開發人員指南

AWS Cloud Map

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

AWS Cloud Map: 開發人員指南

Copyright © 2025 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，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS Cloud Map？ 1
的元件 AWS Cloud Map 1
存取 AWS Cloud Map 2
AWS Identity and Access Management 3
AWS Cloud Map 定價 4
AWS Cloud Map 和 AWS 雲端合規 4
開始使用 5
設定 5
註冊 AWS 5
存取 API AWS CLIAWS Tools for Windows PowerShell、 或 AWS SDKs 7
設定 AWS Command Line Interface 或 AWS Tools for Windows PowerShell 8
下載 AWS SDK 9
AWS Cloud Map 搭配 DNS 查詢和 API 呼叫使用 9
先決條件 9
步驟 1：建立命名空間 10
步驟 2：建立 服務 10
步驟 3：建立服務執行個體 11
步驟 4：探索服務執行個體 12
步驟 5：清除 13
使用 搭配 DNS 查詢和 API 呼叫使用 AWS Cloud Map 服務探索 AWS CLI 13
..... 14
先決條件 14
建立 AWS Cloud Map 命名空間 14
建立 AWS Cloud Map 服務 15
註冊 AWS Cloud Map 服務執行個體 16
探索 AWS Cloud Map 服務執行個體 18
清除資源 19
AWS Cloud Map 搭配自訂屬性使用 20
先決條件 21
步驟 1：建立命名空間 21
步驟 2：建立 DynamoDB 資料表 21
步驟 3：建立資料服務 22
步驟 4：建立執行角色 22
步驟 5：建立 Lambda 函數以寫入資料 23

步驟 6：建立應用程式服務 24
步驟 7：建立 Lambda 函數以讀取資料 25
步驟 8：建立服務執行個體 26
步驟 9：建立和執行用戶端應用程式 26
步驟 10：清除 29
使用 搭配自訂屬性使用 AWS Cloud Map 服務探索 AWS CLI 30
..... 30
先決條件 30
建立 AWS Cloud Map 命名空間 30
建立 DynamoDB 資料表 31
建立 AWS Cloud Map 資料服務並註冊 DynamoDB 資料表 31
為 Lambda 函數建立 IAM 角色 32
建立 Lambda 函數以寫入資料 34
建立 AWS Cloud Map 應用程式服務並註冊 Lambda 寫入函數 35
建立 Lambda 函數以讀取資料 36
將 Lambda 讀取函數註冊為服務執行個體 38
建立和執行用戶端應用程式 38
清除資源 41
命名空間 43
建立命名空間 43
執行個體探索選項 43
程序 46
後續步驟 49
列出命名空間 49
刪除命名空間 51
服務 53
運作狀態檢查組態 53
Route 53 運作狀態檢查 54
自訂運作狀態檢查 54
DNS 組態 55
路由政策 55
記錄類型 56
建立服務 57
後續步驟 62
更新服務 62
列出命名空間中的服務 64

- 刪除服務 ... 66
- 服務執行個體 ... 68
  - 註冊服務執行個體 ... 68
  - 列出服務執行個體 ... 73
  - 更新服務執行個體 ... 74
    - 更新服務執行個體的自訂屬性 ... 75
  - 取消註冊服務執行個體 ... 75
- 安全 ... 78
  - 身分和存取權管理 ... 78
    - 目標對象 ... 79
    - 使用身分驗證 ... 79
    - 使用政策管理存取權 ... 82
    - AWS Cloud Map 如何使用 IAM ... 84
    - 身分型政策範例 ... 89
    - AWS 受管政策 ... 96
    - AWS Cloud Map API 許可參考 ... 98
    - 故障診斷 ... 101
  - 合規驗證 ... 103
  - 恢復能力 ... 104
  - 基礎設施安全性 ... 104
    - AWS PrivateLink ... 104
- 監控 ... 107
  - 使用 記錄 AWS Cloud Map API 呼叫 AWS CloudTrail ... 107
    - 資料事件 ... 108
    - 管理事件 ... 109
    - 事件範例 ... 110
- 標記您的 資源 ... 114
  - 如何標記資源 ... 114
  - 限制 ... 115
  - 更新 AWS Cloud Map 資源的標籤 ... 115
- Service Quotas ... 118
  - 管理您的服務配額 ... 119
  - 處理 DiscoverInstances API 請求限流 ... 120
    - 如何套用調節 ... 120
    - 調整 API 限流配額 ... 121
- 文件歷史紀錄 ... 122

..... cxxiv

什麼是 AWS Cloud Map？

AWS Cloud Map 是一種全受管解決方案，可用來將邏輯名稱映射至應用程式依賴的後端服務和資源。它還可協助您的應用程式使用其中一個 AWS SDKs、RESTful API 呼叫或 DNS 查詢來探索資源。 只會 AWS Cloud Map 提供運作狀態良好的資源，可以是 Amazon DynamoDB (DynamoDB) 資料表、Amazon Simple Queue Service (Amazon SQS) 佇列、使用 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或 Amazon Elastic Container Service (Amazon ECS) 任務建置的任何高階應用程式服務等。

的元件 AWS Cloud Map

命名空間

若要開始使用，請先建立 AWS Cloud Map 命名空間，做為將應用程式服務分組的方式。命名空間會識別您要用來尋找資源的名稱，並指定您要如何尋找資源：使用 AWS Cloud Map DiscoverInstances API 呼叫、VPC 中的 DNS 查詢或公有 DNS 查詢。在大多數情況下，命名空間包含應用程式的所有服務，例如帳單應用程式。如需詳細資訊，請參閱AWS Cloud Map 命名空間。

服務

建立命名空間後，您可以為要 AWS Cloud Map 用來尋找端點的每種資源類型建立 AWS Cloud Map 服務。例如，您可能會為 Web 伺服器和資料庫伺服器建立服務。

服務是當您的應用程式新增另一個資源時 AWS Cloud Map 所使用的範本，例如另一個 Web 伺服器。如果您在建立命名空間時使用 DNS 來尋找資源，服務會包含有關您想要用來尋找 web 伺服器之記錄類型的相關資訊。服務也會指出您是否要檢查資源的運作狀態，以及您是否要使用 Amazon Route 53 運作狀態檢查或第三方運作狀態檢查程式。如需詳細資訊，請參閱AWS Cloud Map 服務。

服務執行個體

當您的應用程式新增資源時，您可以在程式碼中呼叫 AWS Cloud Map RegisterInstance API 動作，該動作會在 AWS Cloud Map 服務中建立服務執行個體。服務執行個體包含應用程式如何尋找資源的資訊，無論是使用 DNS 還是使用 AWS Cloud Map DiscoverInstances API 動作。

當您的應用程式需要連線到資源時，它會透過指定與資源相關聯的命名空間和服務來呼叫 DiscoverInstances 或使用公有或私有 DNS 查詢。 AWS Cloud Map 會傳回如何尋找一或多個資源的相關資訊。如果您在建立服務時指定運作狀態檢查， 只會 AWS Cloud Map 傳回運作狀態良好的執行個體。如需詳細資訊，請參閱AWS Cloud Map 服務執行個體。

存取 AWS Cloud Map

您可以透過 AWS Cloud Map 下列方式存取 ：

- AWS Management Console – 本指南中的程序說明如何使用 AWS Management Console 來執行任務。
- AWS SDKs – 如果您使用的是 AWS 提供 開發套件的程式設計語言，您可以使用 開發套件來存取 AWS Cloud Map。開發套件可簡化身分驗證、與您的開發環境輕鬆整合，並可存取 AWS Cloud Map 命令。如需詳細資訊，請參閱 Amazon Web Services 適用工具。
- AWS Command Line Interface – 如需詳細資訊，請參閱AWS Command Line Interface 《 使用者指南》中的開始使用 AWS CLI 。
- AWS Tools for Windows PowerShell – 如需詳細資訊，請參閱AWS Tools for PowerShell 《 使用者指南》中的開始使用 AWS Tools for Windows PowerShell 。
- AWS Cloud Map API – 如果您使用的是無法使用 SDK 的程式設計語言，請參閱 AWS Cloud Map API 參考以取得 API 動作以及如何提出 API 請求的相關資訊。

> **Note**
>
> IPv6 用戶端支援 – 自 2023 年 6 月 22 日起， AWS Cloud Map 從IPv6用戶端傳送至 的任何命令都會路由至新的雙堆疊端點 (`servicediscovery.<region>.api.aws`)。在 2023 年 6 月 22 日之前發行的下列區域中， AWS Cloud Map IPv6只有舊版 (`servicediscovery.<region>.amazonaws.com`) 和雙堆疊端點都可以存取網路：
>
> - 美國東部 (俄亥俄) – us-east-2
> - 美國東部 (維吉尼亞北部) – us-east-1
> - 美國西部 (加利佛尼亞北部) – us-west-1
> - 美國西部 (奧勒岡) – us-west-2
> - 非洲 (開普敦) – af-south-1
> - 亞太區域 (香港) – ap-east-1
> - 亞太區域 （ 海德拉巴 ） – ap-south-2
> - 亞太區域 (雅加達) – ap-southeast-3
> - 亞太區域 （ 墨爾本 ） – ap-southeast-4
> - 亞太區域 (孟買) – ap-south-1
> - 亞太區域 (大阪) - (ap-northeast-3)
> - 亞太區域 (首爾) – ap-northeast-2

- 亞太區域 (新加坡) – ap-southeast-1
- 亞太區域 (雪梨) – ap-southeast-2
- 亞太區域 (東京) – ap-northeast-1
- 加拿大 (中部) – ca-central-1
- 歐洲 (法蘭克福) – eu-central-1
- 歐洲 (愛爾蘭) – eu-west-1
- 歐洲 (倫敦) – eu-west-2
- 歐洲 (米蘭) – eu-south-1
- 歐洲 (巴黎) – eu-west-3
- 歐洲 （ 西班牙 ） – eu-south-2
- 歐洲 (斯德哥爾摩) – eu-north-1
- 歐洲 （ 蘇黎世 ） – eu-central-2
- 中東 (巴林)– me-south-1
- 中東 （ 阿拉伯聯合大公國 ） – me-central-1
- 南美洲 (聖保羅) – sa-east-1
- AWS GovCloud （ 美國東部 ） – us-gov-east-1
- AWS GovCloud （ 美國西部 ） – us-gov-west-1

AWS Identity and Access Management

AWS Cloud Map 與 AWS Identity and Access Management (IAM) 整合，您的組織可用來執行下列動作的服務：

- 在組織的 AWS 帳戶下建立使用者和群組
- 以有效率的方式在 AWS 帳戶中的使用者之間共用您的帳戶資源
- 將唯一安全登入資料指派給每位使用者
- 細微控制每位使用者對服務與資源的存取

例如，您可以使用 IAM 搭配 AWS Cloud Map 來控制您 AWS 帳戶中哪些使用者可以建立新的命名空間或註冊執行個體。

如需 IAM 的一般資訊，請參閱下列資源：

- 的 Identity and Access Management AWS Cloud Map
- AWS Identity and Access Management
- IAM 使用者指南

AWS Cloud Map 定價

AWS Cloud Map 定價是根據您在服務登錄檔中註冊的資源，以及您為了探索而進行的 API 呼叫。由於 AWS Cloud Map 沒有預付款，您只需支付使用量的費用。

或者，您可以為具有 IP 地址的資源啟用 DNS 探索。您也可以使用 Amazon Route 53 運作狀態檢查來啟用資源的運作狀態檢查，無論您是使用 API 呼叫或 DNS 查詢來探索執行個體。您將需要支付與 Route 53 DNS 和運作狀態檢查用量相關的額外費用。

如需詳細資訊，請參閱AWS Cloud Map 定價。

AWS Cloud Map 和 AWS 雲端合規

如需有關 AWS Cloud Map 符合各種安全合規法規和稽核標準的資訊，請參閱下列頁面：

- AWS 雲端合規
- AWS 合規計劃範圍內的服務

入門 AWS Cloud Map

下列指南說明如何設定 ，以使用 AWS Cloud Map 命名空間來使用 AWS Cloud Map 和執行常見任務。

指南概觀	進一步了解
註冊 AWS 並準備使用 AWS Cloud Map	設定 以使用 AWS Cloud Map
使用 DNS 查詢和 API 呼叫來探索後端服務。	了解如何搭配 DNS 查詢和 API 呼叫使用 AWS Cloud Map 服務探索
使用 DNS 查詢和 API 呼叫來探索使用 的後端服務 AWS CLI。	了解如何使用 搭配 DNS 查詢和 API 呼叫使用 AWS Cloud Map 服務探索 AWS CLI
建立範例應用程式，並在程式碼中使用自訂屬性來探索資源。	了解如何搭配自訂屬性使用 AWS Cloud Map 服務探索
建立範例應用程式，並使用程式碼中的自訂屬性來探索使用 的資源 AWS CLI。	了解如何使用 搭配自訂屬性使用 AWS Cloud Map 服務探索 AWS CLI

設定 以使用 AWS Cloud Map

以下章節中的概觀和程序旨在協助您開始使用 ， AWS 並準備好開始使用 AWS Cloud Map。

主題

- 註冊 AWS
- 存取 API AWS CLIAWS Tools for Windows PowerShell、 或 AWS SDKs
- 設定 AWS Command Line Interface 或 AWS Tools for Windows PowerShell
- 下載 AWS SDK

註冊 AWS

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電或簡訊，並在電話鍵盤輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 會在註冊程序完成後傳送確認電子郵件給您。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理存取權的使用者

註冊 後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center和建立管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS Management Console身分登入 。在下一頁中，輸入您的密碼。

   如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。
2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者（主控台）啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的啟用 AWS IAM Identity Center。
2. 在 IAM Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center《 使用者指南》中的使用預設值設定使用者存取 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

  如需使用 IAM Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。
2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

存取 API AWS CLIAWS Tools for Windows PowerShell、 或 AWS SDKs

若要使用 API AWS CLI AWS Tools for Windows PowerShell、 或 AWS SDKs，您必須建立存取金鑰。存取金鑰包含存取金鑰 ID 與私密存取金鑰，用來簽署您對 AWS提出的程式設計請求。

如果使用者想要與 AWS 外部互動，則需要程式設計存取 AWS Management Console。授予程式設計存取權的方式取決於正在存取的使用者類型 AWS。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	根據
人力資源身分 (IAM Identity Center 中管理的使用者)	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。

哪個使用者需要程式設計存取權？	到	根據
		• AWS SDKs、工具和 AWS APIs，請參閱 AWS SDK 和工具參考指南中的 SDKsIAM Identity Center 身分驗證。
IAM	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	遵循《IAM 使用者指南》中將臨時登入資料與 AWS 資源搭配使用中的指示。
IAM	(不建議使用) 使用長期登入資料來簽署對 AWS CLI、 AWS SDKs 或 AWS APIs程式設計請求。	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的使用 IAM 使用者憑證進行身分驗證。 • AWS SDKs和工具，請參閱 AWS SDKs和工具參考指南中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱《IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

設定 AWS Command Line Interface 或 AWS Tools for Windows PowerShell

AWS Command Line Interface (AWS CLI) 是用於管理 AWS 服務的統一工具。如需有關如何安裝和設定 的資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的安裝或更新至最新版本的 AWS CLI。

如果您有使用 Windows PowerShell 的經驗，可能偏好使用 AWS Tools for Windows PowerShell。如需詳細資訊，請參閱 AWS Tools for PowerShell 使用者指南中的設定 AWS Tools for Windows PowerShell。

下載 AWS SDK

如果您使用的是 AWS 提供 開發套件的程式設計語言，我們建議您使用 開發套件，而非 AWS Cloud Map API。使用 SDK 有幾個優點。SDKs可讓身分驗證更簡單、輕鬆與您的開發環境整合，並提供對 AWS Cloud Map 命令的存取。如需詳細資訊，請參閱 Amazon Web Services 適用工具。

了解如何搭配 DNS 查詢和 API 呼叫使用 AWS Cloud Map 服務探索

下列教學會模擬具有兩個後端服務的微服務架構。第一個服務將使用 DNS 查詢來探索。第二個服務只能使用 AWS Cloud Map API 來探索。

> **Note**
>
> 資源詳細資訊，例如網域名稱和 IP 地址，僅供模擬之用。它們無法透過網際網路解析。

如需本教學課程的end-to-end AWS CLI 版本，請參閱 了解如何使用 搭配 DNS 查詢和 API 呼叫使用 AWS Cloud Map 服務探索 AWS CLI。

先決條件

必須符合下列先決條件，才能成功完成教學課程。

- 開始之前，請完成 設定 以使用 AWS Cloud Map 中的步驟。
- 如果您尚未安裝 AWS Command Line Interface，請依照安裝或更新最新版本的 AWS CLI中的步驟進行安裝。

  本教學課程需使用命令列終端機或 Shell 來執行命令。在 Linux 和 macOS 中，使用您偏好的 Shell 和套件管理工具。

  > **Note**
  >
  > 在 Windows 中，作業系統的內建終端不支援您常與 Lambda 搭配使用的某些 Bash CLI 命令 (例如 `zip`)。若要取得 Ubuntu 和 Bash 的 Windows 整合版本，請安裝適用於 Linux 的 Windows 子系統。

- 本教學課程需要具有 `dig` DNS 查詢公用程式命令的本機環境。

步驟 1：建立 AWS Cloud Map 命名空間

在此步驟中，您會建立公有 AWS Cloud Map namespace。 會使用相同的名稱代表您 AWS Cloud Map 建立 Route 53 託管區域。這可讓您探索在此命名空間中建立的服務執行個體，無論是使用公有 DNS 記錄或使用 AWS Cloud Map API 呼叫。

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 選擇 Create namespace (建立命名空間)。
3. 針對命名空間名稱，指定 `cloudmap-tutorial.com`。

   > **Note**
   > 如果您要在生產環境中使用此項目，建議您確保已指定您擁有或可存取的網域名稱。但是，基於此教學的目的，它不需要是正在使用的實際網域。

4. （選用）針對命名空間描述，指定您要使用命名空間的描述。
5. 針對執行個體探索，選取 API 呼叫和公有 DNS 查詢。
6. 保留其餘的預設值，然後選擇建立命名空間。

步驟 2：建立 AWS Cloud Map 服務

在此步驟中，您會建立兩個 服務。第一個服務將使用公有 DNS 和 API 呼叫來探索。第二個服務只能使用 API 呼叫來探索。

1. 登入 AWS Management Console 並開啟位於 https：//https://console.aws.amazon.com/cloudmap/ 的 AWS Cloud Map 主控台。
2. 在左側導覽窗格中，選擇命名空間以列出您已建立的命名空間。
3. 從命名空間清單中，選取`cloudmap-tutorial.com`命名空間，然後選擇檢視詳細資訊。
4. 在服務區段中，選擇建立服務，然後執行下列動作來建立第一個服務。
   a. 對於服務名稱，輸入 `public-service`。服務名稱將套用至 AWS Cloud Map 建立的 DNS 記錄。使用的格式為 `<service-name>.<namespace-name>`。
   b. 針對服務探索組態，選取 API 和 DNS。
   c. 在 DNS 組態區段中，針對路由政策，選取多值回答路由。

> **Note**
>
> 選取後，主控台會將其翻譯為 MULTIVALUE。如需可用路由選項的詳細資訊，請參閱《Route 53 開發人員指南》中的選擇路由政策。

d. 保留其餘的預設值，然後選擇建立服務，這會讓您返回命名空間詳細資訊頁面。

5. 在服務區段中，選擇建立服務，然後執行下列動作來建立第二個服務。

   a. 對於服務名稱，輸入 `backend-service`。

   b. 針對服務探索組態，僅選取 API。

   c. 保留其餘的預設值，然後選擇建立服務。

步驟 3：註冊 AWS Cloud Map 服務執行個體

在此步驟中，您會建立兩個服務執行個體，一個用於我們命名空間中的每個服務。

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 從命名空間清單中，選取您在步驟 1 中建立的命名空間，然後選擇檢視詳細資訊。
3. 在命名空間詳細資訊頁面上，從服務清單中，選取`public-service`服務，然後選擇檢視詳細資訊。
4. 在服務執行個體區段中，選擇註冊服務執行個體，然後執行下列動作來建立第一個服務執行個體。

   a. 針對服務執行個體 ID，指定 `first`。

   b. 針對 IPv4 地址，指定 `192.168.2.1`。

   c. 保留其餘的預設值，然後選擇註冊服務執行個體。

5. 使用頁面頂端的導覽列，選取 cloudmap-tutorial.com 以導覽回命名空間詳細資訊頁面。
6. 在命名空間詳細資訊頁面上，從服務清單中，選取後端服務，然後選擇檢視詳細資訊。
7. 在服務執行個體區段中，選擇註冊服務執行個體，然後執行下列動作來建立第二個服務執行個體。

   a. 針對服務執行個體 ID，指定 `second` 以指出這是第二個服務執行個體。

   b. 針對執行個體類型，選取識別另一個資源的資訊。

   c. 對於自訂屬性，使用 `service-name`作為索引鍵和 `backend`作為值來新增索引鍵/值對。

   d. 選擇 Register service instance (註冊服務執行個體)。

步驟 4：探索 AWS Cloud Map 服務執行個體

現在已建立 AWS Cloud Map 命名空間、服務和服務執行個體，您可以透過探索執行個體來驗證一切是否正常運作。使用 `dig`命令來驗證公有 DNS 設定和 AWS Cloud Map API 來驗證後端服務。如需 `dig`命令的詳細資訊，請參閱 dig - DNS 查詢公用程式。

1. 登入 AWS Management Console 並開啟 Route 53 主控台，網址為 https：//https://console.aws.amazon.com/route53/。
2. 在左側導覽窗格中，選擇 Hosted zones (託管區域)。
3. 選取 cloudmap-tutorial.com 託管區域。這會在單獨的窗格中顯示託管區域詳細資訊。請記下與您的託管區域相關聯的名稱伺服器，因為我們會在下一個步驟中使用這些伺服器。
4. 使用 dig 命令和託管區域的 Route 53 名稱伺服器之一，查詢服務執行個體的 DNS 記錄。

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

輸出`ANSWER SECTION`中的 應該會顯示您與服務相關聯的 IPv4 地址`public-service`。

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

5. 使用 AWS CLI查詢第二個服務執行個體的屬性。

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

輸出會將您與服務相關聯的屬性顯示為鍵/值對。

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

```
    "InstancesRevision": 71462688285136850
}
```

步驟 5：清除資源

完成教學課程後，您可以刪除 資源。 AWS Cloud Map 要求您以相反順序清理它們，先是服務執行個體，再是 服務，最後是 命名空間。當您完成這些步驟時， AWS Cloud Map 會代表您清理 Route 53 資源。

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 從命名空間清單中，選取`cloudmap-tutorial.com`命名空間，然後選擇檢視詳細資訊。
3. 在命名空間詳細資訊頁面上，從服務清單中，選取`public-service`服務，然後選擇檢視詳細資訊。
4. 在服務執行個體區段中，選取`first`執行個體，然後選擇取消註冊。
5. 使用頁面頂端的導覽列，選取 cloudmap-tutorial.com 以導覽回命名空間詳細資訊頁面。
6. 在命名空間詳細資訊頁面上，從服務清單中，選取公有服務，然後選擇刪除。
7. 針對 重複步驟 3-6`backend-service`。
8. 在左側導覽中，選擇命名空間。
9. 選取`cloudmap-tutorial.com`命名空間，然後選擇刪除。

> **Note**
>
> 雖然 會代表您 AWS Cloud Map 清除 Route 53 資源，但您可以導覽至 Route 53 主控台，以確認`cloudmap-tutorial.com`託管區域已刪除。

了解如何使用 搭配 DNS 查詢和 API 呼叫使用 AWS Cloud Map 服務探索 AWS CLI

本教學課程示範如何使用 AWS Command Line Interface (CLI) 來使用 AWS Cloud Map 服務探索。您將建立具有兩個後端服務的微服務架構 – 一個可使用 DNS 查詢探索，另一個可使用 AWS Cloud Map API 探索。

如需包含 AWS Cloud Map 主控台步驟的教學課程，請參閱 了解如何搭配 DNS 查詢和 API 呼叫使用 AWS Cloud Map 服務探索。

先決條件

必須符合下列先決條件，才能成功完成教學課程。

- 開始之前，請完成 設定 以使用 AWS Cloud Map 中的步驟。
- 如果您尚未安裝 AWS Command Line Interface，請依照安裝或更新最新版本的 AWS CLI中的步驟進行安裝。

  本教學課程需使用命令列終端機或 Shell 來執行命令。在 Linux 和 macOS 中，使用您偏好的 Shell 和套件管理工具。

  > **Note**
  >
  > 在 Windows 中，作業系統的內建終端不支援您常與 Lambda 搭配使用的某些 Bash CLI 命令 (例如 `zip`)。若要取得 Ubuntu 和 Bash 的 Windows 整合版本，請安裝適用於 Linux 的 Windows 子系統。

- 本教學課程需要具有 `dig` DNS 查詢公用程式命令的本機環境。

建立 AWS Cloud Map 命名空間

首先，您將建立公有 AWS Cloud Map 命名空間。 AWS Cloud Map 將建立同名的 Route 53 託管區域，透過 DNS 記錄和 API 呼叫啟用服務探索。

1. 建立公有 DNS 命名空間：

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

   命令會傳回操作 ID，您可以用來檢查命名空間建立的狀態：

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

2. 檢查操作狀態以確認命名空間已成功建立：

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

3. 操作成功後，取得命名空間 ID：

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

此命令會傳回命名空間 ID，您需要此 ID 才能進行後續步驟：

```
ns-abcd1234xmplefgh
```

建立 AWS Cloud Map 服務

現在，在您的命名空間中建立兩項服務。第一個服務將可使用 DNS 和 API 呼叫來探索，而第二個服務將只能使用 API 呼叫來探索。

1. 建立第一個已啟用 DNS 探索的服務：

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

命令會傳回所建立服務的詳細資訊：

```
{
    "Service": {
        "Id": "srv-abcd1234xmplefgh",
        "Arn": "arn:aws:servicediscovery:us-east-2:123456789012:service/srv-
abcd1234xmplefgh",
        "Name": "public-service",
        "NamespaceId": "ns-abcd1234xmplefgh",
        "DnsConfig": {
```

```
            "NamespaceId": "ns-abcd1234xmplefgh",
            "RoutingPolicy": "MULTIVALUE",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 300
                }
            ]
        },
        "CreateDate": 1673613600.000,
        "CreatorRequestId": "public-service-request"
    }
}
```

2. 使用僅限 API 探索建立第二個服務：

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

命令會傳回所建立服務的詳細資訊：

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

註冊 AWS Cloud Map 服務執行個體

接著，為您的每個服務註冊服務執行個體。這些執行個體代表將探索的實際資源。

1. 使用 DNS 探索的 IPv4 地址註冊第一個執行個體：

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

命令會傳回 操作 ID：

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

2. 檢查操作狀態以確認執行個體已成功註冊：

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

3. 使用 API 探索的自訂屬性註冊第二個執行個體：

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

命令會傳回 操作 ID：

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

4. 檢查操作狀態以確認執行個體已成功註冊：

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

探索 AWS Cloud Map 服務執行個體

現在您已建立並註冊您的服務執行個體，您可以使用 DNS 查詢和 AWS Cloud Map API 來探索它們，以驗證一切是否正常運作。

1. 首先，取得 Route 53 託管區域 ID：

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

   這會傳回託管區域 ID：

```
/hostedzone/Z1234ABCDXMPLEFGH
```

2. 取得託管區域的名稱伺服器：

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

   這會傳回其中一個名稱伺服器：

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

3. 使用 `dig`命令來查詢公有服務的 DNS 記錄：

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

   輸出應該會顯示您與服務相關聯的 IPv4 地址：

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

4. 使用 AWS CLI 探索後端服務執行個體：

```
aws servicediscovery discover-instances \
    --namespace-name cloudmap-tutorial.com \
```

```
    --service-name backend-service \
    --region us-east-2
```

輸出會顯示您與服務相關聯的屬性：

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

清除資源

完成教學課程後，請清除資源以避免產生費用。 AWS Cloud Map 要求您以相反的順序進行清除：先是服務執行個體，再是 服務，最後是 命名空間。

1. 取消註冊第一個服務執行個體：

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

2. 取消註冊第二個服務執行個體：

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

3. 刪除公有服務：

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

4. 刪除後端服務：

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

5. 刪除命名空間：

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

6. 確認 Route 53 託管區域已刪除：

```
aws route53 list-hosted-zones-by-name \
    --dns-name cloudmap-tutorial.com
```

了解如何搭配自訂屬性使用 AWS Cloud Map 服務探索

下列教學課程示範如何使用 AWS Cloud Map 服務探索搭配可使用 AWS Cloud Map API 探索的自訂屬性。本教學課程會逐步引導您使用 建立和執行用戶端應用程式 AWS CloudShell。應用程式使用兩個 Lambda 函數將資料寫入 DynamoDB 資料表，然後從資料表讀取。Lambda 函數和 DynamoDB 資料表會在 中註冊 AWS Cloud Map 為服務執行個體。用戶端應用程式和 Lambda 函數中的程式碼使用 AWS Cloud Map 自訂屬性來探索執行任務所需的資源。

如需本教學課程的 AWS CLI版本，請參閱 了解如何使用 搭配自訂屬性使用 AWS Cloud Map 服務探索 AWS CLI。

> **Important**
>
> 您將在研討會期間建立 AWS 資源，這會在您的帳戶 AWS 中產生費用。建議您在完成研討會後立即清理資源，以將成本降至最低。

先決條件

開始之前，請完成 設定 以使用 AWS Cloud Map 中的步驟。

步驟 1：建立 AWS Cloud Map 命名空間

在此步驟中，您會建立 AWS Cloud Map 命名空間。命名空間是用來將應用程式服務分組的建構。建立命名空間時，您可以指定如何探索資源。在此步驟中建立的命名空間中建立的資源，將可透過使用自訂屬性的 AWS Cloud Map API 呼叫來探索。

1. 登入 AWS Management Console 並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 選擇 Create namespace (建立命名空間)。
3. 針對命名空間名稱，指定 `cloudmap-tutorial`。
4. （選用）針對命名空間描述，指定您要使用命名空間的描述。
5. 針對執行個體探索，選取 API 呼叫。
6. 保留其餘的預設值，然後選擇建立命名空間。

步驟 2：建立 DynamoDB 資料表

在此步驟中，您會建立 DynamoDB 資料表。資料表用於存放和擷取您將在下列步驟中建立之範例應用程式的資料。

如需有關如何建立 DynamoDB 的資訊，請參閱 DynamoDB 開發人員指南中的步驟 1：在 DynamoDB 中建立資料表，並使用下表來決定要指定哪些選項。 DynamoDB

選項	Value
資料表名稱	cloudmap
分割區索引鍵	id

保留其餘設定的預設值並建立資料表。

步驟 3：建立 AWS Cloud Map 資料服務並將 DynamoDB 資料表註冊為執行個體

在此步驟中，您會建立 AWS Cloud Map 服務，然後將最後一個步驟中建立的 DynamoDB 資料表註冊為服務執行個體。

1. 在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台
2. 從命名空間清單中，選取`cloudmap-tutorial`命名空間，然後選擇檢視詳細資訊。
3. 在服務區段中，選擇建立服務並執行下列動作。
   a. 對於服務名稱，輸入 `data-service`。
   b. 保留其餘的預設值，然後選擇建立服務。
4. 在服務區段中，選取`data-service`服務，然後選擇檢視詳細資訊。
5. 在服務執行個體區段中，選擇註冊服務執行個體。
6. 在註冊服務執行個體頁面上，執行下列動作。
   a. 針對執行個體類型，選取識別另一個資源的資訊。
   b. 針對服務執行個體 ID，指定 `data-instance`。
   c. 在自訂屬性區段中，指定下列鍵/值對：鍵 = `tablename`，值 = `cloudmap`。

步驟 4：建立 AWS Lambda 執行角色

在此步驟中，您會建立 AWS Lambda 函數在下一個步驟中使用的 IAM 角色。您可以命名 IAM 角色`cloudmap-tutorial-role`並省略許可界限，因為該角色僅用於本教學課程，之後可以將其刪除。

建立 Lambda 的服務角色 (IAM 主控台)

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇角色，然後選擇建立角色。
3. 對於 Trusted entity type (信任的實體類型)，請選擇 AWS 服務。
4. 針對服務或使用案例，選擇 Lambda，然後選擇 Lambda 使用案例。
5. 選擇下一步。
6. 搜尋並選取`PowerUserAccess`政策旁的方塊，然後選擇下一步。

7. 選擇下一步。
8. 針對角色名稱，指定 `cloudmap-tutorial-role`。
9. 檢閱角色，然後選擇 Create role (建立角色)。

步驟 5：建立 Lambda 函數以寫入資料

在此步驟中，您會建立從頭開始編寫的 Lambda 函數，使用 AWS Cloud Map API 查詢您建立 AWS Cloud Map 的服務，將資料寫入 DynamoDB 資料表。

如需有關建立 Lambda 函數的資訊，請參閱《 AWS Lambda 開發人員指南》中的使用主控台建立 Lambda 函數，並使用下表來決定要指定或選擇哪些選項。

選項	Value
函數名稱	寫入函數
執行期	Python 3.12
架構	x86_64
許可	使用現有角色
現有角色	cloudmap-tutorial-role

建立函數之後，請更新範例程式碼以反映下列 Python 程式碼，然後部署函數。請注意，您要指定與為 DynamoDB 資料表建立之 AWS Cloud Map 服務執行個體相關聯的`datatable`自訂屬性。函數會產生介於 1 到 100 之間的隨機數字索引鍵，並將其與呼叫函數時傳遞給函數的值建立關聯。

```
import json
import boto3
import random

def lambda_handler(event, context):

    serviceclient = boto3.client('servicediscovery')

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

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

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

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

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

部署函數之後，若要避免逾時錯誤，請將函數逾時更新為 5 秒。如需詳細資訊，請參閱《 AWS Lambda 開發人員指南》中的設定 Lambda 函數逾時。

步驟 6：建立 AWS Cloud Map 應用程式服務，並將 Lambda 寫入函數註冊為執行個體

在此步驟中，您會建立 AWS Cloud Map 服務，然後將 Lambda 寫入函數註冊為服務執行個體。

1. 在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台
2. 在左側導覽中，選擇命名空間。
3. 從命名空間清單中，選取`cloudmap-tutorial`命名空間，然後選擇檢視詳細資訊。
4. 在服務區段中，選擇建立服務並執行下列動作。
   a. 對於服務名稱，輸入 `app-service`。
   b. 保留其餘的預設值，然後選擇建立服務。
5. 在服務區段中，選取`app-service`服務，然後選擇檢視詳細資訊。
6. 在服務執行個體區段中，選擇註冊服務執行個體。
7. 在註冊服務執行個體頁面上，執行下列動作。
   a. 針對執行個體類型，選取識別另一個資源的資訊。
   b. 針對服務執行個體 ID，指定 `write-instance`。
   c. 在自訂屬性區段中，指定下列鍵/值對。
      - key = `action`， value = `write`

- key = `functionname`， value = `writefunction`

步驟 7：建立 Lambda 函數以讀取資料

在此步驟中，您會建立從頭開始撰寫的 Lambda 函數，將資料寫入您建立的 DynamoDB 資料表。

如需有關建立 Lambda 函數的資訊，請參閱《 AWS Lambda 開發人員指南》中的使用主控台建立 Lambda 函數，並使用下表來決定要指定或選擇哪些選項。

選項	Value
函數名稱	讀取函數
執行期	Python 3.12
架構	x86_64
許可	使用現有角色
現有角色	cloudmap-tutorial-role

建立函數之後，請更新範例程式碼以反映下列 Python 程式碼，然後部署函數。函數掃描資料表 amd 會傳回所有項目。

```
import json
import boto3

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

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

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

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

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

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

部署函數之後，若要避免逾時錯誤，請將函數逾時更新為 5 秒。如需詳細資訊，請參閱《 AWS Lambda 開發人員指南》中的設定 Lambda 函數逾時。

步驟 8：將 Lambda 讀取函數註冊為 AWS Cloud Map 服務執行個體

在此步驟中，您會在先前建立的 服務中，將 Lambda 讀取函數註冊為`app-service`服務執行個體。

1. 在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台
2. 在左側導覽中，選擇命名空間。
3. 從命名空間清單中，選取`cloudmap-tutorial`命名空間，然後選擇檢視詳細資訊。
4. 在服務區段中，選取`app-service`服務，然後選擇檢視詳細資訊。
5. 在服務執行個體區段中，選擇註冊服務執行個體。
6. 在註冊服務執行個體頁面上，執行下列動作。
   a. 針對執行個體類型，選取識別另一個資源的資訊。
   b. 針對服務執行個體 ID，指定 `read-instance`。
   c. 在自訂屬性區段中，指定下列鍵/值對。
      - key = `action`， value = `read`
      - key = `functionname`， value = `readfunction`

步驟 9：在 上建立並執行讀取和寫入用戶端 AWS CloudShell

您可以在 中建立和執行使用程式碼 AWS CloudShell 的用戶端應用程式，以探索您在 中設定的服務，AWS Cloud Map 並呼叫這些服務。

1. 在 https：//https://console.aws.amazon.com/cloudshell/ 開啟 AWS CloudShell 主控台
2. 使用下列命令來建立名為 的檔案`writefunction.py`。

```
vim writeclient.py
```

3. 在 `writeclient.py` 檔案中，按下 `i` 按鈕進入插入模式。然後，複製並貼上下列程式碼。此程式碼探索 Lambda 函數，透過搜尋`app-service`服務`name=writeservice`中的自訂屬性來寫入資料。會傳回負責將資料寫入 DynamoDB 資料表的 Lambda 函數名稱。然後調用 Lambda 函數，將寫入資料表的範例承載做為值傳遞。

```
import boto3

serviceclient = boto3.client('servicediscovery')

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

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

lambdaclient = boto3.client('lambda')

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

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

4. 按逸出鍵，輸入 `:wq`，然後按 Enter 鍵儲存檔案並退出。
5. 使用下列命令來執行 Python 程式碼。

```
python3 writeclient.py
```

輸出應為`200`回應，如下所示。

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

6. 若要在上一個步驟中驗證寫入是否成功，請建立讀取用戶端。

   a. 使用下列命令來建立名為 的檔案`readfunction.py`。

   ```
   vim readclient.py
   ```

b. 在 `readclient.py` 檔案中，按下 `i` 按鈕以進入插入模式。然後，複製並貼上下列程式碼。此程式碼會掃描資料表，並傳回您在上一個步驟中寫入資料表的值。

```
import boto3

serviceclient = boto3.client('servicediscovery')

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

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

lambdaclient = boto3.client('lambda')

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

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

c. 按逸出鍵，輸入 `:wq`，然後按 Enter 鍵儲存檔案並退出。

d. 使用下列命令來執行 Python 程式碼。

```
python3 readclient.py
```

輸出看起來應該類似以下內容，列出透過執行 寫入資料表的值，`writefunction.py`以及在 Lambda 寫入函數中產生的隨機索引鍵。

```
  b'{"statusCode": 200, "body": "{\\"Items\\": [{\\"id\\": \\"45\
\", \\"todo\\": \\"This is a test data\\"}], \\"Count\\": 1, \
\"ScannedCount\\": 1, \\"ResponseMetadata\\": {\\"RequestId\\": \
\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"HTTPStatusCode\
\": 200, \\"HTTPHeaders\\": {\\"server\\": \\"Server\\", \\"date\\": \\"Thu, 25
 Jul 2024 20:43:33 GMT\\", \\"content-type\\": \\"application/x-amz-json-1.0\
\", \\"content-length\\": \\"91\\", \\"connection\\": \\"keep-alive\\", \\"x-
amzn-requestid\\": \\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\",
 \\"x-amz-crc32\\": \\"1163081893\\"}, \\"RetryAttempts\\": 0}}"}'
```

步驟 10：清除資源

完成教學課程後，請刪除資源以避免產生額外費用。 AWS Cloud Map 要求您以相反的順序、服務執行個體先、服務，最後是命名空間進行清除。下列步驟會逐步引導您清除教學課程中使用 AWS Cloud Map 的資源。

刪除 AWS Cloud Map 資源

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 從命名空間清單中，選取`cloudmap-tutorial`命名空間，然後選擇檢視詳細資訊。
3. 在命名空間詳細資訊頁面上，從服務清單中，選取`data-service`服務，然後選擇檢視詳細資訊。
4. 在服務執行個體區段中，選取`data-instance`執行個體，然後選擇取消註冊。
5. 使用頁面頂端的導覽列，選取 cloudmap-tutorial.com 以導覽回命名空間詳細資訊頁面。
6. 在命名空間詳細資訊頁面上，從服務清單中，選取資料服務，然後選擇刪除。
7. 針對 `app-service`服務和 `write-instance``read-instance`和服務執行個體重複步驟 3-6。
8. 在左側導覽中，選擇命名空間。
9. 選取`cloudmap-tutorial`命名空間，然後選擇刪除。

下表列出您可以遵循的程序，以刪除教學中使用的其他資源。

資源	步驟
DynamoDB 表	步驟 6：（選用） 刪除 DynamoDB 資料表以清除 Amazon DynamoDB 開發人員指南中的資源 DynamoDB
Lambda 函數和相關聯的 IAM 執行角色	《 AWS Lambda 開發人員指南》中的清除

了解如何使用 搭配自訂屬性使用 AWS Cloud Map 服務探索 AWS CLI

本教學課程示範如何搭配自訂屬性使用 AWS Cloud Map 服務探索。您將建立使用 的微服務應用程式， AWS Cloud Map 以使用自訂屬性動態探索資源。應用程式由兩個 Lambda 函數組成，這些函數會將資料寫入 DynamoDB 資料表並從中讀取，並在其中註冊所有資源 AWS Cloud Map。

如需教學課程的 AWS Management Console 版本，請參閱 了解如何搭配自訂屬性使用 AWS Cloud Map 服務探索。

先決條件

開始本教學課程之前，請先完成 中的步驟設定 以使用 AWS Cloud Map。

建立 AWS Cloud Map 命名空間

命名空間是用來將應用程式服務分組的建構。在此步驟中，您將建立一個命名空間，允許透過 AWS Cloud Map API 呼叫探索資源。

1. 執行下列命令來建立 HTTP 命名空間：

```
aws servicediscovery create-http-namespace \
  --name cloudmap-tutorial \
  --creator-request-id cloudmap-tutorial-request
```

   命令會傳回 操作 ID。您可以使用下列命令檢查操作的狀態：

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

2. 建立命名空間後，您可以擷取其 ID 以用於後續命令：

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

3. 將命名空間 ID 存放在變數中以供日後使用：

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

```
  --output text)
```

建立 DynamoDB 資料表

接著，建立 DynamoDB 資料表來存放應用程式的資料：

1. 執行下列命令來建立資料表：

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

2. 等待資料表變成作用中，再繼續：

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

   此命令會等到資料表完全建立且可供使用。

建立 AWS Cloud Map 資料服務並註冊 DynamoDB 資料表

現在，在您的命名空間中建立服務來代表資料儲存資源：

1. 執行下列命令來建立資料儲存資源 AWS Cloud Map 的服務：

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

2. 取得資料服務的服務 ID：

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

3. 使用指定資料表名稱的自訂屬性，將 DynamoDB 資料表註冊為服務執行個體：

```
aws servicediscovery register-instance \
```

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

自訂屬性`tablename=cloudmap`可讓其他 服務動態探索 DynamoDB 資料表名稱。

為 Lambda 函數建立 IAM 角色

建立 Lambda 函數用來存取 AWS 資源的 IAM 角色：

1. 建立 IAM 角色的信任政策文件：

```
cat > lambda-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

2. 執行下列命令，使用信任政策建立 IAM 角色：

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

3. 為具有最低權限許可的自訂 IAM 政策建立 檔案：

```
cat > cloudmap-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "servicediscovery:DiscoverInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/cloudmap"
    }
  ]
}
EOF
```

4. 建立政策並將其連接至 IAM 角色：

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

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

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

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

建立 Lambda 函數以寫入資料

若要建立將資料寫入 DynamoDB 資料表的 Lambda 函數，請遵循下列步驟：

1. 建立寫入函數的 Python 檔案：

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

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

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

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

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

        dynamodbclient = boto3.resource('dynamodb')

        table = dynamodbclient.Table(tablename)

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

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

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

此函數使用 從自訂屬性 AWS Cloud Map 探索 DynamoDB 資料表名稱，然後將資料寫入資料表。

2. 封裝和部署 Lambda 函數：

```
zip writefunction.zip writefunction.py

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

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

3. 更新函數逾時以避免逾時錯誤：

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

建立 AWS Cloud Map 應用程式服務並註冊 Lambda 寫入函數

若要在命名空間中建立其他服務來代表應用程式函數，請依照下列步驟執行：

1. 為應用程式函數建立服務：

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

2. 取得應用程式服務的服務 ID：

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

3. 使用自訂屬性將 Lambda 寫入函數註冊為服務執行個體：

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

自訂屬性`action=write`，並`functionname=writefunction`允許用戶端根據其用途探索此函數。

建立 Lambda 函數以讀取資料

若要建立從 DynamoDB 資料表讀取資料的 Lambda 函數，請遵循下列步驟：

1. 為讀取函數建立 Python 檔案：

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

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

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

        if not response.get("Instances"):
```

```
            return {
                'statusCode': 500,
                'body': json.dumps({"error": "No instances found"})
            }

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

        dynamodbclient = boto3.resource('dynamodb')

        table = dynamodbclient.Table(tablename)

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

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

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

此函數也會使用 AWS Cloud Map 來探索 DynamoDB 資料表名稱，然後從資料表讀取資料。它包含錯誤處理和分頁註解。

2. 封裝和部署 Lambda 函數：

```
zip readfunction.zip readfunction.py

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

3. 更新函數逾時：

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

將 Lambda 讀取函數註冊為服務執行個體

若要將 Lambda 讀取函數註冊為應用程式服務中的另一個服務執行個體，請遵循下列步驟：

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

自訂屬性`action=read`，並`functionname=readfunction`允許用戶端根據其用途探索此函數。

建立和執行用戶端應用程式

若要建立 Python 用戶端應用程式，以使用 AWS Cloud Map 來探索和叫用寫入函數，請遵循下列步驟：

1. 為寫入用戶端應用程式建立 Python 檔案：

```
cat > writeclient.py << EOF
```

```
import boto3
import json

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

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

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

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

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

    lambdaclient = boto3.client('lambda')

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

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

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

此用戶端使用 `QueryParameters`選項來尋找具有 `action=write` 屬性的服務執行個體。

2. 為讀取用戶端應用程式建立 Python 檔案：

```
cat > readclient.py << EOF
```

```
import boto3
import json

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

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

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

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

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

    lambdaclient = boto3.client('lambda')

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

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

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

3. 執行寫入用戶端，將資料新增至 DynamoDB 資料表：

```
python3 writeclient.py
```

輸出應會顯示 HTTP 狀態碼為 200 的成功回應。

4. 執行讀取用戶端以從 DynamoDB 資料表擷取資料：

```
python3 readclient.py
```

輸出應會顯示寫入資料表的資料，包括隨機產生的 ID 和「這是測試資料」值。

清除資源

完成教學課程後，請清除資源以避免產生額外費用。

1. 首先，執行下列命令來取消註冊服務執行個體：

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

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

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

2. 執行下列命令來刪除服務：

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

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

3. 執行下列命令來刪除命名空間：

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

4. 執行下列命令來刪除 Lambda 函數：

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

5. 執行下列命令來刪除 IAM 角色和政策：

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

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

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

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

6. 執行下列命令來刪除 DynamoDB 資料表：

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

7. 執行下列命令來清除暫存檔案：

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

AWS Cloud Map 命名空間

命名空間是 中的邏輯實體 AWS Cloud Map ，用於將應用程式的服務分組為通用名稱和可探索性層級。當您建立命名空間時，請指定下列項目：

- 您希望應用程式用來探索執行個體的名稱。
- AWS Cloud Map 可以探索您向 註冊之服務執行個體的方法。您可以決定是否需要透過網際網路、在特定虛擬私有雲端 (VPC) 中私下公開探索您的資源，或僅透過 API 呼叫來探索。

以下是有關命名空間的一般概念。

- 命名空間專屬於在其中建立 AWS 區域 的 。若要 AWS Cloud Map 在多個區域中使用 ，您需要在每個區域中建立命名空間。
- 如果您建立命名空間以允許 VPC 中的 DNS 查詢探索執行個體， AWS Cloud Map 會自動建立私有 Route 53 託管區域。此託管區域可以與多個 VPCs建立關聯。如需詳細資訊，請參閱《Amazon Route 53 API 參考》中的 AssociateVPCWithHostedZone。

主題

- 建立 AWS Cloud Map 命名空間以將應用程式服務分組
- 列出 AWS Cloud Map 命名空間
- 刪除 AWS Cloud Map 命名空間

建立 AWS Cloud Map 命名空間以將應用程式服務分組

您可以建立命名空間，以易記的名稱為您的應用程式分組服務，允許透過 API 呼叫或 DNS 查詢探索應用程式資源。

執行個體探索選項

下表摘要說明 中的不同執行個體探索選項， AWS Cloud Map 以及您可以建立的對應命名空間類型，視您應用程式的服務和設定而定。

命名空間類型	執行個體探索方法	運作方式	其他資訊
HTTP	API 呼叫	您應用程式中的資源只能呼叫 `DiscoverInstances` API 來探索其他資源。	• DiscoverInstances • CreateHttpNamespace
私有 DNS	VPC 中的 API 呼叫和 DNS 查詢	您應用程式中的資源可以透過呼叫 `DiscoverInstances` API，以及查詢 AWS Cloud Map 自動建立之私有 Route 53 託管區域中的 名稱伺服器，來探索其他資源。 建立的託管區域 AWS Cloud Map 與命名空間的名稱相同，並包含名稱格式為 *service-name*.*namespace-name* 的 DNS 記錄。 **Note** Route 53 Resolver 會使用私有託管區域中的記錄來解析源自 VPC 的 DNS 查詢。如果私有託管區域不包含符合 DNS 查詢	• DiscoverInstances • CreatePrivateDnsNamespace

命名空間類型	執行個體探索方法	運作方式	其他資訊
		中網域名稱的記錄，Route 53 會使用 `NXDOMAIN`（不存在的網域）回應查詢。	
公有 DNS	API 呼叫和公有 DNS 查詢	您應用程式中的資源可以透過呼叫 `DiscoverInstances` API 和查詢 AWS Cloud Map 自動建立的公有 Route 53 託管區域中的 名稱伺服器，來探索其他資源。 公有託管區域與命名空間的名稱相同，且包含名稱格式為 *`service-name.namespace-name`* 的 DNS 記錄。 **Note** 在這種情況下，命名空間名稱必須是您已註冊的網域名稱。	• DiscoverInstances • CreatePublicDnsNamespace

程序

您可以依照下列步驟，使用 AWS CLI AWS Management Console或適用於 Python 的 SDK 建立命名空間。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 選擇 Create namespace (建立命名空間)。
3. 針對命名空間名稱，輸入將用於探索執行個體的名稱。

   > **Note**
   >
   > - 針對公有 DNS 查詢設定的命名空間必須以最上層網域結尾。例如：`.com`。
   > - 您可以先將國際化網域名稱 (IDN) 轉換為 Punycode，來指定其名稱。如需線上轉換器的詳細資訊，請在網際網路上搜尋「punycode 轉換器」。
   >
   > 您也可以在以程式設計的方式建立命名空間時，將國際化網域名稱轉換為 Punycode。例如，如果您使用 Java，可以透過使用 java.net.IDN 程式庫的 `toASCII` 方法，將 Unicode 值轉換為 Punycode。

4. （選用）對於命名空間描述，輸入將在命名空間頁面和命名空間資訊下顯示之命名空間的相關資訊。您可以使用此資訊輕鬆識別命名空間。
5. 對於執行個體探索，您可以在 VPC 中的 API 呼叫、API 呼叫和 DNS 查詢，以及 API 呼叫和公有 DNS 查詢之間進行選擇，分別建立 HTTP、私有 DNS 或公有 DNS 命名空間。 VPCs 如需詳細資訊，請參閱執行個體探索選項。

   根據您的選擇，遵循下列步驟。

   - 如果您在 VPCs 中選擇 API 呼叫和 DNS 查詢，對於 VPC，請選擇您要與命名空間建立關聯的虛擬私有雲端 (VPC)。
   - 如果您在 VPCs或 API 呼叫和公有 DNS 查詢中選擇 API 呼叫和 DNS 查詢，請在 TTL 中指定以秒為單位的數值。存留時間 (TTL) 值會決定 DNS 解析程式針對使用命名空間建立之 Route 53 託管區域的授權開始 (SOA) DNS 記錄快取資訊的時間長度。如需 TTL 的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 TTL（秒）。

6. （選用）在標籤下，選擇新增標籤，然後指定要標記命名空間的索引鍵和值。您可以指定要新增至命名空間的一或多個標籤。標籤可讓您將 AWS 資源分類，以便更輕鬆地管理資源。如需詳細資訊，請參閱標記您的 AWS Cloud Map 資源。
7. 選擇 Create namespace (建立命名空間)。您可以使用 ListOperations 檢視操作的狀態。如需詳細資訊，請參閱 AWS Cloud Map API 參考中的 ListOperations

AWS CLI

- 使用您偏好的執行個體探索類型（使用您自己的值取代##值）的命令來建立命名空間。
  - 使用 建立 HTTP 命名空間create-http-namespace。使用 HTTP 命名空間註冊的服務執行個體可以使用DiscoverInstances請求來探索，但無法使用 DNS 來探索。

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

  - 根據 DNS 建立私有命名空間，且只能使用 在指定的 Amazon VPC 內看見create-private-dns-namespace。您可以使用DiscoverInstances請求或使用 DNS，探索已向私有 DNS 命名空間註冊的執行個體

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

  - 使用 ，根據網際網路上可見的 DNS 建立公有命名空間create-public-dns-namespace。您可以使用 DiscoverInstances 請求或 DNS，探索已向公有 DNS 命名空間註冊的執行個體。

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

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 servicediscovery做為您的服務。

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

3. 使用您偏好的執行個體探索類型的 命令來建立命名空間（使用您自己的值取代##值）：

- 使用 建立 HTTP 命名空間`create_http_namespace()`。使用 HTTP 命名空間註冊的服務執行個體可以使用 進行探索`discover_instances()`，但無法使用 DNS 進行探索。

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

- 根據 DNS 建立私有命名空間，且只能使用 在指定的 Amazon VPC 內看見`create_private_dns_namespace()`。您可以使用 `discover_instances()`或使用 DNS，探索已向私有 DNS 命名空間註冊的執行個體

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

- 使用 ，根據網際網路上可見的 DNS 建立公有命名空間`create_public_dns_namespace()`。您可以使用 `discover_instances()`或使用 DNS，探索已向公有 DNS 命名空間註冊的執行個體。

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

- 回應輸出範例

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

後續步驟

建立命名空間後，您可以在命名空間中建立 服務，將集合的應用程式資源分組，共同用於應用程式中的特定用途。服務可做為將應用程式資源註冊為執行個體的範本。如需建立 AWS Cloud Map 服務的詳細資訊，請參閱 為應用程式元件建立 AWS Cloud Map 服務。

列出 AWS Cloud Map 命名空間

建立命名空間後，您可以依照下列步驟檢視已建立的命名空間清單。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇命名空間以檢視命名空間清單。您可以依名稱、描述、執行個體探索模式或命名空間 ID 來排序命名空間。您也可以在搜尋欄位中輸入命名空間名稱或 ID，以尋找和檢視特定命名空間。

AWS CLI

- 使用 `list-namespaces`命令列出命名空間。

```
aws servicediscovery list-namespaces
```

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 `servicediscovery`做為您的服務。

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

3. 使用 列出命名空間`list_namespaces()`。

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

回應輸出範例

```
{
    'Namespaces': [
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1585354387.357,
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'myFirstNamespace',
            'Properties': {
                'DnsProperties': {
                    'HostedZoneId': 'Z06752353VBUDTC32S84S',
                },
                'HttpProperties': {
                    'HttpName': 'myFirstNamespace',
                },
            },
            'Type': 'DNS_PRIVATE',
        },
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1586468974.698,
            'Description': 'My second namespace',
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'mySecondNamespace.com',
            'Properties': {
                'DnsProperties': {
                },
                'HttpProperties': {
                    'HttpName': 'mySecondNamespace.com',
                },
            },
            'Type': 'HTTP',
        },
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1587055896.798,
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'myThirdNamespace.com',
            'Properties': {
```

```
                'DnsProperties': {
                    'HostedZoneId': 'Z09983722P0QME1B3KC8I',
                },
                'HttpProperties': {
                    'HttpName': 'myThirdNamespace.com',
                },
            },
            'Type': 'DNS_PRIVATE',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

刪除 AWS Cloud Map 命名空間

使用命名空間完成後，您可以將其刪除。刪除命名空間時，您即無法再使用該空間來註冊或探索服務執行個體。

> **Note**
>
> 當您建立命名空間時，如果您指定要使用公有 DNS 查詢或 VPCs中的 DNS 查詢來探索服務執行個體， 會 AWS Cloud Map 建立 Amazon Route 53 公有或私有託管區域。當您刪除命名空間時， 會 AWS Cloud Map 刪除對應的託管區域。

刪除命名空間之前，您必須先取消註冊所有服務執行個體，然後刪除在命名空間中建立的所有服務。如需詳細資訊，請參閱 取消註冊 AWS Cloud Map 服務執行個體 和 刪除 AWS Cloud Map 服務。

取消註冊執行個體並刪除在命名空間中建立的服務之後，請依照下列步驟刪除命名空間。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。
3. 選取您要刪除的命名空間，然後選擇刪除。
4. 再次選擇刪除，確認您想要刪除服務。

AWS CLI

- 使用 `delete-namespace`命令刪除命名空間（使用您自己的值取代##值）。如果命名空間仍包含一或多個 服務，請求會失敗。

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

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 `servicediscovery`做為您的服務。

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

3. 使用 刪除命名空間 `delete_namespace()`（使用您自己的值取代##值）。如果命名空間仍包含一或多個 服務，請求會失敗。

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

回應輸出範例

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

AWS Cloud Map 服務

AWS Cloud Map 服務是註冊服務執行個體的範本，其中包含服務的 服務名稱和 DNS 組態，如果適用的話。您也可以設定運作狀態檢查，以判斷服務中執行個體的運作狀態，並篩選掉運作狀態不佳的資源。服務可以代表應用程式的元件。例如，您可以為處理應用程式付款的資源建立 服務，並為管理使用者的資源建立 服務。

服務可讓您透過返回一或多個端點來尋找應用程式的資源，這些端點可用來連線至資源。資源的位置是使用 DNS 查詢或 API AWS Cloud Map `DiscoverInstances`動作完成，取決於您設定命名空間的方式。您可以使用 AWS Cloud Map 主控台在服務層級範圍執行個體探索。

您也可以使用 `UpdateServiceAttributes` API，在服務層級將自訂中繼資料指定為屬性。您可以設定服務屬性，以避免跨執行個體複製屬性，並修改這些屬性，而不需要對執行個體屬性進行任何變更。您可以在服務層級指定為屬性的資訊包括但不限於下列項目：

- 在漸進式部署期間轉移流量的端點權重。
- API 逾時和建議的重試政策等服務偏好設定。

如需詳細資訊，請參閱 AWS Cloud Map API 參考中的 UpdateServiceAttributes。

下列主題說明 服務的運作狀態檢查和 DNS 組態，並包含建立、列出、更新和刪除服務的指示。

主題

- AWS Cloud Map 服務運作狀態檢查組態
- AWS Cloud Map 服務 DNS 組態
- 為應用程式元件建立 AWS Cloud Map 服務
- 更新 AWS Cloud Map 服務
- 列出命名空間中的 AWS Cloud Map 服務
- 刪除 AWS Cloud Map 服務

AWS Cloud Map 服務運作狀態檢查組態

運作狀態檢查有助於判斷服務執行個體是否正常運作。如果您在服務建立期間未設定運作狀態檢查，無論執行個體的運作狀態為何，流量都會路由至服務執行個體。當您設定運作狀態檢查時，

預設會 AWS Cloud Map 傳回運作狀態良好的資源。您可以使用 `DiscoverInstances` API 的 `HealthStatus` 參數，依運作狀態篩選資源，並取得運作狀態不佳的資源清單。您也可以使用 `GetInstancesHealthStatus` API 來擷取特定服務執行個體的運作狀態。

您可以在建立 AWS Cloud Map 服務時設定 Route 53 運作狀態檢查或自訂的第三方運作狀態檢查。

Route 53 運作狀態檢查

如果您指定 Amazon Route 53 運作狀態檢查的設定， 會在您註冊執行個體時 AWS Cloud Map 建立 Route 53 運作狀態檢查，並在取消註冊執行個體時刪除運作狀態檢查。

對於公有 DNS 命名空間， 會將運作狀態檢查與註冊執行個體時 AWS Cloud Map 建立的 Route 53 記錄建立 AWS Cloud Map 關聯。如果您在服務的 DNS 組態中同時指定 `A`和 `AAAA`記錄類型， 會 AWS Cloud Map 建立使用 IPv4 地址來檢查資源運作狀態的運作狀態檢查。如果 IPv4 地址指定的端點狀況不良，Route 53 會將 `A`和 `AAAA`記錄視為狀況不良。如果您在服務的 DNS 組態中指定`CNAME`記錄類型，則無法設定 Route 53 運作狀態檢查。

對於您使用 API 呼叫來探索執行個體的命名空間， AWS Cloud Map 會建立 Route 53 運作狀態檢查。不過，沒有 AWS Cloud Map 要與運作狀態檢查建立關聯的 DNS 記錄。若要判斷運作狀態檢查是否良好，您可以使用 Route 53 主控台或使用 Amazon CloudWatch 設定監控。如需使用 Route 53 主控台的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的在運作狀態檢查失敗時取得通知。如需使用 CloudWatch 的詳細資訊，請參閱《Amazon CloudWatch API 參考》中的 PutMetricAlarm。

> **Note**
>
> - 您無法為在私有 DNS 命名空間中建立的服務設定 Amazon Route 53 運作狀態檢查。
> - 每個運作狀態檢查中的 Route 53 運作狀態檢查程式會每 30 秒 AWS 區域 將運作狀態檢查請求傳送至端點。您的端點平均約每隔兩秒就會收到一次運作狀態檢查請求。但是，運作狀態檢查程式不會彼此協調。因此，有時會看到一秒數個請求，接下來幾秒又完全沒有運作狀態檢查的情況。如需運作狀態檢查區域的清單，請參閱 區域。

如需 Route 53 運作狀態檢查費用的資訊，請參閱 Route 53 定價。

自訂運作狀態檢查

如果您 AWS Cloud Map 將 設定為在註冊執行個體時使用自訂運作狀態檢查，則必須使用第三方運作狀態檢查程式來評估資源的運作狀態。在以下情況下自訂運作狀態檢查很有用：

- 您無法使用 Route 53 運作狀態檢查，因為資源無法透過網際網路使用。例如，假設您的執行個體位於 Amazon VPC 中。您可以使用此執行個體的自訂運作狀態檢查。不過，若要讓運作狀態檢查正常運作，您的運作狀態檢查程式也必須與執行個體位於相同的 VPC 中。
- 不論資源位於何處，建議您使用第三方運作狀態檢查程式。

當您使用自訂運作狀態檢查時， AWS Cloud Map 不會直接檢查指定資源的運作狀態。相反地，第三方運作狀態檢查程式會檢查資源的運作狀態，並將狀態傳回給您的應用程式。然後，您的應用程式將需要提交轉送此狀態的`UpdateInstanceCustomHealthStatus`請求 AWS Cloud Map。如果初始轉送狀態為 `UNHEALTHY`，且如果 30 秒`UpdateInstanceCustomHealthStatus`內沒有另一個轉送狀態為 `HEALTHY`，則確認資源運作狀態不佳。 會 AWS Cloud Map 停止將流量路由至該資源。

AWS Cloud Map 服務 DNS 組態

當您在支援 DNS 查詢執行個體探索的命名空間中建立服務時， 會 AWS Cloud Map 建立 Route 53 DNS 記錄。您必須指定將套用至 AWS Cloud Map 建立之所有 Route 53 DNS 記錄的 Route 53 路由政策和 DNS 記錄類型。

路由政策

路由政策會決定 Route 53 如何回應用於服務執行個體探索的 DNS 查詢。支援的路由政策及其關聯方式 AWS Cloud Map 如下。

加權路由

Route 53 會從您使用相同服務註冊的執行個體中，從一個隨機選取的 AWS Cloud Map AWS Cloud Map 服務執行個體傳回適用的值。所有記錄的權重都相同，因此您無法將更多或更少的流量路由到任何執行個體。

例如，假設服務包含一個 A 記錄和運作狀態檢查的組態，而您使用服務註冊 10 個執行個體。Route 53 使用從運作狀態良好的執行個體中隨機選取的執行個體 IP 地址回應 DNS 查詢。如果沒有正常運作的執行個體，Route 53 會回應 DNS 查詢，就好像所有執行個體都正常運作一樣。

如未定義服務的運作狀態檢查，Route 53 會假設所有執行個體都運作狀況良好，並傳回其中一個隨機選取執行個體的適當值。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的加權路由。

多值回答路由

如果您定義服務的運作狀態檢查，且運作狀態檢查的結果良好，Route 53 會傳回最多八個執行個體的適用值。

例如，假設服務包含一個 A 記錄和運作狀態檢查的組態。您使用此服務登錄 10 個執行個體。Route 53 以 IP 地址回應 DNS 查詢，最多八個運作狀態良好的執行個體。如果少於八個執行個體正常運作，Route 53 會使用所有正常運作執行個體的 IP 地址來回應每個 DNS 查詢。

如不定義服務的運作狀態檢查，Route 53 會假設所有執行個體都運作狀態良好，並傳回最多八個執行個體的值。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的多值回答路由。

記錄類型

Route 53 DNS 記錄類型會決定 Route 53 傳回的值類型，以回應用於服務執行個體探索的 DNS 查詢。您可以指定的不同 DNS 記錄類型，以及 Route 53 傳回以回應查詢的相關值，如下所示。

A

如果您指定此類型，Route 53 會以 IPv4 格式傳回資源的 IP 地址，例如 192.0.2.44。

AAAA

如果您指定此類型，Route 53 會以 IPv6 格式傳回資源的 IP 地址，例如 2001：0db8：85a3：0000：0000：abcd：0001：2345。

CNAME

如果您指定此類型，Route 53 會傳回資源的網域名稱（例如 www.example.com)。

> **Note**
> - 若要設定 CNAME DNS 記錄，您必須指定加權路由路由政策。
> - 設定 CNAME DNS 記錄時，您無法設定 Route 53 運作狀態檢查。

SRV

如果您指定此類型，Route 53 會傳回SRV記錄的值。SRV 記錄的值會使用以下值：

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

考慮下列各項：

- `priority` 和 `weight` 值都設為 1 且無法變更。
- 對於 `port`，當您註冊執行個體時， AWS Cloud Map 會使用您為連接埠 (AWS_INSTANCE_PORT) 指定的值。
- `service-hostname` 的值為以下值的串接：
  - 您在註冊執行個體時為服務執行個體 ID (InstanceID) 指定的值
  - 服務的名稱
  - 命名空間的名稱

  例如，假設您在註冊執行個體時將測試指定為執行個體 ID。服務的名稱為後端，命名空間的名稱為 example.com。 AWS Cloud Map 會將下列值指派給 SRV 記錄中的 `service-hostname` 屬性：

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

> **Note**
>
> 如果您在註冊執行個體時指定 IPv4 地址、IPv6 地址或兩者的值， AWS Cloud Map 會自動建立與 SRV 記錄中 值同名的 A 和/或 AAAA `service-hostname` 記錄。

您可以使用下列組合來指定記錄類型：

- A
- AAAA
- A (A) 和 AAAA (AAAA)
- CNAME
- SRV (SRV)

如果您指定 A (A) 和 AAAA (AAAA) 記錄類型，您可以在註冊執行個體時指定 IPv4 IP 地址、IPv6 IP 地址或兩者。

為應用程式元件建立 AWS Cloud Map 服務

建立命名空間後，您可以建立 服務來代表應用程式的不同元件，以達成特定目的。例如，您可以為應用程式中處理付款的資源建立服務。

> **Note**
>
> 您無法建立多個可由 DNS 查詢存取的服務，其名稱僅因案例而異（例如 EXAMPLE 和範例）。嘗試這樣做會導致這些服務具有相同的 DNS 名稱。如果您使用僅供 API 呼叫存取的命名空間，則可以建立名稱僅因案例而異的服務。

請依照下列步驟，使用 AWS CLI AWS Management Console和適用於 Python 的 SDK 來建立服務。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。
3. 在 Namespaces (命名空間) 頁面，選擇您要新增服務的命名空間。
4. 在命名空間：*Namespace-name* 頁面上，選擇建立服務。
5. 針對服務名稱，輸入描述您使用此服務時註冊之執行個體的名稱。此值用於在 API 呼叫或 DNS 查詢中探索 AWS Cloud Map 服務執行個體。

   > **Note**
   >
   > 如果您想要在註冊執行個體時 AWS Cloud Map 建立 SRV 記錄，而且您使用的系統需要特定的 SRV 格式（例如 HAProxy)，請針對服務名稱指定下列項目：
   >
   > - 以底線 (_) 開頭名稱，例如 _exampleservice。
   > - 使用 *._protocol* 結束名稱，例如 ._tcp。
   >
   > 當您註冊執行個體時， 會 AWS Cloud Map 建立 SRV 記錄，並透過串連服務名稱和命名空間名稱來指派名稱，例如：
   > _exampleservice._tcp.example.com

6. （選用）針對服務描述，輸入服務的描述。您在此處輸入的描述會顯示在服務頁面和每個服務的詳細資訊頁面上。
7. 如果命名空間支援 DNS 查詢，則在服務探索組態下，您可以在服務層級設定可探索性。選擇允許 API 呼叫和 DNS 查詢，或只允許 API 呼叫來探索此服務中的執行個體。

> **Note**
>
> 如果您選擇 API 呼叫， AWS Cloud Map 將不會在您註冊執行個體時建立 SRV 記錄。

如果您選擇 API 和 DNS，請依照下列步驟設定 DNS 記錄。您可以新增或移除 DNS 記錄。

1. 對於路由政策，為註冊執行個體時建立的 AWS Cloud Map DNS 記錄選取 Amazon Route 53 路由政策。您可以選擇加權路由和多值答案路由。如需詳細資訊，請參閱路由政策。

   > **Note**
   >
   > 您無法使用 主控台來設定 AWS Cloud Map ，以在註冊執行個體時建立 Route 53 別名記錄。如果您想要 AWS Cloud Map 在以程式設計方式註冊執行個體時為 Elastic Load Balancing 負載平衡器建立別名記錄，請選擇路由政策的加權路由。

2. 針對記錄類型，選擇決定 Route 53 傳回哪些項目以回應 DNS 查詢的 DNS 記錄類型 AWS Cloud Map。如需詳細資訊，請參閱記錄類型。
3. 對於 TTL，指定數值，以定義服務層級的存留時間 (TTL) 值，以秒為單位。TTL 的值決定解析程式將此記錄的 DNS 解析程式快取資訊的時間，之後解析程式會將另一個 DNS 查詢轉送到 Amazon Route 53 以取得更新的設定。

8. 在運作狀態檢查組態下，針對運作狀態檢查選項，選擇適用於服務執行個體的運作狀態檢查類型。您可以選擇不設定任何運作狀態檢查，也可以選擇 Route 53 運作狀態檢查或執行個體的外部運作狀態檢查。如需詳細資訊，請參閱AWS Cloud Map 服務運作狀態檢查組態。

   > **Note**
   >
   > Route 53 運作狀態檢查只能針對公有 DNS 命名空間中的服務進行設定。

   如果您選擇 Route 53 運作狀態檢查，請提供下列資訊。

   1. 對於失敗閾值，請提供介於 1 到 10 之間的數字，以定義服務執行個體必須通過或失敗才能變更其運作狀態的連續 Route 53 運作狀態檢查次數。
   2. 針對運作狀態檢查通訊協定，選取 Route 53 將用於檢查服務執行個體運作狀態的方法。

3. 如果您選擇 HTTP 或 HTTPS 運作狀態檢查通訊協定，對於運作狀態檢查路徑，請提供您希望 Amazon Route 53 在執行運作狀態檢查時請求的路徑。路徑可以是任何值，例如檔案 `/docs/route53-health-check.html`。當資源正常運作時，傳回的值是 2xx 或 3xx 格式的 HTTP 狀態碼。您也可以包含查詢字串參數，例如 `/welcome.html?language=jp&login=y`。 AWS Cloud Map 主控台會自動新增前導斜線 (/) 字元。

如需 Route 53 運作狀態檢查的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 Amazon Route 53 如何判斷運作狀態檢查是否正常運作。

9. （選用）在標籤下，選擇新增標籤，然後指定要標記命名空間的索引鍵和值。您可以指定要新增至命名空間的一或多個標籤。標籤可讓您將 AWS 資源分類，讓您更輕鬆地管理資源。如需詳細資訊，請參閱標記您的 AWS Cloud Map 資源。
10. 選擇 Create service (建立服務)。

AWS CLI

- 使用 `create-service`命令建立 服務。將##值取代為您自己的值。

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

輸出：

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

```
                }
            ]
        },
        "CreateDate": 1587081768.334,
        "CreatorRequestId": "567c1193-6b00-4308-bd57-ad38a8822d25"
    }
}
```

AWS SDK for Python (Boto3)

如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。

1. 匯入Boto3並使用 servicediscovery做為您的服務。

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

2. 使用 建立服務create_service()。將##值取代為您自己的值。如需詳細資訊，請參閱 create_service。

```
response = client.create_service(
    DnsConfig={
        'DnsRecords': [
            {
                'TTL': 60,
                'Type': 'A',
            },
        ],
        'NamespaceId': 'ns-xxxxxxxxxxx',
        'RoutingPolicy': 'MULTIVALUE',
    },
    Name='service-name',
    NamespaceId='ns-xxxxxxxxxxx',
)
```

回應輸出範例

```
{
    'Service': {
        'Arn': 'arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxx',
```

```
        'CreateDate': 1587081768.334,
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 60,
                    'Type': 'A',
                },
            ],
            'NamespaceId': 'ns-xxxxxxxxxxx',
            'RoutingPolicy': 'MULTIVALUE',
        },
        'Id': 'srv-xxxxxxxxxxx',
        'Name': 'service-name',
        'NamespaceId': 'ns-xxxxxxxxxxx',
    },
    'ResponseMetadata': {
        '...': '...',
    },
}
```

後續步驟

建立服務之後，您可以將應用程式資源註冊為服務執行個體，其中包含應用程式如何尋找資源的相關資訊。如需註冊 AWS Cloud Map 服務執行個體的詳細資訊，請參閱將資源註冊為 AWS Cloud Map 服務執行個體。

您也可以指定自訂中繼資料，例如端點權重、API 逾時和重試政策，做為建立服務之後的服務屬性。如需詳細資訊，請參閱 AWS Cloud Map API 參考`UpdateServiceAttributes`中的 ServiceAttributes 和 。

更新 AWS Cloud Map 服務

根據服務的組態，您可以更新其標籤、Route 53 運作狀態檢查失敗閾值，以及 DNS 解析程式的存留時間 (TTL)。若要更新服務，請執行下列程序。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。

3. 在命名空間頁面上，選擇建立服務的命名空間。
4. 在命名空間：***Namespace-name*** 頁面上，選取您要編輯的服務，然後選擇檢視詳細資訊。
5. 在服務：***####***頁面上，選擇編輯。

> **Note**
> 您無法使用編輯按鈕工作流程來編輯僅允許執行個體探索 API 呼叫之服務的值。不過，您可以在 Service： ***service-name*** 頁面上新增或移除標籤。

6. 在編輯服務頁面的服務描述下，您可以更新服務的任何先前設定描述，或新增新描述。您也可以新增標籤並更新 DNS 解析程式的 TTL。
7. 在 DNS 組態下，對於 TTL，您可以指定更新的時段，以秒為單位，決定解析程式將此記錄的 DNS 解析程式快取資訊在解析程式將另一個 DNS 查詢轉送到 Amazon Route 53 以取得更新設定之前的時間長度。
8. 如果您已設定 Route 53 運作狀態檢查，對於失敗閾值，您可以指定介於 1 到 10 之間的新數字，以定義服務執行個體必須通過或失敗的連續 Route 53 運作狀態檢查數目，其運作狀態才會變更。
9. 選擇更新服務。

AWS CLI

- 使用 `update-service`命令更新服務 （使用您自己的值取代##值）。

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

輸出：

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

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 `servicediscovery`做為您的服務。

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

3. 使用 更新服務 `update_service()`（使用您自己的值取代##值）。

```
response = client.update_service(
    Id='srv-xxxxxxxxxxx',
    Service={
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 300,
                    'Type': 'A',
                },
            ],
        },
        'Description': "new description",
    }
)
```

回應輸出範例

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

列出命名空間中的 AWS Cloud Map 服務

若要檢視您在命名空間中建立的服務清單，請執行以下程序。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。

3. 選擇包含您要列出之服務的命名空間名稱。您可以在服務下檢視所有服務的清單，並在搜尋欄位中輸入服務名稱或 ID，以尋找特定服務。

AWS CLI

- 使用 `list-services`命令列出服務。下列命令會使用命名空間 ID 做為篩選條件，列出命名空間中的所有 服務。將##值取代為您自己的值。

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

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 `servicediscovery`做為您的服務。

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

3. 使用 列出服務`list_services()`。

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

回應輸出範例

```
{
    'Services': [
        {
            'Arn': 'arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxxxxxxx',
            'CreateDate': 1587081768.334,
            'DnsConfig': {
                'DnsRecords': [
                    {
                        'TTL': 60,
                        'Type': 'A',
                    },
```

```
                ],
                'RoutingPolicy': 'MULTIVALUE',
            },
            'Id': 'srv-xxxxxxxxxxxxxxx',
            'Name': 'myservice',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

刪除 AWS Cloud Map 服務

在可以刪除服務前，您必須取消註冊使用該服務註冊的所有服務執行個體。如需詳細資訊，請參閱取消註冊 AWS Cloud Map 服務執行個體。

取消註冊使用 服務註冊的所有執行個體後，請執行下列程序來刪除服務。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。
3. 選擇包含您要刪除之服務的命名空間選項。
4. 在命名空間：***Namespace-name*** 頁面上，選擇要刪除之服務的 選項。
5. 選擇 刪除 。
6. 確認您要刪除該服務。

AWS CLI

- 使用 `delete-service`命令刪除服務 （使用您自己的值取代##值）。

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

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 `servicediscovery`做為您的服務。

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

3. 使用 刪除服務 `delete_service()`（使用您自己的值取代##值）。

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

回應輸出範例

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

AWS Cloud Map 服務執行個體

服務執行個體會包含如何尋找應用程式資源 (像是 web 伺服器) 的相關資訊。註冊執行個體之後，您可以使用 DNS 查詢或 AWS Cloud Map DiscoverInstances API 動作來尋找執行個體。您可以註冊的資源包括但不限於下列項目：

- Amazon EC2 執行個體
- Amazon DynamoDB 資料表
- Amazon S3 儲存貯體
- Amazon Simple Queue Service (Amazon SQS) 佇列
- 部署在 APIs Amazon API Gateway

您可以指定服務執行個體的屬性值，而用戶端可以使用這些屬性來篩選 AWS Cloud Map 傳回的資源。例如，應用程式可以要求在特定部署階段 (像是 BETA 或 PROD) 的資源。您也可以使用屬性進行版本控制。

下列程序說明如何將應用程式中的資源註冊為服務執行個體、檢視服務中已註冊執行個體的清單、編輯特定執行個體參數，以及取消註冊執行個體。

主題

- 將資源註冊為 AWS Cloud Map 服務執行個體
- 列出 AWS Cloud Map 服務執行個體
- 更新 AWS Cloud Map 服務執行個體
- 取消註冊 AWS Cloud Map 服務執行個體

將資源註冊為 AWS Cloud Map 服務執行個體

您可以在 AWS Cloud Map 服務中將應用程式的資源註冊為執行個體。例如，假設您已建立`users`針對管理使用者資料的所有應用程式資源呼叫 的服務。然後，您可以註冊 DynamoDB 資料表，該資料表用於將使用者資料儲存為此服務中的執行個體。

> **Note**
>
> 下列功能無法在 AWS Cloud Map 主控台上使用：

- 當您使用主控台註冊服務執行個體時，您無法建立別名記錄，將流量路由到 Elastic Load Balancing (ELB) 負載平衡器。註冊執行個體時，您必須包含 `AWS_ALIAS_DNS_NAME` 屬性。如需詳細資訊，請參閱 AWS Cloud Map API 參考中的 RegisterInstance。
- 如果您使用包含自訂運作狀態檢查的服務註冊執行個體，您無法為自訂運作狀態檢查指定初始狀態。自訂運作狀態檢查的初始運作狀態預設是 Healthy (良好)。如果您希望初始運作狀態是 Unhealthy (不良)，請以程式設計的方式註冊執行個體並包含 `AWS_INIT_HEALTH_STATUS` 屬性。如需詳細資訊，請參閱 AWS Cloud Map API 參考中的 RegisterInstance。

若要在 服務中註冊執行個體，請遵循下列步驟。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。
3. 在 Namespaces (命名空間) 頁面中，選擇包含您要用做為註冊服務執行個體範本之服務的命名空間。
4. 在命名空間：***Namespace-name*** 頁面上，選擇您要使用的服務。
5. 在服務：***####***頁面上，選擇註冊服務執行個體。
6. 在註冊服務執行個體頁面上，選擇執行個體類型。根據命名空間執行個體探索組態，您可以選擇為沒有 IP 地址的資源指定 IP 地址、Amazon EC2 執行個體 ID 或其他識別資訊。

   > **Note**
   > 您只能在 HTTP 命名空間中選擇 EC2 執行個體。

7. 針對服務執行個體 ID，提供與服務執行個體相關聯的識別符。

   > **Note**
   > 如果您想要更新現有的執行個體，請提供與要更新之執行個體相關聯的識別符。然後，使用後續步驟來更新值並重新註冊執行個體。

8. 根據您選擇的執行個體類型，執行下列步驟。

> **Important**
>
> 當您指定自訂屬性時，無法在金鑰中使用AWS_字首（不區分大小寫）。

執行個體類型	步驟	
IP 地址	a. 在標準屬性下，對於 IPv4 地址，如果有，請提供 IPv4 地址，您的應用程式可以存取與此服務執行個體相關聯的資源。 b. 針對 IPv6 地址，請提供 IPv6 IP 地址，如果有的話，您的應用程式可以存取與此服務執行個體相關聯的資源。 c. 針對連接埠，指定您的應用程式必須包含的任何連接埠，以存取與此服務執行個體相關聯的資源。當服務包含 SRV 記錄或 Amazon Route 53 運作狀態檢查時，需要連接埠。 d. （選用）在自訂屬性下，指定您要與資源建立關聯的任何鍵/值對。	

執行個體類型	步驟
EC2 執行個體	a. 針對 EC2 執行個體 ID，選取您要註冊為 AWS Cloud Map 服務執行個體的 Amazon EC2 執行個體 ID。 b. （選用）在自訂屬性下，指定您要與資源建立關聯的任何鍵/值對。
為另一個資源識別資訊	a. 在標準屬性下，如果服務組態包含 CNAME DNS 記錄，您會看到 CNAME 欄位。針對 CNAME，指定您希望 Route 53 傳回的網域名稱，以回應 DNS 查詢（例如 `example.com`）。 b. 在自訂屬性下，為非 IP 地址或 Amazon EC2 執行個體 ID 的資源指定任何識別資訊，做為索引鍵/值對。例如，您可以指定名為 的金鑰`function`，並提供 Lambda 函數的名稱做為值，以註冊 Lambda 函數。您也可以指定名為 的金鑰，`name`並提供可用於程式設計執行個體探索的名稱。

9. 選擇 Register service instance (註冊服務執行個體)。

AWS CLI

- 當您提交RegisterInstance請求時：
  - 對於您在 指定的服務中定義的每個 DNS 記錄ServiceId，會在與對應命名空間相關聯的託管區域中建立或更新記錄。
  - 如果服務包含 HealthCheckConfig，會根據運作狀態檢查組態中的設定建立運作狀態檢查。
  - 任何運作狀態檢查都會與每個新的或更新的記錄相關聯。

  使用 register-instance命令註冊服務執行個體 （使用您自己的值取代##值）。

```
aws servicediscovery register-instance \
    --service-id srv-xxxxxxxxx \
    --instance-id myservice-xx \
    --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 servicediscovery做為您的服務。

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

3. 當您提交RegisterInstance請求時：
   - 對於您在 指定的服務中定義的每個 DNS 記錄ServiceId，會在與對應命名空間相關聯的託管區域中建立或更新記錄。
   - 如果服務包含 HealthCheckConfig，會根據運作狀態檢查組態中的設定建立運作狀態檢查。
   - 任何運作狀態檢查都會與每個新的或更新的記錄相關聯。

   向 註冊服務執行個體 register_instance()（使用您自己的值取代##值）。

```
response = client.register_instance(
    Attributes={
```

```
        'AWS_INSTANCE_IPV4': '172.2.1.3',
        'AWS_INSTANCE_PORT': '808',
    },
    InstanceId='myservice-xx',
    ServiceId='srv-xxxxxxxxx',
)
# If you want to see the response
print(response)
```

回應輸出範例

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

列出 AWS Cloud Map 服務執行個體

若要檢視您使用服務註冊的服務執行個體清單，請執行以下程序。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。
3. 選擇包含您要列出服務執行個體之服務的命名空間名稱。
4. 選擇您用來建立服務執行個體的服務名稱。您會在服務執行個體下看到執行個體清單。您可以在搜尋欄位中輸入執行個體 ID，以列出特定執行個體。

AWS CLI

- 使用 `list-instances`命令列出服務執行個體 （使用您自己的值取代##值）。

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

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 `servicediscovery`做為您的服務。

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

3. 使用 列出服務執行個體 `list_instances()`（使用您自己的值取代##值）。

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

回應輸出範例

```
{
    'Instances': [
        {
            'Attributes': {
                'AWS_INSTANCE_IPV4': '172.2.1.3',
                'AWS_INSTANCE_PORT': '808',
            },
            'Id': 'i-xxxxxxxxxxxxxxxxx',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

更新 AWS Cloud Map 服務執行個體

您可以根據您要更新哪些值，透過下列兩種方式更新服務執行個體：

- 更新任何值：如果您想要更新註冊服務執行個體時為服務執行個體指定的任何值，包括自訂屬性，則需要重新註冊服務執行個體並重新指定所有值。請遵循 中的步驟將資源註冊為 AWS Cloud Map 服務執行個體，為服務執行個體 ID 指定現有服務執行個體的執行個體 ID。

或者，您可以使用 `RegisterInstance` API。您可以使用 和 `InstanceId` `ServiceId` 參數指定現有執行個體和服務 ID，並重新指定其他值。

- 僅更新自訂屬性：如果您只要更新服務執行個體的自訂屬性，則不需要重新註冊執行個體。您可以僅更新這些值。請參閱 更新服務執行個體的自訂屬性。

更新服務執行個體的自訂屬性

只要更新服務執行個體的自訂屬性

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。
3. 在 Namespaces (命名空間) 頁面中，選擇包含您原本要用來註冊服務執行個體之服務的命名空間。
4. 在命名空間： ***namespace-name*** 頁面上，選擇您用來註冊服務執行個體的服務。
5. 在 Service: ***service-name*** (服務：service-name) 頁面中，選擇您要更新的服務執行個體名稱。
6. 在 Custom attributes (自訂屬性) 區段中，選擇 Edit (編輯)。
7. 在 Edit service instance: ***instance-name*** (編輯服務執行個體：instance-name) 頁面上，新增、移除或更新自訂屬性。您可以同時更新現有屬性的索引鍵和值。
8. 選擇 Update service instance (更新服務執行個體)。

取消註冊 AWS Cloud Map 服務執行個體

在可以刪除服務前，您必須取消註冊使用該服務註冊的所有服務執行個體。

若要取消註冊服務執行個體，請執行以下程序。

AWS Management Console

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/cloudmap/ 開啟 AWS Cloud Map 主控台。
2. 在導覽窗格中，選擇 Namespaces (命名空間)。
3. 選擇包含您要取消註冊之服務執行個體的命名空間選項。
4. 在命名空間：***Namespace-name*** 頁面上，選擇您用來註冊服務執行個體的服務。

5. 在服務：####頁面上，選擇您要取消註冊的服務執行個體。
6. 選擇 Deregister (取消註冊)。
7. 確認是否要取消註冊此服務執行個體。

AWS CLI

- 使用 deregister-instance命令取消註冊服務執行個體（使用您自己的值取代##值）。此命令會刪除 Amazon Route 53 DNS 記錄，以及為指定執行個體 AWS Cloud Map 建立的任何運作狀態檢查。

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

AWS SDK for Python (Boto3)

1. 如果您尚未Boto3安裝 ，您可以Boto3在這裡找到安裝、設定和使用 的指示。
2. 匯入Boto3並使用 servicediscovery做為您的服務。

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

3. 使用 取消註冊服務執行個體 deregister-instance()（使用您自己的值取代##值）。此命令會刪除 Amazon Route 53 DNS 記錄，以及為指定執行個體 AWS Cloud Map 建立的任何運作狀態檢查。

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

回應輸出範例

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

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

中的安全性 AWS Cloud Map

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。在 AWS 合規計畫中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用的合規計劃 AWS Cloud Map，請參閱AWS 合規計劃範圍內的服務。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。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。

下列文件可協助您了解如何在使用 時套用共同責任模型 AWS Cloud Map。下列主題說明如何設定 AWS Cloud Map 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS Cloud Map 資源。

主題

- 的 Identity and Access Management AWS Cloud Map
- 的合規驗證 AWS Cloud Map
- 中的彈性 AWS Cloud Map
- 中的基礎設施安全 AWS Cloud Map

的 Identity and Access Management AWS Cloud Map

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行驗證 （登入） 和授權 （具有許可） 來使用 AWS Cloud Map 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- AWS Cloud Map 如何使用 IAM

- 的身分型政策範例 AWS Cloud Map
- AWS 的 受管政策 AWS Cloud Map
- AWS Cloud Map API 許可參考
- 對 AWS Cloud Map 身分和存取進行故障診斷

目標對象

使用方式 AWS Identity and Access Management (IAM) 會有所不同，取決於您在其中執行的工作 AWS Cloud Map。

服務使用者 – 如果您使用 AWS Cloud Map 服務來執行任務，您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS Cloud Map 功能來執行工作時，您可能需要額外的許可。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 AWS Cloud Map中的某項功能，請參閱 對 AWS Cloud Map 身分和存取進行故障診斷。

服務管理員 – 如果您負責公司 AWS Cloud Map 的資源，您可能擁有 的完整存取權 AWS Cloud Map。您的任務是判斷服務使用者應存取哪些 AWS Cloud Map 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何搭配 使用 IAM AWS Cloud Map，請參閱 AWS Cloud Map 如何使用 IAM。

IAM 管理員：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 AWS Cloud Map存取權的詳細資訊。若要檢視您可以在 IAM 中使用的以 AWS Cloud Map 身分為基礎的政策範例，請參閱 的身分型政策範例 AWS Cloud Map。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的 憑證以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需

使用建議的方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的適用於 API 請求的AWS Signature 第 4 版。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來提高帳戶的安全性。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的多重要素驗證和《IAM 使用者指南》中的 IAM 中的AWS 多重要素驗證。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取帳戶中的所有 AWS 服務 和資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳實務是， 要求人類使用者，包括需要管理員存取權的使用者，使用 聯合身分提供者 AWS 服務 來使用臨時憑證來存取 。

聯合身分是來自您的企業使用者目錄、Web 身分提供者、 AWS Directory Service、Identity Center 目錄或任何使用透過身分來源提供的登入資料 AWS 服務 存取的使用者。當聯合身分存取時 AWS 帳戶，它們會擔任 角色，而角色會提供臨時登入資料。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群組，以便在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是 中的身分 AWS 帳戶 ，具有單一人員或應用程式的特定許可。建議您盡可能依賴臨時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色（主控台）。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱《IAM 使用者指南》中的為第三方身分提供者 (聯合) 建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以將政策直接連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有在服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。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《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。

- 服務連結角色 – 服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件， AWS 當與身分或資源建立關聯時， 會定義其許可。當委託人（使用者、根使用者或角色工作階段）發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，請參閱《IAM 使用者指南》中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 之多個 的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括支援 RCPs AWS 服務 的 清單，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 (RCPs)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作

階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS Cloud Map 如何使用 IAM

在您使用 IAM 管理對 的存取之前 AWS Cloud Map，請先了解可使用哪些 IAM 功能 AWS Cloud Map。

IAM 功能	AWS Cloud Map 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
轉送存取工作階段 (FAS)	是
服務角色	否
服務連結角色	否

若要全面了解 AWS Cloud Map 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱《AWS IAM 使用者指南》中的與 IAM 搭配使用的 服務。

的身分型政策 AWS Cloud Map

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

的身分型政策範例 AWS Cloud Map

若要檢視 AWS Cloud Map 身分型政策的範例，請參閱 的身分型政策範例 AWS Cloud Map。

中的資源型政策 AWS Cloud Map

支援資源型政策：否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當委託人和資源位於不同位置時 AWS 帳戶，信任帳戶中的 IAM 管理員也必須授予委託人實體 （使用者或角色） 存取資源的許可。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

的政策動作 AWS Cloud Map

支援政策動作：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS Cloud Map 動作清單，請參閱《服務授權參考》中的 定義的動作 AWS Cloud Map。

中的政策動作在動作之前 AWS Cloud Map 使用下列字首：

```
servicediscovery
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

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

若要檢視 AWS Cloud Map 身分型政策的範例，請參閱 的身分型政策範例 AWS Cloud Map。

的政策資源 AWS Cloud Map

支援政策資源：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。陳述式必須包含 `Resource` 或 `NotResource` 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

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

若要查看 AWS Cloud Map 資源類型及其 ARNs，請參閱《服務授權參考》中的 定義的資源 AWS Cloud Map。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 AWS Cloud Map定義的動作。

若要檢視 AWS Cloud Map 身分型政策的範例，請參閱 的身分型政策範例 AWS Cloud Map。

的政策條件索引鍵 AWS Cloud Map

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素 (或 `Condition` 區塊) 可讓您指定使陳述式生效的條件。`Condition` 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 `Condition` 元素，或是在單一 `Condition` 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值， 會使用邏輯OR操作 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定的條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 AWS Cloud Map 條件索引鍵的清單，請參閱《服務授權參考》中的 的條件索引鍵 AWS Cloud Map。若要了解您可以使用條件索引鍵的動作和資源，請參閱 定義的動作 AWS Cloud Map。

AWS Cloud Map 支援下列服務特定的條件金鑰，您可以用來為 IAM 政策提供精細篩選。

`servicediscovery:NamespaceArn`

可讓您透過指定相關命名空間的 Amazon Resource Name (ARN) 來取得物件的篩選條件。

`servicediscovery:NamespaceName`

可讓您透過指定相關命名空間的名稱來取得物件的篩選條件。

`servicediscovery:ServiceArn`

可讓您透過指定相關服務的 Amazon Resource Name (ARN) 來取得物件的篩選條件。

`servicediscovery:ServiceName`

可讓您透過指定相關服務的名稱來取得物件的篩選條件。

若要檢視 AWS Cloud Map 身分型政策的範例，請參閱 的身分型政策範例 AWS Cloud Map。

中的 ACLs AWS Cloud Map

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 搭配 AWS Cloud Map

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在 中 AWS，這些屬性稱為標籤。您可以將標籤連接至 IAM 實體 （使用者或角色） 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的使用屬性型存取控制 (ABAC)。

搭配 使用臨時登入資料 AWS Cloud Map

支援臨時憑證：是

當您使用臨時登入資料登入時，有些 AWS 服務 無法運作。如需詳細資訊，包括哪些 AWS 服務 使用臨時登入資料，請參閱《AWS 服務 IAM 使用者指南》中的 使用 IAM。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則會使用臨時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的從使用者切換至 IAM 角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

轉送 的存取工作階段 AWS Cloud Map

支援轉寄存取工作階段 (FAS)：是

當您使用 IAM 使用者或角色在 中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱轉發存取工作階段。

AWS Cloud Map的服務角色

支援服務角色：否

服務角色是服務擔任的 IAM 角色，可代您執行動作。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《IAM 使用者指南》中的建立角色以委派許可給 AWS 服務服務。

> **Warning**
> 變更服務角色的許可有可能會讓 AWS Cloud Map 功能出現故障。只有在 AWS Cloud Map 提供指引時，才能編輯服務角色。

的服務連結角色 AWS Cloud Map

支援服務連結角色：否

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱可搭配 IAM 運作的AWS 服務。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

的身分型政策範例 AWS Cloud Map

根據預設，使用者和角色不具備建立或修改 AWS Cloud Map 資源的權限。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行任務。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需 定義的動作和資源類型的詳細資訊 AWS Cloud Map，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的 的動作、資源和條件索引鍵 AWS Cloud Map。

主題

- 政策最佳實務
- 使用 AWS Cloud Map 主控台
- AWS Cloud Map 主控台存取範例
- AWS Cloud Map 允許使用者檢視自己的許可
- 允許所有 AWS Cloud Map 資源的讀取存取權
- AWS Cloud Map 服務執行個體範例
- 建立 AWS Cloud Map 服務範例
- 建立 AWS Cloud Map 命名空間範例

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 AWS Cloud Map 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並轉向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 等使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access

Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。

- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_configure-api-require.html中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

使用 AWS Cloud Map 主控台

若要存取 AWS Cloud Map 主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 AWS Cloud Map 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

為了確保使用者和角色仍然可以使用 AWS Cloud Map 主控台，請將 AWS Cloud Map *ConsoleAccess*或 *ReadOnly* AWS 受管政策連接到實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

AWS Cloud Map 主控台存取範例

若要授予 AWS Cloud Map 主控台的完整存取權，您可以在下列許可政策中授予許可：

JSON

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

```
            "route53:ChangeResourceRecordSets",
            "route53:CreateHealthCheck",
            "route53:GetHealthCheck",
            "route53:DeleteHealthCheck",
            "route53:UpdateHealthCheck",
            "ec2:DescribeInstances",
            "ec2:DescribeVpcs",
            "ec2:DescribeRegions"
         ],
         "Resource":"*"
      }
   ]
}
```

需要許可的原因如下：

`servicediscovery:*`

可讓您執行所有 AWS Cloud Map 動作。

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

可讓您在建立和刪除公有和私有 DNS 命名空間時 AWS Cloud Map 管理託管區域。

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

當您在建立服務時包含 Amazon Route 53 運作狀態檢查時，讓 AWS Cloud Map 管理運作狀態檢查。

`ec2:DescribeVpcs` 和 **`ec2:DescribeRegions`**

讓 AWS Cloud Map 管理私有託管區域。

AWS Cloud Map 允許使用者檢視自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

```
{
    "Version": "2012-10-17",
```

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

允許所有 AWS Cloud Map 資源的讀取存取權

下列許可政策會授予使用者所有 AWS Cloud Map 資源的唯讀存取許可：

JSON

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
```

```
            "servicediscovery:Get*",
            "servicediscovery:List*",
            "servicediscovery:DiscoverInstances"
         ],
         "Resource":"*"
      }
   ]
}
```

AWS Cloud Map 服務執行個體範例

下列範例顯示許可政策，授予使用者註冊、取消註冊和探索服務執行個體的許可。`Sid` (陳述式 ID) 為選用：

JSON

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

該政策會授予註冊和管理服務執行個體所需動作的許可。如果您使用公有或私有 DNS 命名空間，因為 會在您註冊和取消註冊執行個體時 AWS Cloud Map 建立、更新和刪除 Route 53 記錄和運作狀態檢查，因此需要 Route 53 許可。中的萬用字元 (*) `Resource`授予所有 AWS Cloud Map 執行個體的存取權，以及目前 AWS 帳戶擁有的 Route 53 記錄和運作狀態檢查。

建立 AWS Cloud Map 服務範例

新增許可政策以允許 IAM 身分建立 AWS Cloud Map 服務時，您必須在資源欄位中指定命名空間和服務的 Amazon Resource Name (ARN) AWS Cloud Map 。ARN 包含區域、帳戶 ID 和命名空間 ID。由於您不知道服務的服務 ID 為何，我們建議您使用萬用字元。以下是範例政策程式碼片段。

JSON

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

建立 AWS Cloud Map 命名空間範例

下列許可政策允許使用者建立所有類型的 AWS Cloud Map 命名空間：

JSON

```
{
   "Version": "2012-10-17",
   "Statement":[
```

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

AWS 的 受管政策 AWS Cloud Map

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住， AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受 AWS 管政策中定義的許可，則更新會影響政策連接的所有委託人身分（使用者、群組和角色）。當新的 AWS 服務 啟動或新的 API 操作可供現有服務使用時， AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 受管政策：AWSCloudMapDiscoverInstanceAccess

您可以將 `AWSCloudMapDiscoverInstanceAccess` 連接到 IAM 實體。提供 AWS Cloud Map Discovery API 的存取權。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSCloudMapDiscoverInstanceAccess。

AWS 受管政策：AWSCloudMapReadOnlyAccess

您可以將 `AWSCloudMapReadOnlyAccess` 連接到 IAM 實體。授予所有 AWS Cloud Map 動作的唯讀存取權。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSCloudMapReadOnlyAccess。

AWS 受管政策：AWSCloudMapRegisterInstanceAccess

您可以將 `AWSCloudMapRegisterInstanceAccess` 連接到 IAM 實體。授予命名空間和服務唯讀存取權，並授予註冊和取消註冊服務執行個體的許可。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSCloudMapRegisterInstanceAccess。

AWS 受管政策：AWSCloudMapFullAccess

您可以將 `AWSCloudMapFullAccess` 連接到 IAM 實體。提供所有 AWS Cloud Map 動作的完整存取權

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSCloudMapFullAccess。

AWS Cloud MapAWS 受管政策的更新

檢視自此服務開始追蹤這些變更 AWS Cloud Map 以來， AWS 受管政策更新的詳細資訊。如需變更的自動提醒，請訂閱 AWS Cloud Map 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	日期
AWSCloudMapDiscoverInstanceAccess、AWSCloudMapRegisterInstanceAccess、AWSCloudMapReadOnlyAccess – 現有政策的更新。	AWS Cloud Map 已更新這些政策，以提供對新 AWS Cloud Map `DiscoverInstanceRevision` API 操作的存取權。	2023 年 8 月 15 日

AWS Cloud Map API 許可參考

當您設定存取控制並撰寫可連接到 IAM 身分的許可政策（以身分為基礎的政策）時，您可以使用下列清單做為參考。此清單包含每個 AWS Cloud Map API 動作，以及您必須授予許可存取權的動作。您可以在政策的 `Action` 欄位中指定動作。如需您必須在 `Resource` 欄位或 IAM 政策中指定之資源值的詳細資訊，請參閱《服務授權參考》中的 的動作、資源和條件索引鍵 AWS Cloud Map。

您可以在某些操作的 IAM 政策中使用 AWS Cloud Map特定條件索引鍵。如需詳細資訊，請參閱《服務授權參考》中的 的條件金鑰 AWS Cloud Map。

若要指定動作，請使用 `servicediscovery` 字首後接 API 動作名稱 (例如，`servicediscovery:CreatePublicDnsNamespace` 和 `route53:CreateHostedZone`)。

AWS Cloud Map 動作所需的許可

CreateHttpNamespace

必要許可 (API 動作）：

- `servicediscovery:CreateHttpNamespace`

CreatePrivateDnsNamespace

必要許可 (API 動作）：

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

CreatePublicDnsNamespace

必要許可 (API 動作）：

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

CreateService

所需許可 (API 動作)：`servicediscovery:CreateService`

DeleteNamespace

必要許可 (API 動作)：

- `servicediscovery:DeleteNamespace`

DeleteService

所需許可 (API 動作)：`servicediscovery:DeleteService`

DeleteServiceAttributes

所需許可 (API 動作)：`servicediscovery:DeleteServiceAttributes`

DeregisterInstance

必要許可 (API 動作)：

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

DiscoverInstances

所需許可 (API 動作)：`servicediscovery:DiscoverInstances`

GetInstance

所需許可 (API 動作)：`servicediscovery:GetInstance`

GetInstancesHealthStatus

所需許可 (API 動作)：`servicediscovery:GetInstancesHealthStatus`

GetNamespace

所需許可 (API 動作)：`servicediscovery:GetNamespace`

GetOperation

所需許可 (API 動作)：`servicediscovery:GetOperation`

GetService

所需許可 (API 動作)：`servicediscovery:GetService`

GetServiceAttributes

所需許可 (API 動作)：`servicediscovery:GetServiceAttributes`

ListInstances

所需許可 (API 動作)：`servicediscovery:ListInstances`

ListNamespaces

所需許可 (API 動作)：`servicediscovery:ListNamespaces`

ListOperations

所需許可 (API 動作)：`servicediscovery:ListOperations`

ListServices

所需許可 (API 動作)：`servicediscovery:ListServices`

ListTagsForResource

所需許可 (API 動作)：`servicediscovery:ListTagsForResource`

RegisterInstance

必要許可 (API 動作)：

- `servicediscovery:RegisterInstance`
- `route53:GetHealthCheck`
- `route53:CreateHealthCheck`
- `route53:UpdateHealthCheck`
- `ec2:DescribeInstances`

TagResource

所需許可 (API 動作)：`servicediscovery:TagResource`

UntagResource

所需許可 (API 動作)：`servicediscovery:UntagResource`

UpdateHttpNamespace

所需許可 (API 動作)：`servicediscovery:UpdateHttpNamespace`

UpdateInstanceCustomHealthStatus

所需許可 (API 動作)：`servicediscovery:UpdateInstanceCustomHealthStatus`

UpdatePrivateDnsNamespace

必要許可 (API 動作)：

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

UpdatePublicDnsNamespace

必要許可 (API 動作)：

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

UpdateService

必要許可 (API 動作)：

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

UpdateServiceAttributes

所需許可 (API 動作)：`servicediscovery:UpdateServiceAttributes`

對 AWS Cloud Map 身分和存取進行故障診斷

使用以下資訊來協助您診斷和修正使用 AWS Cloud Map 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 中執行動作 AWS Cloud Map
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 以外的人員 AWS 帳戶 存取我的 AWS Cloud Map 資源

我無權在 中執行動作 AWS Cloud Map

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在`mateojackson` IAM 使用者嘗試使用主控台檢視一個虛構 *`my-example-widget`* 資源的詳細資訊，但卻無虛構 `servicediscovery:`*`GetWidget`* 許可時發生。

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

在此情況下，必須更新 `mateojackson` 使用者的政策，允許使用 `servicediscovery:`*`GetWidget`* 動作存取 *`my-example-widget`* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您未獲授權執行 `iam:PassRole` 動作，您的政策必須更新，允許您將角色傳遞給 AWS Cloud Map。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 `marymajor` 的 IAM 使用者嘗試使用主控台在 AWS Cloud Map中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

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

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 AWS Cloud Map 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 是否 AWS Cloud Map 支援這些功能，請參閱 AWS Cloud Map 如何使用 IAM。

- 若要了解如何提供您擁有 AWS 帳戶 的資源存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 中為 IAM 使用者提供存取權。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

的合規驗證 AWS Cloud Map

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- HIPAA 合格服務參考 – 列出 HIPAA 合格服務。並非所有 AWS 服務 都符合 HIPAA 資格。
- AWS 合規資源 – 此工作手冊和指南集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同責任模型。本指南摘要說明跨多個架構 （包括國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 保護 AWS 服務 和映射指南至安全控制的最佳實務。
- 《 AWS Config 開發人員指南》中的使用 規則評估資源 – AWS Config 服務會評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub – 這 AWS 服務 可讓您全面檢視其中的安全狀態 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱「Security Hub 控制參考」。
- Amazon GuardDuty – 這會監控您的環境是否有可疑和惡意活動，以 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。
- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險和符合法規和業界標準的方式。

中的彈性 AWS Cloud Map

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置。 AWS 區域提供多個實體分隔和隔離的可用區域，這些可用區域以低延遲、高輸送量和高度備援的聯網連接。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

AWS Cloud Map 主要是全球服務。不過，您可以使用 AWS Cloud Map 建立 Route 53 運作狀態檢查，以檢查特定區域中資源的運作狀態，例如 Amazon EC2 執行個體和 Elastic Load Balancing 負載平衡器。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

中的基礎設施安全 AWS Cloud Map

作為受管服務， AWS Cloud Map 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務設計您的 AWS 環境，請參閱 Security Pillar AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 已發佈的 API 呼叫， AWS Cloud Map 透過網路存取 。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

您可以設定 AWS Cloud Map 以使用介面 VPC 端點，藉此改善 VPC 的安全狀態。如需詳細資訊，請參閱AWS Cloud Map 使用介面端點 (AWS PrivateLink) 存取。

AWS Cloud Map 使用介面端點 (AWS PrivateLink) 存取

您可以使用 在 VPC 與 之間 AWS PrivateLink 建立私有連線 AWS Cloud Map。您可以 AWS Cloud Map 像在 VPC 中一樣存取 ，無需使用網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址即可存取 AWS Cloud Map。

您可以建立由 AWS PrivateLink提供支援的介面端點來建立此私有連線。我們會在您為介面端點啟用的每個子網中建立端點網路介面。這些是請求者管理的網路介面，可作為目的地為 AWS Cloud Map之流量的進入點。

如需詳細資訊，請參閱「AWS PrivateLink 指南」中的透過 AWS PrivateLink存取 AWS 服務。

的考量事項 AWS Cloud Map

設定介面端點之前 AWS Cloud Map，請檢閱 AWS PrivateLink 指南中的考量事項。

如果您的 Amazon VPC 沒有網際網路閘道，且您的任務使用`awslogs`日誌驅動程式將日誌資訊傳送至 CloudWatch Logs，則必須為 CloudWatch Logs 建立介面 VPC 端點。如需詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的將 CloudWatch Logs 與介面 VPC 端點搭配使用。Amazon CloudWatch

VPC 端點不支援 AWS 跨區域請求。請確實在計劃發出 AWS Cloud Map API 呼叫的相同區域中建立端點。

透過 Amazon Route 53，VPC 端點僅支援 Amazon 提供的 DNS。如果您想要使用自己的 DNS，您可以使用條件式 DNS 轉送。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 DHCP 選項集。

連接至 VPC 端點的安全群組必須允許連接埠 443 上來自 Amazon VPC 私有子網路的傳入連線。

建立 的介面端點 AWS Cloud Map

您可以使用 Amazon VPC AWS Cloud Map 主控台或 AWS Command Line Interface () 建立 的介面端點AWS CLI。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的建立介面端點。

AWS Cloud Map 使用下列服務名稱建立 的介面端點：

> **Note**
>
> `DiscoverInstances` API 將無法在這兩個端點上使用。

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

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

使用下列服務名稱，為 AWS Cloud Map 資料平面建立存取 `DiscoverInstances` API 的介面端點：

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

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

> **Note**
>
> 當您`DiscoverInstances`使用資料平面端點的區域或區域 VPCE DNS 名稱呼叫 時，您將需要停用主機字首注入。當您呼叫每個 API 操作時， AWS CLI 和 AWS SDKs會在服務端點前面加上各種主機字首，這會在您指定 VPC 端點時產生無效的 URL。

如果您為介面端點啟用私有 DNS，您可以使用 AWS Cloud Map 其預設的區域 DNS 名稱向 提出 API 請求。例如：`servicediscovery.us-east-1.amazonaws.com`。

支援 的任何區域中 AWS Cloud Map 都支援 VPCE AWS PrivateLink 連線；不過，客戶需要在定義端點之前檢查哪些可用區域支援 VPCE。若要了解區域中介面 VPC 端點支援哪些可用區域，請使用 describe-vpc-endpoint-services 命令或使用 AWS Management Console。例如，下列命令會傳回可用區域，您可以在美國東部（俄亥俄）區域內部署 AWS Cloud Map 介面 VPC 端點：

```
aws --region us-east-2 ec2 describe-vpc-endpoint-services --query 'ServiceDetails[?
ServiceName==`com.amazonaws.us-east-2.servicediscovery`].AvailabilityZones[]'
```

監控 AWS Cloud Map

監控是維護您 AWS 解決方案之可靠性、可用性和效能的重要部分。您應該從 AWS 解決方案的所有部分收集監控資料，以便在發生多點失敗時更輕鬆地偵錯。不過，在開始監控之前，您應該建立監控計劃，在其中回答下列問題：

- 監控目標是什麼？
- 要監控哪些資源？
- 監控這些資源的頻率為何？
- 要使用哪些監控工具？
- 誰將執行監控任務？
- 發生問題時應該通知誰？

主題

- 使用 記錄 AWS Cloud Map API 呼叫 AWS CloudTrail

使用 記錄 AWS Cloud Map API 呼叫 AWS CloudTrail

AWS Cloud Map 已與 整合AWS CloudTrail，此服務提供使用者、角色或 所採取動作的記錄 AWS 服務。CloudTrail 會將 的所有 API 呼叫擷取 AWS Cloud Map 為事件。擷取的呼叫包括來自 AWS Cloud Map 主控台的呼叫，以及對 AWS Cloud Map API 操作的程式碼呼叫。使用 CloudTrail 收集的資訊，您可以判斷提出的請求 AWS Cloud Map、提出請求的 IP 地址、提出時間，以及其他詳細資訊。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是使用根使用者還是使用者憑證提出。
- 請求是否代表 IAM Identity Center 使用者提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務服務提出。

當您建立帳戶 AWS 帳戶 時CloudTrail 會在 中處於作用中狀態，而且您會自動存取 CloudTrail 事件歷史記錄。CloudTrail 事件歷史記錄為 AWS 區域中過去 90 天記錄的管理事件，提供可檢視、可搜尋、可下載且不可變的記錄。如需詳細資訊，請參閱「AWS CloudTrail 使用者指南」中的使用 CloudTrail 事件歷史記錄。檢視事件歷史記錄不會產生 CloudTrail 費用。

如需 AWS 帳戶 過去 90 天內持續記錄的事件，請建立線索或 CloudTrail Lake 事件資料存放區。

CloudTrail 追蹤

線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。使用 建立的所有線索 AWS Management Console 都是多區域。您可以使用 AWS CLI建立單一或多區域追蹤。建議您建立多區域追蹤，因為您擷取 AWS 區域 帳戶中所有 的活動。如果您建立單一區域追蹤，您只能檢視追蹤 AWS 區域中記錄的事件。如需追蹤的詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的為您的 AWS 帳戶建立追蹤和為組織建立追蹤。

您可以透過建立追蹤，免費將持續管理事件的一個複本從 CloudTrail 傳遞至您的 Amazon S3 儲存貯體，但這樣做會產生 Amazon S3 儲存費用。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。如需 Amazon S3 定價的相關資訊，請參閱 Amazon S3 定價。

CloudTrail Lake 事件資料存放區

CloudTrail Lake 讓您能夠對事件執行 SQL 型查詢。CloudTrail Lake 會將分列式 JSON 格式的現有事件轉換為 Apache ORC 格式。ORC 是一種單欄式儲存格式，針對快速擷取資料進行了最佳化。系統會將事件彙總到事件資料存放區中，事件資料存放區是事件的不可變集合，其依據為您透過套用進階事件選取器選取的條件。套用於事件資料存放區的選取器控制哪些事件持續存在並可供您查詢。如需 CloudTrail Lake 的詳細資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 AWS CloudTrail Lake。

CloudTrail Lake 事件資料存放區和查詢會產生費用。建立事件資料存放區時，您可以選擇要用於事件資料存放區的定價選項。此定價選項將決定擷取和儲存事件的成本，以及事件資料存放區的預設和最長保留期。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。

AWS Cloud Map CloudTrail 中的資料事件

資料事件提供有關在資源上執行或在資源中執行的資源操作的資訊（例如，在命名空間中探索已註冊的執行個體）。這些也稱為資料平面操作。資料事件通常是大量資料的活動。根據預設，CloudTrail 不會記錄資料事件。CloudTrail 事件歷史記錄不會記錄資料事件。

資料事件需支付額外的費用。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。

您可以使用 CloudTrail 主控台或 CloudTrail API 操作 AWS CLI來記錄 AWS Cloud Map 資源類型的資料事件。如需如何記錄資料事件的詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的使用 AWS Management Console記錄資料事件和使用 AWS Command Line Interface記錄資料事件。

下表列出您可以記錄資料事件 AWS Cloud Map 的資源類型。資料事件類型 (主控台) 資料行會顯示從 CloudTrail 主控台上的資料事件類型清單中選擇的值。resources.type 值欄會顯示值，您會在使用

AWS CLI 或 CloudTrail APIs 設定進階事件選取器時指定該`resources.type`值。記錄到 CloudTrail 的資料 API 資料行會針對資源類型顯示記錄到 CloudTrail 的 API 呼叫。

資料事件類型 (主控台)	resources.type 值	記錄到 CloudTrail 的資料 API
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances • DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • DiscoverInstancesRevision

您可以設定進階事件選取器來篩選 `eventName`、`readOnly` 和 `resources.ARN` 欄位，以僅記錄對您重要的事件。如需這些欄位的詳細資訊，請參閱AWS CloudTrail API 參考中的AdvancedFieldSelector。

下列範例示範如何設定進階事件選取器來記錄所有資料 AWS Cloud Map 事件。

```
"AdvancedEventSelectors":
[
    {
        "Name": "Log all AWS Cloud Map data events",
        "FieldSelectors": [
            { "Field": "eventCategory", "Equals": ["Data"] },
            { "Field": "resources.type", "Equals":
 ["AWS::ServiceDiscovery::Namespace"] }
        ]
    }
]
```

AWS Cloud Map CloudTrail 中的管理事件

管理事件提供有關在 資源上執行的管理操作的資訊 AWS 帳戶。這些也稱為控制平面操作。根據預設，CloudTrail 記錄管理事件。

AWS Cloud Map 會將所有 AWS Cloud Map 控制平面操作記錄為管理事件。如需 AWS Cloud Map 記錄到 CloudTrail 的 AWS Cloud Map 控制平面操作清單，請參閱 AWS Cloud Map API 參考。

AWS Cloud Map 事件範例

一個事件代表任何來源提出的單一請求，並包含請求 API 操作的相關資訊、操作的日期和時間、請求參數等。CloudTrail 日誌檔案不是公有 API 呼叫的已排序堆疊追蹤，因此事件不會以任何特定順序顯示。

下列範例顯示示範 `CreateHTTPNamespace`操作的 CloudTrail 管理事件。

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:alejandro_rosalez",
        "arn": "arn:aws:sts::111122223333:assumed-role/users/alejandro_rosalez",
        "accountId": "111122223333",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/readonly-role",
                "accountId": "111122223333",
                "userName": "alejandro_rosalez"
            },
            "attributes": {
                "creationDate": "2024-03-19T16:15:37Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-03-19T19:23:13Z",
    "eventSource": "servicediscovery.amazonaws.com",
    "eventName": "CreateHttpNamespace",
    "awsRegion": "eu-west-3",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36",
    "requestParameters": {
        "name": "example-namespace",
        "creatorRequestId": "eda8b524-ca14-4f68-a176-dc4dfd165c26",
        "tags": []
    },
    "responseElements": {
```

```
        "operationId": "7xm4i7ghhkaalma666nrg6itf2eylcbp-gwipo38o"
    },
    "requestID": "641274d0-dbbe-4e64-9b53-685769a086c7",
    "eventID": "4a1ab076-ef1b-4bcf-aa95-cec5fb64f2bd",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "servicediscovery.eu-west-3.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

下列範例顯示示範 `DiscoverInstances`操作的 CloudTrail 資料事件。

```
{
            "eventVersion": "1.09",
            "userIdentity": {
                "type": "AssumedRole",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE:alejandro_rosalez",
                "arn": "arn:aws:sts::111122223333:assumed-role/role/Admin",
                "accountId": "111122223333",
                "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "AROA123456789EXAMPLE",
                        "arn": "arn:aws:iam::"111122223333":role/Admin",
                        "accountId": "111122223333",
                        "userName": "Admin"
                    },
                    "attributes": {
                        "creationDate": "2024-03-19T16:15:37Z",
                        "mfaAuthenticated": "false"
                    }
                }
            },
            "eventTime": "2024-03-19T21:19:12Z",
            "eventSource": "servicediscovery.amazonaws.com",
```

```
            "eventName": "DiscoverInstances",
            "awsRegion": "eu-west-3",
            "sourceIPAddress": "13.38.34.79",
            "userAgent": "Boto3/1.20.34 md/Botocore#1.34.60 ua/2.0 os/linux#6.5.0-1014-
aws md/arch#x86_64 lang/python#3.10.12 md/pyimpl#CPython cfg/retry-mode#legacy
 Botocore/1.34.60",
            "requestParameters": {
                "namespaceName": "example-namespace",
                "serviceName": "example-service",
                "queryParameters": {"example-key": "example-value"}
            },
            "responseElements": null,
            "requestID": "e5ee36f1-edb0-4814-a4ba-2e8c97621c79",
            "eventID": "503cedb6-9906-4ee5-83e0-a64dde27bab0",
            "readOnly": true,
            "resources": [
                {
                    "accountId": "111122223333",
                    "type": "AWS::ServiceDiscovery::Namespace",
                    "ARN": "arn:aws:servicediscovery:eu-west-3:111122223333:namespace/
ns-vh4nbmhEXAMPLE"
                },
                {
                    "accountId": "111122223333",
                    "type": "AWS::ServiceDiscovery::Service",
                    "ARN": "arn:aws:servicediscovery:eu-west-3:111122223333:service/
srv-h46op6ylEXAMPLE"
                }
            ],
            "eventType": "AwsApiCall",
            "managementEvent": false,
            "recipientAccountId": "111122223333",
            "eventCategory": "Data",
            "tlsDetails": {
                "tlsVersion": "TLSv1.3",
                "cipherSuite": "TLS_AES_128_GCM_SHA256",
                "clientProvidedHostHeader": "data-servicediscovery.eu-
west-3.amazonaws.com"
            },
            "sessionCredentialFromConsole": "true"
        }
```

如需有關 CloudTrail 記錄內容的資訊，請參閱《AWS CloudTrail 使用者指南》中的 CloudTrail record contents。

標記您的 AWS Cloud Map 資源

標籤是您指派給 AWS 資源的標籤。每個標籤皆包含由您定義的一個金鑰與一個選用值。

標籤可讓您依用途、擁有者或環境等方式將 AWS 資源分類。當您有許多相同類型的資源時，您可以依據先前指派的標籤，快速識別特定的資源。例如，您可以為您的 AWS Cloud Map 服務定義一組標籤，以協助您追蹤每個服務的擁有者和堆疊層級。建議您為每個資源類型設計一組一致的標籤金鑰。

標籤不會自動指派給您的資源。新增標籤後，您可以隨時編輯標籤索引鍵和值，或從資源移除標籤。如果您刪除資源，也會刪除任何該資源的標籤。

標籤對 沒有任何語意意義， AWS Cloud Map 並嚴格解譯為字元字串。您可以將標籤的值設為空白字串，但您無法將標籤的值設為 Null。若您將與現有標籤具有相同鍵的標籤新增到該資源，則新值會覆寫舊值。

您可以使用 AWS Management Console、 AWS CLI和 AWS Cloud Map API 來使用標籤。

如果您使用的是 AWS Identity and Access Management (IAM)，您可以控制 AWS 帳戶中哪些使用者具有建立、編輯或刪除標籤的許可。

如何標記資源

您可以標記新的或現有的 AWS Cloud Map 命名空間和服務。

如果您使用的是 AWS Cloud Map 主控台，您可以隨時使用相關資源頁面上的標籤索引標籤，將標籤套用至新資源或現有資源。

如果您使用的是 AWS Cloud Map API AWS CLI、 或 AWS SDK，您可以使用相關 API 動作上的 `tags` 參數將標籤套用至新資源，或使用 TagResource API 動作將標籤套用至現有資源。如需詳細資訊，請參閱 TagResource。

有些資源建立動作可讓您在建立資源時指定資源的標籤。如果無法在資源建立時套用標籤，則資源建立程序會失敗。這可確保您要在建立時標記的資源是以指定的標籤建立，不然就根本不會建立。如果您在建立時標記資源，則不需要在建立資源之後執行自訂標記指令碼。

下表說明可標記 AWS Cloud Map 的資源，以及在建立時可標記的資源。

資源的標記支援 AWS Cloud Map

資源	支援標籤	支援標籤傳播	支援建立時標記 (AWS Cloud Map API、AWS CLI AWS SDK)
AWS Cloud Map 命名空間	是	否。命名空間標籤不會傳播到與命名空間相關聯的任何其他資源。	是
AWS Cloud Map 服務	是	否。服務標籤不會傳播到與服務相關聯的任何其他資源。	是

限制

以下基本限制適用於標籤：

- 每個資源的標籤數量上限 – 50
- 對於每一個資源，每個標籤金鑰必須是唯一的，且每個標籤金鑰只能有一個值。
- 索引鍵長度上限 - 128 個 UTF-8 Unicode 字元
- 值的長度上限 - 256 個 UTF-8 Unicode 字元
- 如果您的標記結構描述用於多個 AWS 服務和資源，請記住，其他 服務可能有允許的字元限制。通常允許的字元包括：可用 UTF-8 表示的英文字母、數字和空格，還有以下字元：+ - = . _ : / @。
- 標籤鍵與值皆區分大小寫。
- 請勿使用 `aws:`、 `AWS:`或任何大寫或小寫的組合，例如索引鍵或值的字首，因為其保留供 AWS 使用。您不可編輯或刪除具此字首的標籤金鑰或值。具此字首的標籤不算在每一資源的標籤數限制內。

更新 AWS Cloud Map 資源的標籤

使用以下 AWS CLI 命令或 AWS Cloud Map API 操作來新增、更新、列出和刪除資源的標籤。

資源的標記支援 AWS Cloud Map

任務	API 動作	AWS CLI	AWS Tools for Windows PowerShell
新增或覆寫一或多個標籤。	TagResource	tag-resource	Add-SDResourceTag
刪除一或多個標籤。	UntagResource	untag-resource	Remove-SDResourceTag
列出資源的標籤	ListTagsForResource	list-tags-for-resource	Get-SDResourceTag

下列範例示範如何使用 AWS CLI來標記或取消標記資源。

範例 1：標記現有資源

以下命令會標記現有的資源。

```
aws servicediscovery tag-resource --resource-arn resource_ARN --tags team=devs
```

範例 2：取消標記現有的資源

以下命令會從現有的資源刪除標籤。

```
aws servicediscovery untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

範例 3：列出資源的標籤

以下命令列出與現有資源相關聯的標籤。

```
aws servicediscovery list-tags-for-resource --resource-arn resource_ARN
```

有些資源建立動作可讓您在建立資源時指定標籤。下列動作支援在建立時新增標籤。

任務	API 動作	AWS CLI	AWS Tools for Windows PowerShell
建立 HTTP 命名空間	CreateHttpNamespace	create-http-namespace	New-SDHttpNamespace
根據 DNS 建立私有命名空間	CreatePrivateDnsNamespace	create-private-dns-namespace	New-SDPrivateDnsNamespace
根據 DNS 建立公用命名空間	CreatePublicDnsNamespace	create-public-dns-namespace	New-SDPublicDnsNamespace
建立服務	CreateService	create-service	New-SDService

AWS Cloud Map 服務配額

AWS Cloud Map 資源受限於下列帳戶層級服務配額。列出的每個配額適用於您建立 AWS Cloud Map 資源的每個 AWS 區域。

名稱	預設	可調整	描述
每個執行個體的自訂屬性	每個受支援的區域：30	否	您可以在註冊執行個體時指定的自訂屬性數量上限。
每個帳戶爆量率的 DiscoverInstances 操作	每個受支援的區域：2,000	是	從單一帳戶呼叫 DiscoverInstances 操作的最大爆量速率。
每個帳戶穩定速率的 DiscoverInstances 操作	每個受支援的區域：1,000	是	從單一帳戶呼叫 DiscoverInstances 操作的最大穩定速率。
每個帳戶速率的 DiscoverInstancesRevision 操作	每個受支援的區域：3,000	是	從單一帳戶呼叫 DiscoverInstancesRevision 操作的最大速率。
每個命名空間的執行個體數	每個受支援的區域：2,000	是	您可以使用相同命名空間註冊的 服務執行個體數量上限。
每個服務的執行個體數	每個支援的區域：1,000	否	您可以使用相同服務在區域中註冊的執行個體數量上限。
每個區域的命名空間	每個受支援的區域：50	是	您可以為每個區域建立的命名空間數量上限。

* 當您建立命名空間時，我們會自動建立 Amazon Route 53 託管區域。此託管區域會計入您可以使用 AWS 帳戶建立的託管區域數量上的配額。如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的託管區域的配額。

** 增加 的 DNS AWS Cloud Map 命名空間執行個體需要增加每個託管區域 Route 53 限制的記錄，這會產生額外費用。

管理您的 AWS Cloud Map 服務配額

AWS Cloud Map 已與 Service Quotas 整合，此服務 AWS 可讓您從中央位置檢視和管理配額。如需詳細資訊，請參閱《Service Quotas 使用者指南》中的「什麼是 Service Quotas？」。

Service Quotas 可讓您輕鬆地查詢 AWS Cloud Map 服務配額的值。

AWS Management Console

使用 檢視 AWS Cloud Map 服務配額 AWS Management Console

1. 開啟 Service Quotas 主控台，網址為 https://console.aws.amazon.com/servicequotas/。
2. 在導覽窗格中，選擇 AWS services (AWS 服務)。
3. 從 AWS services (AWS 服務) 清單中，搜尋並選取 AWS Cloud Map。
4. 在 的服務配額清單中 AWS Cloud Map，您可以看到服務配額名稱、套用的值（如果有的話）、 AWS 預設配額，以及配額值是否可以調整。

   若要檢視有關服務配額的其他資訊，例如描述，請選擇配額名稱以調出配額詳細資訊。
5. （選用）若要請求增加配額，請選取您要增加的配額，然後在帳戶層級選擇請求增加。

若要使用 使用服務配額， AWS Management Console 請參閱 Service Quotas 使用者指南。

AWS CLI

使用 檢視 AWS Cloud Map 服務配額 AWS CLI

執行下列命令以檢視預設 AWS Cloud Map 配額。

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code AWSCloudMap \
    --output table
```

執行下列命令以檢視您套用的 AWS Cloud Map 配額。

```
aws service-quotas list-service-quotas \
    --service-code AWSCloudMap
```

如需使用 使用服務配額的詳細資訊 AWS CLI，請參閱 Service Quotas AWS CLI 命令參考。若要請求提升配額，請參閱 AWS CLI 命令參考中的 `request-service-quota-increase` 命令。

Handle AWS Cloud Map DiscoverInstances API 請求限流

AWS Cloud Map 依區域調節每個 AWS 帳戶的 DiscoverInstances API 請求。調節有助於改善服務的效能，並有助於為所有 AWS Cloud Map 客戶提供公平的使用。調節可確保對 AWS Cloud Map DiscoverInstances API 的呼叫不超過允許的最大 DiscoverInstances API 請求配額。來自下列任何來源的 DiscoverInstances API 呼叫受請求配額限制：

- 第三方應用程式
- 命令列工具
- AWS Cloud Map 主控台

如果您超過 API 限流配額，會收到`RequestLimitExceeded`錯誤碼。如需詳細資訊，請參閱the section called “請求率限制”。

如何套用調節

AWS Cloud Map 使用權杖儲存貯體演算法來實作 API 限流。透過此演算法，您的帳戶具有儲存貯體，可存放特定數量的字符。儲存貯體中的字符數量代表您在任何指定秒的限流配額。單一 區域有一個儲存貯體，且適用於該區域中的所有端點。

請求率限制

調節會限制您可以提出的 DiscoverInstances API 請求數量。每個請求都會從儲存貯體移除一個權杖。例如，DiscoverInstances API 操作的儲存貯體大小為 2，000 個字符，因此您可以在一秒內提出最多 2，000 個 DiscoverInstances 請求。如果您在一秒內超過 2，000 個請求，系統會調節您，而該秒內剩餘的請求會失敗。

儲存貯體會自動以設定的速率重新填充。如果儲存貯體沒有容量，每秒會新增一組字符數量，直到儲存貯體達到容量為止。如果補充字符到達時儲存貯體容量為容量，則會捨棄這些字

符。DiscoverInstances API 操作的儲存貯體大小為 2，000 個字符，補充速率為每秒 1，000 個字符。如果您在一秒內提出 2，000 個 DiscoverInstances API 請求，儲存貯體會立即減少為零 (0) 個字符。然後，儲存貯體每秒最多可重新填充 1，000 個字符，直到達到其 2，000 個字符的最大容量為止。

您可以在權杖新增至儲存貯體時使用權杖。在提出 API 請求之前，您不需要等待儲存貯體達到最大容量。如果您在一秒內提出 2，000 個 DiscoverInstances API 請求以耗盡儲存貯體，您仍然可以在之後視需要每秒提出最多 1，000 個 DiscoverInstances API 請求。這表示您可以在補充字符新增至儲存貯體時立即使用它們。儲存貯體只會在您每秒提出比重新填充率更少的 API 請求時，開始重新填充至最大容量。

重試或批次處理

如果 API 請求失敗，您的應用程式可能需要重試請求。若要減少 API 請求的數量，請在連續請求之間使用適當的睡眠間隔。為了獲得最佳結果，請使用較長或可變的休眠間隔。

計算休眠間隔

當您需要輪詢或重試 API 請求時，建議您使用指數退避演算法來計算 API 呼叫之間的休眠間隔。透過在連續錯誤回應的重試之間使用逐步較長的等待時間，您可以減少失敗的請求數量。如需此演算法的詳細資訊和實作範例，請參閱 AWS SDKs和工具參考指南中的重試行為。

調整 API 限流配額

您可以請求提高 AWS 帳戶的 API 限流配額。若要請求調節配額，請聯絡 AWS 支援 中心。

的文件歷史記錄 AWS Cloud Map

下表說明 AWS Cloud Map 開發人員指南的主要更新和新功能。我們也會經常更新文件，以處理您傳送給我們的意見回饋。

變更	描述	日期
AWS Cloud Map 服務屬性	您現在可以在服務層級指定屬性，以避免在已註冊服務的執行個體之間複製屬性。您可以使用這些屬性進行複雜的流量路由、設定逾時和重試值，以及協調服務和外部整合。	2024 年 12 月 13 日
已新增教學課程	兩個教學課程顯示使用 AWS Cloud Map 新增的常見使用案例。	2024 年 3 月 27 日
已更新 CloudTrail 整合文件	描述與 CloudTrail AWS Cloud Map 整合以記錄 API 活動的文件已更新。	2024 年 3 月 20 日
受管政策更新	`AWSCloudMapDiscoverInstanceAccess` 、`AWSCloudMapRegisterInstanceAccess` 和 `AWSCloudMapReadOnlyAccess` 政策已更新。	2023 年 9 月 20 日
雲端地圖和 AWS PrivateLink	您現在可以使用 AWS PrivateLink 在 VPC 和 之間建立私有連線 AWS Cloud Map。	2023 年 9 月 15 日
受管政策更新	`AWSCloudMapDiscoverInstanceAccess` 政策已更新。	2023 年 8 月 15 日
AWS 適用於 Python 的 SDK	新增 Python 命令列範例。	2022 年 9 月 13 日

IPv6 支援	API 端點現在可在IPv6僅限 的網路中使用。	2022 年 1 月 28 日
服務執行個體探索	AWS Cloud Map 新增了在命名空間中建立服務的支援，該命名空間支援 DNS 查詢，這些查詢只能使用 DiscoverInstances API 操作探索，而無法使用 DNS 查詢。	2021 年 3 月 24 日
資源標記	AWS Cloud Map 已新增支援，可使用 將中繼資料標籤新增至您的命名空間和服務 AWS Management Console。	2021 年 2 月 8 日
資源標記	AWS Cloud Map 已新增支援，可讓您使用 和 AWS CLI APIs 將中繼資料標籤新增至命名空間和服務。	2020 年 6 月 22 日
初始版本	這是AWS Cloud Map 開發人員指南的第一個版本。	2018 年 11 月 28 日

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。