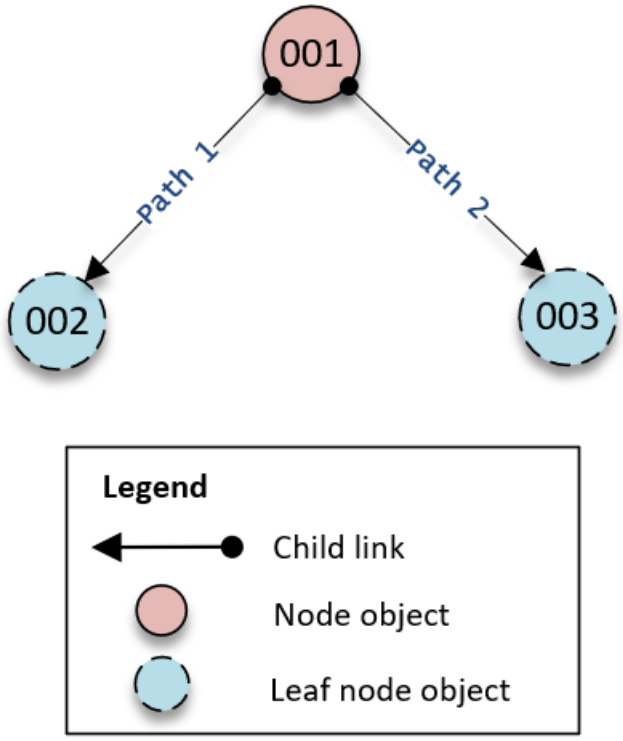
批次操作的限制

無論請求內含多少操作數，向伺服器發出的每個請求 (包括批次請求) 均有可操作的資源數上限。這可讓您極彈性地撰寫批次請求，只要不超過資源上限即可。如需資源上限的詳細資訊，請參閱「Amazon Cloud Directory 限制」。

限制的計算方式為將批次中各個操作的寫入或讀取相加。例如，目前每個 API 呼叫的讀取操作限制為 200 個物件。假如您想撰寫新增 9 個 ListObjectChildren API 呼叫的批次，且每個呼叫均需讀取 20 個物件。由於讀取物件總數 (9 x 20 = 180) 未超過 200 個，因此批次操作會成功。

計算寫入操作的概念也是如此。例如，目前的寫入操作限制為 20。如果您將批次設定為新增 2 個 UpdateObjectAttributes API 呼叫，且每個呼叫均執行 9 個寫入操作，這也會成功。無論哪一種情況，只要批次操作超出限制，操作就會失敗，並會拋出 `LimitExceededException`。

計算批次內物件數的正確方式是同時包含實際的節點或 leaf_node 物件，如果使用路徑型方法來反覆操作您的樹狀目錄，您還需要包含批次內反覆操作的每個路徑。例如，下圖所示為基本樹狀目錄，若要讀取物件 `003` 的屬性值，則物件的總讀取計數為 3。



沿著樹狀目錄向下周遊讀取的運作方式如下：

1. 讀取物件的 001 物件，以判斷物件 003 的路徑。

2. 沿著 Path 2 向下移動

3. 讀取物件 003

同樣地，在屬性數量方面，我們也需計算物件 001 和 003 中的屬性數量，以確保不會達到限制。

例外狀況處理

Cloud Directory 中的 Batch 操作有時會失敗。在這些情況下，了解如何處理這類失敗非常重要。解決寫入操作失敗的方法與解決讀取操作失敗的方法不同。

批次寫入操作失敗

如果批次寫入操作失敗，Cloud Directory 就無法執行整個批次操作，並會傳回例外狀況。例外狀況內含失敗操作的索引，以及例外狀況類型和訊息。若發生 RetryableConflictException，您可以使用指數退避重試。執行此作業的簡單方式是在每次收到例外狀況或失敗後，加倍等待的時間。例如，如果您的第一個批次寫入操作失敗，請等待 100 毫秒後，再重試請求。如果第二個請求失敗，請等待 200 毫秒後再重試。如果第三個請求失敗，請等待 400 毫秒後再重試。

批次讀取操作失敗

如果批次讀取操作失敗，回應會包含成功回應或例外狀況回應。個別批次讀取操作失敗並不會導致整個批次讀取操作失敗，Cloud Directory 會針對每個操作，傳回個別的成功或失敗回應。

相關 Cloud Directory 部落格文章

- 使用 Batch 操作在 Amazon Cloud Directory 中寫入和讀取多個物件
- 如何在 Amazon Cloud Directory 中使用 Batch 參考來參考 Batch 請求中的新物件

Amazon Cloud Directory 合規服務

Amazon Cloud Directory 已針對下列標準展開稽核，可在您需要取得合規認證時做為解決方案的一部分。

Amazon Cloud Directory 符合美國聯邦風險與授權管理計劃 (FedRAMP) 安全性規定，並已獲得 FedRAMP 聯合授權委員會 (JAB) 依 FedRAMP 中度基準所核發的臨時操作授權 (P-ATO)。如需 FedRAMP 合規的詳細資訊，請參閱 FedRAMP 合規。

Amazon Cloud Directory 具備服務供應商第 1 級的支付卡產業 (PCI) 資料安全標準 (DSS) 3.2 版的合規聲明文件。使用 AWS 產品和服務存放、處理或傳輸持卡人資料的客戶，可以使用 Cloud Directory 來管理自己的 PCI DSS 合規認證。如需 PCI DSS 的詳細資訊，包括如何索取 AWS PCI 合規套裝服務的複本，請參閱 PCI DSS 第 1 級。

AWS 已擴大其 Health 保險流通與責任法案 (HIPAA) 合規計劃，並加入 Amazon Cloud Directory 做為HIPAA 合格服務。如果您與 AWS 簽署了執行商業夥伴協定 (BAA)，則可以使用 Cloud Directory 協助建立符合 HIPAA 標準的應用程式。AWS 提供以 HIPAA 為中心的白皮書，讓想要進一步了解如何利用 AWS 處理和存放健康資訊的客戶查閱。如需詳細資訊，請參閱 HIPAA 合規。

	Amazon Cloud Directory 已成功完成 ISO/IEC 27001、ISO/IEC 27017、ISO/IEC 27018 及 ISO 9001 的合規認證。如需詳細資訊，請參閱「ISO 27001」、「ISO 27017」、「ISO 27018」及「ISO 9001」。
	系統與組織控制 (SOC) 報告是獨立的第三方檢驗報告，其中展現了 Amazon Cloud Directory 如何達成關鍵合規控制與目標。您和稽核人員可以透過這些報告，了解為了支援操作與合規所建立的 AWS 控制。如需詳細資訊，請參閱 SOC 合規。

共同的責任

安全 (包括 HIPAA 及 PCI 合規) 是共同的責任。請務必了解，Cloud Directory 合規狀態不會自動套用到 AWS 雲端中執行的應用程式。您必須確保 AWS 服務的使用符合標準。

使用 Cloud Directory API

Amazon Cloud Directory 包含一組 API 操作，可讓您以程式設計方式存取 Cloud Directory 功能。您可以使用Amazon Cloud Directory API 參考指南，了解如何對 Cloud Directory API 發出請求，以建立及管理各種元素。它也涵蓋請求元件、回應內容，以及如何驗證請求。

Cloud Directory 提供所有必要的 API 操作，可讓開發人員建立新的應用程式。它提供 API 呼叫的下列類別：

- 適用於結構描述的建立、讀取、更新、刪除 (CRUD)
- 適用於面向的 CRUD
- 適用於目錄的 CRUD
- 適用於物件 (節點、政策等) 的 CRUD
- 適用於索引定義的 CRUD
- 批次讀取、批次寫入

API 的計費方 Cloud Directory

API 呼叫的計費會因所進行之 API 呼叫的特定類型而有所不同。最終一致讀取 API 呼叫、強式一致讀取 API 呼叫與寫入 API 呼叫會有特定的計費費率。中繼資料 API 呼叫是免費的。

強式一致操作可在讀取值時用於寫入後讀取一致性。最終一致操作可在更新執行時用於擷取值。在最終一致操作中，由於您讀取值的特定主機仍在處理更新，因此擷取的結果可能不是最準確的。不過，當您擷取效能呼叫時，這類讀取操作的延遲會很低。

當您從 Cloud Directory 讀取資料時，您必須指定最終一致讀取或強式一致讀取類型操作。讀取類型取決於一致性層級。這兩個一致性層級為適用於最終一致讀取的 EVENTUAL 與適用於強式一致讀取的 SERIALIZABLE。如需詳細資訊，請參閱 一致性層級。

下表列出所有 Cloud Directory API 及其如何影響您 AWS 帳戶的計費。

API	最終一致讀取 [1]	強式一致讀取 [2]	寫入 [3]	中繼資料 [4]
AddFacetToObject			X	
ApplySchema				X

API	最終一致讀取 [1]	強式一致讀取 [2]	寫入 [3]	中繼資料 [4]
AttachObject			X	
AttachPolicy			X	
AttachToIndex			X	
AttachTypedLink			X	
BatchRead	X	X		
BatchWrite			X	
CreateDirectory			X	
CreateFacet				X
CreateIndex			X	
CreateObject			X	
CreateSchema				X
CreateTypedLinkFacet				X
DeleteDirectory				X
DeleteFacet				X
DeleteObject			X	
DeleteSchema				X
DetachFromIndex			X	
DetachObject			X	
DetachPolicy			X	

API	最終一致讀取 [1]	強式一致讀取 [2]	寫入 [3]	中繼資料 [4]
DetachTypedLink			X	
DeleteTypedLinkFacet				X
DisableDirectory				X
EnableDirectory			X	
GetAppliedSchemaVersion				X
GetDirectory				X
GetFacet				X
GetLinkAttributes	X	X		
GetObjectAttributes	X	X		
GetObjectInformation	X	X		
GetSchemaAsJson				X
GetTypedLinkFacetInformation				X
ListAppliedSchemaArns				X
ListAttachedIndices	X	X		

API	最終一致讀取 [1]	強式一致讀取 [2]	寫入 [3]	中繼資料 [4]
ListDevelopmentSchemaArns				X
ListDirectories				X
ListFacetAttributes				X
ListFacetNames				X
ListIncomingTypedLinks	X	X		
ListIndex	X	X		
ListManagedSchemaArns				X
ListObjectAttributes	X	X		
ListObjectChildren	X	X		
ListObjectParentPaths	X			
ListObjectParents	X	X		
ListObjectPolicies	X	X		
ListOutgoingTypedLinks	X	X		

API	最終一致讀取 [1]	強式一致讀取 [2]	寫入 [3]	中繼資料 [4]
ListPolicyAttachments	X	X		
ListPublishedSchemaArns				X
ListTagsForResource				X
ListTypedLinkFacetAttributes				X
ListTypedLinkFacetNames				X
LookupPolicy	X			
PublishSchema				X
PutSchemaFromJson				X
RemoveFacetFromObject			X	
TagResource				X
UntagResource				X
UpdateFacet				X
UpdateLinkAttributes			X	
UpdateObjectAttributes			X	

API	最終一致讀取 [1]	強式一致讀取 [2]	寫入 [3]	中繼資料 [4]
UpdateSchema				X
UpdateTypedLinkFacet				X
UpgradeAppliedSchema				X
UpgradePublishedSchema				X

[1] 最終一致讀取 API 會以 EVENTUAL 一致性層級呼叫

[2] 強式一致讀取 API 會以 SERIALIZABLE 一致性層級呼叫

[3] 寫入 API 會依寫入 API 呼叫計費

[4] 中繼資料 API 不會收取費用，但會歸類為中繼資料 API 呼叫

如需計費方式的其他資訊，請參閱Amazon Cloud Directory 定價。

Amazon Cloud Directory 限制

以下是 Cloud Directory 的預設限制。除非另有說明，否則每個限制皆依區域規定。

Amazon Cloud Directory

結構描述和目錄限制

限制/概念	數量
每個面向的屬性數目 (包括必要)	1000
每個物件的面向數目	5
物件連接的唯一索引數目	3
每個結構描述的面向數目	30
每個屬性的規則數目	5
每個面向具預設值的屬性數目	10
每個面向的必要屬性數目	30
開發結構描述的數目	20
已發佈結構描述的數目	20
已套用結構描述的數目	5
目錄的數目	100
頁面元素上限	30
輸入大小上限 (總合所有輸入)	200 KB
回應大小上限 (總合所有輸出)	1 MB
結構描述 JSON 檔案大小限制	200 KB
面向名稱長度	64 UTF-8 編碼的位元組

限制/概念	數量
目錄名稱長度	64 UTF-8 編碼的位元組
結構描述名稱長度	64 UTF-8 編碼的位元組

物件限制

限制/概念	數量
已寫入物件的數目	每個 API 呼叫 20 個
已讀取物件的數目	每個 API 呼叫 200 個
已寫入屬性值的數目	每個 API 呼叫 1000 個
已讀取屬性值的數目	每個 API 呼叫 1000 個
路徑深度	15
輸入大小上限 (總合所有輸入)	200 KB
回應大小上限 (總合所有輸出)	1 MB
政策大小限制	10 KB
刪除物件期間，可刪除的屬性數量	30
類型連結身分屬性的彙總值長度	64 UTF-8 編碼的位元組
邊緣或連結名稱長度	64 UTF-8 編碼的位元組
已建立索引之屬性的值長度	512 UTF-8 編碼的位元組
未建立索引之屬性的值長度	2 KB
連接到物件的政策數目	4

批次操作的限制

您在批次內可呼叫的操作數目不限。如需詳細資訊，請參閱 批次操作的限制。

無法修改的限制

無法變更或增加數目的 Amazon Cloud Directory 限制包括：

- 面向名稱長度
- 目錄名稱長度
- 結構描述名稱長度
- 頁面元素上限
- 邊緣或連結名稱長度
- 已建立索引之屬性的值長度

Cloud Directory 資源

下表列有相關實用資源，有助您使用此服務。

Cloud Directory 入門	連結
Cloud Directory 網路研討會	https://www.youtube.com/watch?v=UANm3DC_IxE
Cloud Directory 範例 Java 程式碼	https://github.com/aws-samples/AmazonCloudDirectory-sample

Cloud Directory 部落格文章	描述
How to rapidly develop applications on Amazon Cloud Directory with Managed Schema	此部落格文章旨在說明如何透過受管結構描述，在 Cloud Directory 上快速建立原型並進行開發。其也包含 Java 程式碼範本。
如何更有效率地在 Amazon Cloud Directory 搜尋	此部落格文章說明如何透過以面向為基礎的索引，進行更有效率的搜尋。其也包含 Java 程式碼範本。
如何透過就地結構描述升級，輕鬆套用 Amazon Cloud Directory 結構描述變更	此部落格文章說明如何針對任何運作 (執行的) 雲端目錄，執行就地結構描述升級。其也包含 Java 程式碼範本。
使用 Batch 操作在 Amazon Cloud Directory 中寫入和讀取多個物件	說明如何使用批次讀取和寫入。其也包含 Java 程式碼範本。
如何使用 Amazon Cloud Directory 中的 Batch 參考來參考 Batch 請求中的新物件	說明如何使用批次參考。其也包含 Java 程式碼範本。
Cloud Directory Update — Support Directory Update	說明如何使用類型連結在 Cloud Directory 中跨階層建立和搜尋關係。其也包含 Java 程式碼範本。

Cloud Directory 部落格文章	描述
全新 Cloud Directory API 讓您更輕鬆地查詢多個維度的資料	說明如何使用 `ListObjectParentPaths` API 以單一呼叫查詢多個維度的資料。
如何使用 Amazon Cloud Directory 建立具有獨立階層的組織圖表	說明如何使用 Java 程式碼範本建立結構描述與目錄。
Amazon Cloud Directory-適用於分層式資料的 Cloud Directory	說明 AWS 推出了新服務 Cloud Directory。

Cloud Directory	連結
Cloud Directory 開發人員指南	https://docs.aws.amazon.com/clouddirectory/latest/developerguide/what_is_cloud_directory.html
Cloud Directory API 參考	https://docs.aws.amazon.com/clouddirectory/latest/APIReference/welcome.html
Cloud Directory 限制	https://docs.aws.amazon.com/clouddirectory/latest/developerguide/limits.html

Cloud Directory	連結
Cloud Directory 產品資訊	https://aws.amazon.com/cloud-directory/
Cloud Directory 定價	https://aws.amazon.com/cloud-directory/pricing/

文件歷史記錄

下表會說明自上次發行後，該文件的變更內容。Amazon Cloud Directory 開發人員指南。

- 最新文件更新時間： 2018 年 6 月 21 日

update-history-change	update-history-description	update-history-date
新的受管結構描述	新增受管結構描述選項的內容。	2018 年 6 月 21 日
將內容遷移至本指南	從 AWS Directory Service 管理員指南中，將全部的現有 Cloud Directory 內容轉移至這個新的 Amazon Cloud Directory 開發人員指南，以期更加符合客戶的需求。	2018 年 6 月 20 日
就地結構描述升級	使用就地結構描述升級，在您的 Amazon Cloud Directory 目錄之間新增套用結構描述變更的內容。	2017 年 12 月 6 日
以面向為基礎的索引	新增「以面向為基礎的索引」一節。	2017 年 8 月 9 日
批次	Amazon Cloud Directory 的更新批次的相關資訊。	2017 年 7 月 26 日
合規	新增 HIPAA 和 PCI 合規的相關資訊。	2017 年 7 月 14 日
類型連結	Amazon Cloud Directory 新增新類型連結內容。	2017 年 5 月 31 日
Amazon Cloud Directory Service	引進新的目錄類型。	2017 年 1 月 26 日

AWS 詞彙表

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。