aws

使用者指南

Amazon Verified Permissions

Amazon Verified Permissions: 使用者指南

Table of Contents

什麼是 Amazon 驗證許可？ 1
已驗證權限中的授權 1
雪松政策語言 1
已驗證權限的好處 2
加速應用程式開 2
更安全的應用 2
使用者功能 2
相關服務 2
存取已驗證權限 2
已驗證權限的定價 4
術語和概念 5
授權模式 6
授權請求 6
授權回應 6
考慮政策 6
上下文資料 6
決定原則 6
實體資料 7
權限、授權和主體 7
政策執行 7
政策存放區 7
滿意政策 7
與雪松的差異 8
名稱區定義 8
原則範本支援 8
架構支援 8
擴展類型支持 8
雪松實體的 JSON 格式 9
動作群組定義 9
長度和尺寸限制 9
開始使用 11
註冊一個 AWS 帳戶 11
建立具有管理權限的使用者 11
IAM 已驗證權限的原則 13

建立第一個原則存放區 14
建立範例原則存放區 14
為範例原則存放區建立範本連結原則 15
測試範例原則存放區 16
建立 API 連結的原則存放區 18
政策商店 20
建立原則存放區 20
API 連結的保單商店 26
運作方式 28
添加 ABAC 29
考量事項 30
故障診斷 33
切換政策存放區 36
刪除原則存放區 36
原則儲存區綱要 38
編輯資料架構-視覺 40
編輯結構定義 41
刪除結構描述 42
原則驗證模式 43
政策 45
實體格式 45
建立靜態政策 50
編輯靜態策略 51
檢視政策 53
政策範例 56
允許存取個別實體 56
允許存取實體群組 56
允許存取任何實體 57
允許存取實體的屬性 (ABAC) 58
拒絕存取 61
政策範本 63
建立策略範本 63
建立範本連結原則 64
編輯策略範本 66
範例原則存放區的範例範本連結原則 67
PhotoFlash範本連結政策範例 67

DigitalPetStore 69
TinyToDo 範本連結政策範例 69
身分提供者 71
使用 Amazon Cognito 身分來源 71
使用 OIDC 身分識別來源 73
用戶端和受眾驗證 74
JWT 的用戶端授權 75
建立識別來源 77
Amazon Cognito 身份來源 78
OIDC 身分識別來源 80
編輯識別來源 83
Amazon Cognito 用者集區身分來源 83
OpenID Connect (OIDC) 身分識別來源 85
身分識別來源綱要和原則 86
有關綱要對應的注意事項 87
映射 ID 令牌 90
映射訪問令牌 94
Amazon Cognito 冒號分隔聲明的替代符號 99
設計授權模型 101
沒有單一正確的模型 102
專注於資源 102
複合授權 103
考慮多租戶 104
比較共用原則存放區和每個租用戶原則存放 105
如何選擇 106
填入原則範圍 106
將所有資源放入容器 107
主體與主體分離 108
不要在屬性中嵌入權限 111
精細存取 113
查詢授權的其他理由 113
測試台 115
授權 118
API 操作 118
API 測試 119
整合應用程式 121

... 124
評估示例上下文 ... 126
安全性 ... 132
資料保護 ... 132
資料加密 ... 133
身分與存取管理 ... 134
物件 ... 134
使用身分驗證 ... 134
使用政策管理存取權 ... 137
Amazon 驗證許可如何與 IAM ... 139
身分型政策範例 ... 144
故障診斷 ... 147
法規遵循驗證 ... 148
復原能力 ... 149
監控 ... 150
CloudTrail 日誌 ... 150
已驗證的權限資訊 CloudTrail ... 150
瞭解已驗證的權限記錄檔項目 ... 151
AWS CloudFormation 資源 ... 169
已驗證的權限和 AWS CloudFormation 範本 ... 169
AWS CDK 構建 ... 169
進一步了解 AWS CloudFormation ... 170
AWS PrivateLink ... 171
考量事項 ... 171
建立介面端點 ... 171
配額 ... 172
資源配額 ... 172
階層配額 ... 173
每秒作業的配額 ... 174
文件歷史紀錄 ... 177
... clxxviii

什麼是 Amazon 驗證許可？

Amazon 驗證許可是可擴展、精細的許可管理和授權服務，適用於您建置的自訂應用程式。驗證權限可讓您的開發人員透過外部化授權並集中原則管理和管理，以更快速地建置安全的應用程式。已驗證的權限會使用 Cedar 原則語言，為應用程式使用者定義精細的權限。

主題

- 已驗證權限中的授權
- 雪松政策語言
- 已驗證權限的好處
- 相關服務
- 存取已驗證權限
- 已驗證權限的定價

已驗證權限中的授權

「已驗證權限」可透過驗證主參與者是否允許對自訂應用程式中指定前後關聯中的資源執行動作，以提供授權。已驗證的許可假設主體先前已透過其他方式 (例如使用 OpenID Connect、託管提供者 (例如 Amazon Cognito) 或其他身份驗證解決方案的通訊協定進行識別和驗證。「已驗證的權限」與管理使用者的位置以及驗證使用者的方式無關。

已驗證的權限是一項服務，可讓客戶在中建立、維護和測試政策 AWS Management Console。使用 Cedar 原則語言來表示權限。用戶端應用程式會呼叫授權 API 來評估隨服務一起儲存的 Cedar 原則，並針對是否允許動作提供存取決策。

雪松政策語言

已驗證權限中的授權原則是使用 Cedar 原則語言撰寫。Cedar 是一種開源語言，用於編寫授權政策並根據這些政策做出授權決策。當您建立應用程式時，您需要確保只有授權的使用者才能存取應用程式，而且只能執行每個使用者授權執行的動作。使用 Cedar，您可以將業務邏輯與授權邏輯分離。在應用程式的程式碼中，您可以透過呼叫 Cedar 授權引擎來對作業發出的要求，詢問「此要求是否已授權？」。然後，如果決定為「允許」，則應用程序可以執行請求的操作，或者如果決定為「拒絕」，則返回錯誤消息。

驗證的權限目前使用雪松版本 2.4。

如需 Cedar 的詳細資訊，請參閱下列內容：

- 雪松政策語言參考指南
- 雪松 GitHub儲存庫

已驗證權限的好處

加速應用程式開

透過將授權與業務邏輯解耦，加速應用程式開發。

更安全的應用

驗證權限可讓開發人員建置更安全的應用程式。

使用者功能

已驗證的權限可讓您提供更豐富的使用者功能以進行權限管理

相關服務

- Amazon Cognito — Amazon Cognito 是一個適用於網絡和移動應用程序的身份平台。是一種使用者目錄、身分驗證伺服器，以及 OAuth 2.0 存取權杖和 AWS 憑證的授權服務。建立政策存放區時，您可以選擇從 Amazon Cognito 使用者集區建立主體和群組。如需詳細資訊，請參閱 Amazon Cognito 開發人員指南。
- Amazon API Gateway — Amazon API Gateway 是一種 AWS 用於建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API 的服務。建立政策存放區時，您可以選擇從 API Gateway 中的 API 建立動作和資源。如需 API Gateway 的詳細資訊，請參閱 API Gateway 開發人員指南。
- AWS IAM Identity Center— 使用 IAM 身分中心，您可以管理員工身分識別的登入安全性，也稱為勞動力使用者。IAM 身分中心提供了一個位置，您可以在其中建立或連接員工使用者，並集中管理其所有使用者 AWS 帳戶 和應用程式的存取權。如需詳細資訊，請參閱 AWS IAM Identity Center 使用者指南。

存取已驗證權限

您可以使用下列任何一種方式使用 Amazon 驗證許可。

AWS Management Console

控制台是一個基於瀏覽器的界面，用於管理已驗證的權限和 AWS 資源。如需有關透過主控台存取已驗證權限的詳細資訊，請參閱《AWS 登入 使用手冊》 AWS中的如何登入。

- Amazon 驗證許可控制

AWS 命令行工具

您可以使用命 AWS 令列工具在系統的命令列中發出指令，以執行已驗證的權限和工 AWS 作。使用命令列可以比主控台更快，也更便利。若您想要建構執行 AWS 任務的指令碼，命令列工具也非常實用。

AWS 提供兩組指令行工具：AWS Command Line Interface(AWS CLI) 和 AWS Tools for Windows PowerShell. 若要取得有關安裝和使用的資訊 AWS CLI，請參閱《使AWS Command Line Interface用指南》。若要取得有關安裝和使用 Windows 工具的資訊 PowerShell，請參閱使用AWS Tools for Windows PowerShell 者指南。

- 指令參考中已驗證的權限 AWS CLI
- Amazon 驗證許可 AWS Tools for Windows PowerShell

AWS 開發套件

AWS 提供 SDK（軟件開發工具包），其中包含各種編程語言和平台（Java，Python，紅寶石，.NET，iOS，安卓等）的示例代碼。SDK 提供了一種方便的方式來創建對已驗證權限和. AWS 例如，開發套件會負責的工作諸如以密碼演算法簽署請求、管理錯誤以及自動重試請求。

若要深入了解並下載 AWS SDK，請參閱. Amazon Web Services

以下是各種 AWS SDK 中已驗證權限資源的文件連結。

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto)
- AWS SDK for Ruby

AWS CDK 構建

這 AWS Cloud Development Kit (AWS CDK) 是一個開放原始碼軟體開發架構，用於在程式碼中定義雲端基礎架構，並透過 AWS CloudFormation. 可以使用構造或可重複使用的雲組件來創建 AWS CloudFormation 模板。然後，您可以使用這些範本來部署您的雲端基礎架構。

若要深入瞭解並下載 AWS CDK，請參閱 AWS Cloud Development Kit。

以下是「已驗證權限」 AWS CDK 資源 (例如建構) 的文件連結。

- Amazon 驗證許可 L2 CDK 構造

已驗證的權限 API

您可以使用驗證權限 API 以 AWS 程式設計方式存取已驗證的權限，該 API 可讓您直接向服務發出 HTTPS 要求。當您使用 API 時，必須包含使用您的登入資料來數位簽署請求的程式碼。

- Amazon 驗證許可 API 參考指南

已驗證權限的定價

「已驗證權限」會根據應用程式每月向「已驗證權限」發出的授權要求量，提供分層定價。政策管理動作的定價是根據您的應用程式每月對已驗證權限發出的 cURL (用戶端 URL) 原則 API 要求量而定。

如需已驗證許可的完整費用和價格清單，請參閱 Amazon 驗證許可定價。

若要查看您的帳單，請前往 AWS Billing and Cost Management 主控台中的帳單與成本管理儀表板。您的帳單內含用量報告的連結，可提供帳單的詳細資訊。若要進一步了解 AWS 帳戶 帳單，請參閱使AWS Billing 用者指南。

如果您對帳 AWS 單、帳戶和活動有任何疑問，請聯絡 AWS Support。

Amazon 驗證許可條款和概念

您應該了解以下概念才能使用 Amazon 驗證許可。

已驗證權限概念

- 授權模式
- 授權請求
- 授權回應
- 考慮政策
- 上下文資料
- 決定原則
- 實體資料
- 權限、授權和主體
- 政策執行
- 政策存放區
- 滿意政策
- 驗證權限與雪松之間的差異

雪松政策語言概念

- 授權
- 實體
- 群組與階層
- 命名空間
- 政策
- 政策範本
- 結構描述

授權模式

授權模型描述了應用程序提出的授權請求的範圍以及評估這些請求的基礎。它是根據不同類型的資源、對這些資源所採取的動作，以及採取這些動作的類型主參與者來定義。它也會考慮採取這些動作的前後關聯。

以角色為基礎的存取控制 (RBAC) 是一種評估基礎，在此基礎中定義角色並與一組權限相關聯。然後可以將這些角色指派給一或多個身分識別。指派的身分會取得與角色相關聯的權限。如果修改與角色相關聯的權限，則修改會自動影響已指派角色的任何身分識別。Cedar 可以通過使用主要組來支持 RBAC 決定。

以屬性為基礎的存取控制 (ABAC) 是評估基礎，其中與識別相關聯的權限是由該識別的屬性決定。Cedar 可以透過使用參照主體屬性的原則條件來支援 ABAC 決策。

Cedar 原則語言允許為具有屬性型條件的使用者群組定義權限，藉此在單一原則中啟用 RBAC 和 ABAC 的組合。

授權請求

授權要求是應用程式提出的「已驗證權限」要求，用來評估一組原則，以判斷主參與者是否可以針對特定前後關聯的資源執行動作。

授權回應

授權響應是對授權請求的響應。其中包括允許或拒絕決定，以及其他資訊，例如決定原則的 ID。

考慮政策

考慮的原則是評估授權要求時，由「已驗證的權限」選取以包含的完整原則集。

上下文資料

上下文數據是提供要評估的其他信息的屬性值。

決定原則

決定原則是決定授權回應的原則。例如，如果有兩個符合的原則，其中一個是拒絕，另一個是允許，則拒絕原則將是決定原則。如果有多個滿意的許可政策和沒有滿意的禁止政策，則有多個決定政策。如果沒有任何原則相符且回應為拒絕，則沒有決定原則。

實體資料

實體資料是有關主參與者、動作和資源的資料。與原則評估相關的實體資料是群組成員資格，其實體階層以及主參與者和資源的屬性值。

權限、授權和主體

已驗證的權限會在您建置的自訂應用程式中管理細微的權限和授權。

主體是指應用程式 (人工或機器) 的使用者，該應用程式具有繫結至識別碼 (例如使用者名稱或機器 ID) 的識別碼。驗證程序會決定主體是否真正是他們宣稱的身分。

與該識別相關聯的是一組應用程式權限，可決定該主體可在該應用程式內執行的動作。授權是評估這些權限的程序，以判斷主體是否允許在應用程式中執行特定動作。這些權限可以表示為策略。

政策執行

原則強制執行是在「已驗證權限」之外的應用程式內強制執行評估決策的程序。如果「已驗證權限」評估傳回拒絕，則強制執行會確保主參與者無法存取資源。

政策存放區

原則存放區是原則和範本的容器。每個存放區都包含一個結構描述，用於驗證新增至儲存區的原則。依預設，每個應用程式都有自己的原則存放區，但是多個應用程式可以共用單一原則存放區。當應用程式提出授權要求時，會識別用來評估該要求的原則存放區。原則存放區提供隔離一組原則的方法，因此可在多租用戶應用程式中使用，以包含每個承租人的結構描述和原則。單一應用程式可以為每個租用戶設定個別的原則存放區。

評估授權要求時，已驗證的權限只會考慮原則存放區中與要求相關的原則子集。相關性是根據政策範圍決定的。範圍會識別套用原則的特定主參與者和資源，以及主參與者可對資源執行的動作。定義範圍可藉由縮小一組考慮的原則來協助改善效能。

滿意政策

滿意的原則是符合授權要求參數的原則。

驗證權限與雪松之間的差異

Amazon 驗證許可使用 Cedar 政策語言引擎來執行其授權任務。不過，原生 Cedar 實作與已驗證權限中找到的 Cedar 實作之間存在一些差異。本主題識別這些差異。

名稱區定義

Cedar 的已驗證權限實作與原生 Cedar 實作有下列差異：

- 已驗證的權限僅支援一個架構中的命名空間在原則存放區中定義。
- 已驗證的權限不允許您建立命名空間具有以下值：aws,amazon，或cedar。

原則範本支援

驗證的權限和 Cedar 只允許在範圍內的佔位符principal和resource。但是，已驗證的權限也不需要principal和resource不受約束。

下列原則在 Cedar 中是有效的，但已驗證的權限會遭到拒絕，因為principal不受約束。

```
permit(principal, action == Action::"view", resource == ?resource);
```

下列兩個範例在 Cedar 和已驗證權限中都是有效的，因為這兩個principal和resource有限制。

```
permit(principal == User::"alice", action == Action::"view", resource == ?resource);
```

```
permit(principal == ?principal, action == Action::"a", resource in ?resource);
```

架構支援

已驗證的權限要求所有結構定義 JSON 金鑰名稱都是非空字串。雪松允許在少數情況下空字符串，例如屬性。

擴展類型支持

驗證的許可支持雪松延伸類型在原則中，但目前不支援將它們包含在結構描述的定義中，或作為結構描述的一部分entities的參數IsAuthorized和IsAuthorizedWithToken操作。

延伸類型包括固定點 (decimal) 和 IP 位址 (ipaddr) 資料類型。

雪松實體的 JSON 格式

此時，「已驗證的權限」要求您使用定義的結構，傳遞要在授權請求中考慮的實體清單。EntitiesDefinition，這是一個數組EntityItem元素。驗證權限目前不支持傳遞要在授權請求中考慮的實體列表雪松格式。如需格式化實體以便在已驗證權限中使用的特定需求，請參閱Amazon 驗證許可中的實體格式。

動作群組定義

Cedar 授權方法要求在評估針對原則的授權要求時，必須考慮實體清單。

您可以定義應用程式在結構描述中使用的動作和動作群組。不過，Cedar 不會將結構描述納入評估要求中。Cedar 只會使用結構描述來驗證您提交的原則和原則範本。由於 Cedar 不會在評估要求期間參考結構描述，因此即使您在結構描述中定義了動作群組，您也必須將任何動作群組清單納入您必須傳遞至授權 API 作業的實體清單中。

已驗證的權限會為您執行此操作。您在結構描述中定義的任何動作群組都會自動附加到您作為參數傳遞給`IsAuthorized`或者`IsAuthorizedWithToken`操作。

長度和尺寸限制

已驗證的權限支援原則存放區形式的儲存空間，以保存您的結構描述、原則和原則範本。該存儲會導致「驗證權限」強加一些與 Cedar 無關的長度和大小限制。

物件	已驗證的權限限制 (位元組)	雪松極限
政策規模 [1]	10,000	無
內嵌政策描述	150	不適用於雪松
策略範本大小	10,000	無
綱要大小	10,000	無
實體類型	200	無
政策 ID	64	無
策略範本識別碼	64	無

物件	已驗證的權限限制 (位元組)	雪松極限
實體 ID	200	無
原則存放區 ID	64	不適用於雪松

[1] 根據在原則存放區中建立之原則的主參與者、動作和資源的合併大小，在「已驗證權限」中每個原則存放區有一個限制。與單一資源相關的所有原則總大小不得超過 200,000 個位元組。對於範本連結的策略，原則範本的大小只會計算一次，再加上用於實體化每個範本連結策略的每組參數大小。

開始使用已驗證權限

使用此教學課程開始使用 Amazon 驗證許可。

主題

- 註冊一個 AWS 帳戶
- 建立具有管理權限的使用者
- IAM 已驗證權限的原則
- 建立您的第一個「驗證權限」原則
- 使用連線的 API 和身分識別提供者建立原則存放區

註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 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，並在電話鍵盤輸入驗證碼。

   當您註冊時 AWS 帳戶，會建立AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。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。

建立具有管理權限的使用者

註冊後，請保護 AWS 帳戶 AWS 帳戶根使用者、啟用和建立系統管理使用者 AWS IAM Identity Center，這樣您就不會將 root 使用者用於日常工作。

保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。AWS Management Console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)。

   如需指示，請參閱《使用指南》中的「IAM 為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 (主控台)」。

建立具有管理權限的使用者

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 身分中心中，將管理存取權授予使用者。

   若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。

以具有管理權限的使用者身分登入

- 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 身分中心使用者登入的說明，請參閱使用AWS 登入 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 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

   如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「建立權限集」。

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

   如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「新增群組」。

IAM 已驗證權限的原則

「已驗證的權限」會管理應用程式中使用者的權限。若要讓您的應用程式呼叫已驗證權限 API，或允許 AWS Management Console 使用者管理已驗證權限原則存放區中的 Cedar 原則，您必須新增必要的 IAM 權限。

以身分識別為基礎的原則是 JSON 權限原則文件，您可以附加至身分識別，例如 IAM 使用者、使用者群組或角色。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。若要瞭解如何建立以身分識別為基礎的策略，請參閱 IAM 使用指南中的建立 IAM 策略。

使用以 IAM 身分為基礎的策略，您可以指定允許或拒絕的動作和資源，以及允許或拒絕動作的條件 (如下所列)。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。若要瞭解可在 JSON 政策中使用的所有元素，請參閱使用 IAM 者指南中的 IAM JSON 政策元素參考資料。

Action	Description
CreatePolicyStore	建立新原則存放區的動作。
DeletePolicyStore	刪除原則存放區的動作。
ListPolicyStores	列出中所有策略存放區的動作 AWS 帳戶。
CreatePolicy	在原則存放區中建立 Cedar 原則的動作。您可以建立靜態策略或連結至策略範本的策略。
DeletePolicy	從策略存放區刪除策略的動作。
GetPolicy	擷取有關指定策略之資訊的動作。
ListPolicies	列出策略存放區中所有策略的動作。
IsAuthorized	根據授權要求中描述的參數取得授權回應的動作。

動作權限的範例 IAM 原 CreatePolicy 則：

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

```
        {
            "Effect": "Allow",
            "Action": [
                "verifiedpermissions:CreatePolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

建立您的第一個「驗證權限」原則

當您第一次登入已驗證權限主控台時，您可以選擇如何建立您的第一個原則存放區和 Cedar 原則。按照《AWS 登入使用者指南》 如何登入 AWS 主題中適合您使用者類型的登入程序操作。在主控台首頁上，選取 Amazon 驗證許可服務。選擇開始使用。

建立範例原則存放區

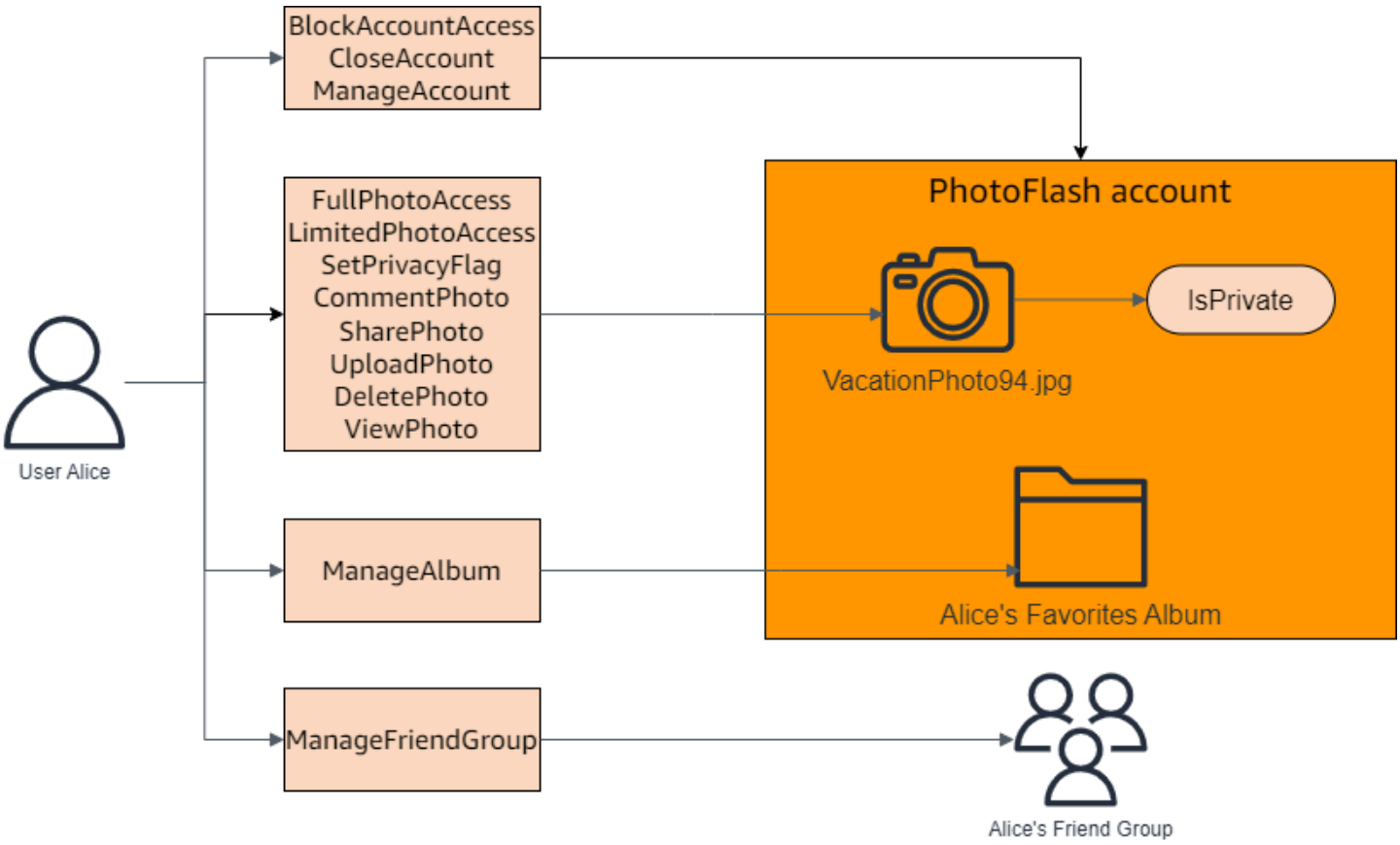
如果這是您第一次使用已驗證的權限，建議您使用其中一個範例原則存放區來熟悉已驗證權限的運作方式。範例原則儲存區提供預先定義的原則和結構描述。

使用範例原則儲存區組態方法建立原則存放區

1. 在 [已驗證的權限] 主控台中，選取 [建立新原則存放區
2. 在 [開始選項] 區段中，選擇 [範例原則存放區]。
3. 在 [範例專案] 區段中，選擇要使用的 [已驗證權限] 應用程式範例類型。對於本教學課程，請選擇PhotoFlash策略存放區。
4. 系統會根據您選擇的範例專案，自動產生範例原則存放區結構描述的命名空間。
5. 選擇建立原則存放區。

   您的原則存放區是使用原則、原則範本和範例原則存放區的結構描述來建立。

下圖說明 PhotoFlash 範例策略存放區動作與套用的資源類型之間的關係。



為範例原則存放區建立範本連結原則

範 PhotoFlash 例原則存放區包含原則、原則範本和結構描述。您可以根據範例原則存放區隨附的原則範本建立範本連結的原則。

建立範例原則存放區的範本連結原則

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 Policies (政策)。
3. 選擇 [建立原則]，然後選擇 [建立範本連結原則]。
4. 選擇包含說明的政策範本旁邊的圓形按鈕：授與非私人共用相片的完整存取權，然後選擇 [下一步]。
5. 在「主體」中，輸入`PhotoFlash::User::"Alice"`。對於「資源」，請輸入`PhotoFlash::Album::"Bob-Vacation-Album"`。
6. 選擇 [建立範本連結原則]。

   新的範本連結政策會顯示在 [原則] 下。

7. 為範例原則存放區建立另一個 PhotoFlash 範本連結原則。選擇 [建立原則]，然後選擇 [建立範本連結原則]。
8. 選擇包含說明的政策範本旁邊的圓形按鈕：授予非私人共用相片的有限存取權，然後選擇 [下一步]。
9. 在「主體」中，輸入`PhotoFlash::FriendGroup::"MySchoolFriends"`。對於「資源」，請輸入`PhotoFlash::Album::"Alice's favorite album"`。
10. 選擇 [建立範本連結原則]。

    新的範本連結政策會顯示在 [原則] 下。

我們將在教學課程的下一節中測試新的範本連結政策。如需可用來建立範本連結政策的值的更多範例 PhotoFlash，請參閱。PhotoFlash範本連結政策範例

測試範例原則存放區

建立範例原則存放區和範本連結原則之後，您可以使用「已驗證權限」測試工作台執行模擬授權要求，來測試範例「已驗證的權限」靜態原則和新的範本連結原則。

視您建立範例原則存放區的時間而定，您的原則範本可能與此程序中的參照不同。在您開始本教學課程的這一部分之前，請檢查您的範例原則存放區中是否有遵循的每個原則 PhotoFlash 範本。如果您的原則不符合這些原則，請編輯現有原則，或從 [範例專案] 選項建立新的原則存放區PhotoFlash。

授予對非私人共享照片的完整訪問權限

```
permit (
    principal in ?principal,
    action in PhotoFlash::Action::"FullPhotoAccess",
    resource in ?resource
)
when { resource.IsPrivate == false };
```

授予對非私人共享照片的有限訪問權限

```
permit (
    principal in ?principal,
    action in PhotoFlash::Action::"LimitedPhotoAccess",
    resource in ?resource
)
when { resource.IsPrivate == false };
```

若要測試範例原則存放區原則

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [測試工作台]。
3. 選擇視覺模式。
4. 在「主參與者」段落中，從綱要的主要項目類型選擇PhotoFlash:: User。在文字方塊中輸入使用者的識別碼。例如 `Alice`。
5. 請勿為主參與者選擇「新增父項」。
6. 對於「帳戶:實體」屬性，請確定已選取PhotoFlash:: 帳戶實體。輸入帳戶的識別碼。例如 `Alice-account`。
7. 在資源部分，選擇:PhotoFlash: Photo 資源類型。在文字方塊中輸入相片的識別碼。例如 `photo.jpeg`。
8. 選擇新增父系，然後選擇PhotoFlash:: 帳戶作為實體類型。針對您在使用者的「帳戶:實體」欄位中指定的相片，輸入父帳戶的相同識別碼。例如 `Alice-account`。
9. 在「動作」區段中，從有效動作清單中選擇PhotoFlash:: Action:: ViewPhoto「」。
10. 在 [其他實體] 區段中，選擇 [新增此實體] 以新增建議的客戶實體。
11. 選擇頁面頂端的 [執行授權要求]，以模擬範例原則存放區中 Cedar 原則的授權要求。測試台應顯示允許請求的決定。

下表提供您可以使用「已驗證權限」測試工作台測試的主參與者、資源和動作的其他值。此表格包含根據 PhotoFlash 範例原則存放區所包含之靜態原則的授權要求決定，以及您在上一節中建立的範本連結原則。

本金價值	主帳戶：實體值	資源值	資源父項值	Action	授權決定
PhotoFlash:: 使用者 | 愛麗絲	PhotoFlash:: 帳號 | 愛麗絲帳號	PhotoFlash:: 相片 photo.jpeg	PhotoFlash:: 戶口 | 中銀賬戶	PhotoFlash:: 動作::「」ViewPhoto	拒絕
PhotoFlash:: 使用者 | 愛麗絲	PhotoFlash:: 帳號 | 愛麗絲帳號	PhotoFlash:: 相片 photo.jpeg	PhotoFlash:: 帳號 | 愛麗絲帳號	PhotoFlash:: 動作::「」ViewPhoto	允許

本金價值	主帳戶：實體值	資源值	資源父項值	Action	授權決定
PhotoFlash::使用者 | 愛麗絲	PhotoFlash::帳號 | 愛麗絲帳號	PhotoFlash::相片集 Bob-photo.jpeg	PhotoFlash::相簿 | 鮑勃度假相簿	PhotoFlash::動作::「」ViewPhoto	允許
PhotoFlash::使用者 | 愛麗絲	PhotoFlash::帳號 | 愛麗絲帳號	PhotoFlash::相片集 Bob-photo.jpeg	PhotoFlash::相簿 | 鮑勃度假相簿	PhotoFlash::動作::「」DeletePhoto	拒絕
PhotoFlash::使用者 | 愛麗絲	PhotoFlash::帳號 | 愛麗絲帳號	PhotoFlash::照片 | Bob-photo.jpeg, IsPrivate: 布爾 | 真	PhotoFlash::相簿 | 鮑勃度假相簿	PhotoFlash::動作::「」ViewPhoto	拒絕
PhotoFlash::使用者 | 簡,: PhotoFlash: | FriendGroup MySchoolFriends	PhotoFlash::帳戶 | 一月賬戶	PhotoFlash:: 相片 photo.jpeg	PhotoFlash::相簿 | 愛麗絲最愛的專輯	PhotoFlash::動作::「」ViewPhoto	允許
PhotoFlash::使用者 | 簡,: PhotoFlash: | FriendGroup MySchoolFriends	PhotoFlash::帳戶 | 一月賬戶	PhotoFlash:: 相片 photo.jpeg	PhotoFlash::相簿 | 愛麗絲最愛的專輯	PhotoFlash::動作::「」DeletePhoto	拒絕

使用連線的 API 和身分識別提供者建立原則存放區

Amazon 驗證許可的常見使用案例是授權從應用程式用戶端到後端 API 的請求。 AWS 具有用於應用程序用戶身份驗證的服務：Amazon Cognito。 AWS 也有一個安全託管的 API 服務：Amazon API Gateway。當您將驗證權限原則存放區與這兩個原則存放區結合使用時 AWS 服務，您可以將應用程式

中的使用者集區驗證和 API 授權與一致的集中式原則集合。經過驗證的許可政策存放區內建 Amazon Cognito 使用者集區身分識別來源和 API Gateway API 的支援。

若要建立連結至現有使用者集區和 API 的政策存放區，請在建立新的政策存放區時選擇「使用 Cognito 和 API Gateway 設定」。

API 連結的原則存放區會針對授權要求自動佈建您的授權模型和資源。使用 Cognito 和 API Gateway 建立程序會產生包含使用者集區身分識別來源的政策存放區，以及將 API Gateway 連線至已驗證權限的 Lambda 授權者。一開始，您可以根據使用者的群組成員資格來授權 API 要求。例如，「已驗證的權限」只能將存取權授與身為`Directors`群組成員的使用者。

隨著應用程序的增長，您可以使用用戶屬性和 OAuth 2.0 範圍實現細粒度授權。例如，已驗證的權限只能授與存取權給在網域中具有`email`屬性的使用者`mycompany.co.uk`。

自動化 API 的授權模型之後，剩下的責任就是驗證使用者並在應用程式中產生 API 要求，以及維護原則存放區。

如需進一步了解，請參閱API 連結的保單商店。

Amazon 驗證許可政策商店

原則存放區是原則和原則範本的容器。每個原則存放區都包含一個結構描述，可用來驗證新增至原則存放區的原則。建議您為每個應用程式建立一個原則存放區，或針對多租用戶應用程式建立一個原則存放區。您必須在提出授權要求時指定原則存放區。

我們建議您在原則存放區中使用 Cedar 實體的命名空間，以防止模糊。命名空間是類型的字符串前綴，由一對冒號（::）作為分隔符分隔。驗證權限支援每個原則存放區一個命名空間 如需詳細資訊，請參閱 Cedar 政策語言參考指南中的命名空間。

主題

- 建立驗證權限原則存放區
- API 連結的保單商店
- 切換驗證權限原則存放區
- 刪除已驗證權限原則商店

建立驗證權限原則存放區

您可以使用下列方法建立原則存放區：

- 遵循引導式設定 — 在建立第一個策略之前，您將定義具有有效動作和主參與者類型的資源類型。
- 使用 API Gateway 和身分來源設定 — 使用身分識別提供者 (IdP) 登入的使用者，以及透過 Amazon API Gateway 的動作和資源實體來定義您的主要實體。如果您希望應用程式以使用者群組成員資格授權 API 要求，建議您使用此選項。
- 從範例原則存放區開始 — 選擇預先定義的範例專案原則存放區。如果您正在了解「已驗證的權限」，並且想要檢視和測試範例原則，我們建議您使用此選項。
- 建立空白原則存放區 — 您將自行定義結構描述和所有存取原則。如果您已經熟悉設定原則存放區，建議您使用此選項。

Guided setup

使用引導式設定組態方法建立原則存放區

引導式設定精靈會引導您完成建立原則存放區的第一個反覆項目的程序。您將為第一個資源類型建立結構描述、描述適用於該資源類型的動作，以及您要授與權限的主參與者類型。然後，您將建立

您的第一個原則。完成此精靈之後，您就可以新增至原則存放區、擴充結構描述以描述其他資源和主參與者類型，以及建立其他原則和範本。

1. 在 [已驗證的權限] 主控台中，選取 [建立新原則存放區
2. 在 [開始選項] 區段中，選擇 [引導式設定]。
3. 輸入「原則」存放區說明。此文字可以是任何適合您組織的文字，作為對目前政策存放區功能的易記參考，例如天氣更新。
4. 在 [詳細資料] 區段中，輸入結構描述的命名空間。
5. 選擇下一步。
6. 在 [資源類型] 視窗中，輸入資源類型的名稱。
7. (選擇性) 選擇新增屬性以新增資源屬性。鍵入屬性名稱，並為資源的每個屬性選擇一個屬性類型。選擇每個屬性是否為「必要」。驗證權限在根據結構描述驗證策略時，會使用指定的屬性值。若要移除已針對資源類型新增的屬性，請選擇屬性旁邊的 [移除]。
8. 在「動作」(Actions) 欄位中，輸入要針對指定資源類型授權的動作。若要為資源類型新增其他動作，請選擇 [新增動作]。若要移除已針對資源類型新增的動作，請選擇動作旁邊的 [移除]。
9. 在 [主參與者類型的名稱] 欄位中，輸入將針對資源類型使用指定動作之主參與者類型的名稱。
10. 選擇下一步。
11. 在「主參與者類型」視窗中，選擇主參與者類型的識別來源。

    - 如果主體的 ID 和屬性將由您的「已驗證權限」應用程式直接提供，請選擇「自訂」。選擇新增屬性以新增主參與者屬性。輸入屬性名稱，然後為每個印刷屬性選擇一個屬性類型。驗證權限在根據結構描述驗證策略時，會使用指定的屬性值。若要移除已針對印刷類型新增的屬性，請選擇屬性旁邊的「移除」(Remove)。
    - 如果將透過 Amazon Cognito 產生的 ID 或存取權杖提供主體的 ID 和屬性，請選擇 Amazon Cognito 使用者集區。選擇 Connect 使用者集區。選取AWS 區域並輸入要連線的 Amazon Cognito 使用者集區的使用者集區識別碼。選擇連線。如需詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用 Amazon 驗證許可授權。
12. 選擇下一步。
13. 在 [原則詳細資料] 區段中，為您的第一個 Cedar 原則輸入選擇性的原則說明。
14. 在 [主參與者範圍] 欄位中，選擇將從原則授與權限的主參與者。

    - 選擇特定主參與者，將保單套用至特定的主參與者。在「主參與者」中選擇允許採取動作的主參與者，然後輸入主參與者的實體識別元。
    - 選擇 [所有主參與者]，將原則套用至原則存放區中的所有主參與者。

15. 在 [資源範圍] 欄位中，選擇要授權指定主參與者執行的資源。
    - 選擇 [特定資源]，將策略套用至特定資源。在 [此策略應套用的資源] 欄位中選擇資源，然後輸入資源的實體識別碼。
    - 選擇 [所有資源]，將策略套用至原則存放區中的所有資源。
16. 在「動作範圍」欄位中，選擇要授權指定主參與者執行的動作。
    - 選擇特定動作集，將策略套用至特定動作。選取 [此策略應套用的處理行動] 欄位中的動作旁邊的核取方塊。
    - 選擇 [所有動作]，將原則套用至原則存放區中的所有動作。
17. 檢閱 [原則預覽] 區段中的原則。選擇建立原則存放區。

Set up with API Gateway and an identity source

使用設定 API Gateway 和身分識別來源組態方法建立原則存放區

API Gateway 選項使用「已驗證的權限」政策來保護 API 的安全，這些策略旨在根據使用者群組或角色做出授權決策。此選項會建立原則存放區，以使用身分識別來源群組測試授權，以及使用 Lambda 授權者提供 API。

IdP 中的用戶及其組成為您的主體（ID 令牌）或您的上下文（訪問令牌）。API Gateway API 中的方法和路徑會成為您的政策授權的動作。您的應用程式會成為資源。此工作流程的結果是，「已驗證的許可」會建立原則存放區、Lambda 函數和 API Lambda 授權者。完成此工作流程後，您必須將 Lambda 授權者指派給您的 API。

1. 在 [已驗證的權限] 主控台中，選取 [建立新原則存放區
2. 在 [開始選項] 區段中，選擇 [使用 API Gateway 和身分識別來源設定]，然後選取 [下一步]。
3. 在 [匯入資源和動作] 步驟的 [API] 下，選擇可做為原則存放區資源和動作模型的 API。
    a. 從 API 中設定的階段選擇部署階段，然後選取匯入 API。如需 API 階段的詳細資訊，請參閱 Amazon API 閘道開發人員指南中的設定 REST API 的階段。
    b. 預覽匯入資源和動作的地圖。
    c. 若要更新資源或動作，請修改您的 API 路徑或方法，然後選取匯入 API。
    d. 如果您滿意您的選擇，請選擇「下一步」。
4. 在身分識別來源中，選擇身分識別提供者類型。您可以選擇 Amazon Cognito 使用者集區或 OpenID Connect (OIDC) IdP 類型。
5. 如果您選擇 Amazon Cognito:

a. 選擇與原則存放區相 AWS 帳戶 同 AWS 區域 的使用者集區。

b. 選擇要傳遞給您要提交以進行授權的 API 的令牌類型。任何一種令牌類型都包含用戶組，這是此 API 鏈接授權模型的基礎。

c. 在「應用程式用戶端驗證」下，您可以將政策存放區的範圍限制為多租用戶使用者集區中 Amazon Cognito 應用程式用戶端的子集區。若要求使用者透過使用者集區中的一或多個指定應用程式用戶端進行驗證，請選取 [僅接受具有預期應用程式用戶端 ID 的權杖 若要接受使用者集區進行驗證的任何使用者，請選取 [不驗證應用程式用戶端 ID]。

d. 選擇下一步。

6. 如果您選擇 OIDC 提供者：

a. 在發行者網址中，輸入 OIDC 發行者的網址。例如，這是提供授權伺服器、簽署金鑰和其他提供者相關資訊的服務端點`https://auth.example.com`。您的發行者 URL 必須在上託管 OIDC 探索文件。`/.well-known/openid-configuration`

b. 在權杖類型中，選擇您希望應用程式提交以進行授權的 OIDC JWT 類型。如需詳細資訊，請參閱 在結構描述和原則中使用身分識別來源。

c. 在 Token 宣告中，選擇您要在策略存放區中設定使用者屬性的方式。這些屬性定義了您的政策可以參考的宣告。

i. 選擇「索賠」來源。

A. 要提供示例令牌，請選擇從 JWT 有效負載中提取並粘貼所選令牌類型的 JWT 的有效負載。JWT 包含標題，有效負載和簽名。您的示例 JWT 必須解碼並僅有效載入。若要剖析承載，請選取擷取。

B. 若要輸入您自己的屬性集，請選擇「手動輸入索賠」。

ii. 輸入或確認您要新增至結構描述中使用者主參與者或動作前後關聯屬性的每個 Token 宣告名稱和宣告值類型。

d. 在 [使用者和群組宣告] 中，選擇身分識別來源的 [使用者] 宣告。這通常是來自您的 ID 或訪問令牌的聲明`sub`，該 ID 或訪問令牌包含要評估的實體的唯一標識符。來自連線 OIDC IdP 的身分識別會對應至原則存放區中的使用者類型。

e. 在 [使用者和群組宣告] 中，選擇身分識別來源的群組宣告。這是一個聲明`groups`，通常來自包含用戶組列表的 ID 或訪問令牌。您的原則存放區將根據群組成員資格授權要求。

f. 在「對象驗證」或「用戶端 ID」中，輸入您希望政策存放區在授權要求中接受的用戶端 ID 或對象 URL (如果有的話)。對於存取權杖，請輸入對象聲明值，例如`https://myapp.example.com`。對於 ID 令牌，請輸入類似的客戶端 ID `1example23456789`。

g. 選擇下一步。

7. 如果您選擇 Amazon Cognito，「已驗證的許可」會查詢您的使用者集區中的群組。對於 OIDC 提供者，請手動輸入群組名稱。[將動作指派給群組] 步驟會為您的原則存放區建立原則，以允許群組成員執行動作。

   a. 選擇或新增您要包含在策略中的群組。

   b. 將動作指派給您選取的每個群組。

   c. 選擇下一步。

8. 在部署應用程式整合中，檢閱已驗證權限建立政策存放區和 Lambda 授權者所採取的步驟。

9. 當您準備好建立新資源時，請選擇 [建立和部署]。

10. 保持瀏覽器中的策略存放區狀態步驟處於開啟狀態，以透過「已驗證的權限」監視資源建立進度。

11. 經過一段時間 (通常約一小時)，或當「部署 Lambda 授權者」步驟顯示「成功」時，請設定您的授權者。

    經過驗證的許可將在您的 API 中建立 Lambda 函數和 Lambda 授權者。選擇「開啟 API」以瀏覽至您的 API。

    若要了解如何指派 Lambda 授權者，請參閱 Amazon API Gateway 開發人員指南中的使用 API 閘道 Lambda 授權器。

    a. 導航到您的 API 的授權者，並記下已驗證權限創建的授權者的名稱。

    b. 導覽至資源，然後在 API 中選取頂層方法。

    c. 選擇方法請求設置下的編輯。

    d. 將授權者設置為您之前提到的授權者姓名。

    e. 展開 HTTP 要求標頭，輸入名稱或`AUTHORIZATION`，然後選取必要。

    f. 部署 API 階段。

    g. 儲存您的變更。

12. 使用您在 [選擇身分識別來源] 步驟中選取的 Token 類型的使用者集區權杖測試您的授權者。如需有關使用者集區登入和擷取權杖的詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用者集區身份驗證流程。

13. 使用 API 請求`AUTHORIZATION`標頭中的用戶池令牌再次測試身份驗證。

14. 檢查您的新原則存放區。新增和調整原則。

Sample policy store

使用範例原則儲存區組態方法建立原則存放區

1. 在 [開始選項] 區段中，選擇 [範例原則存放區]。
2. 在 [範例專案] 區段中，選擇要使用的 [已驗證權限] 應用程式範例類型。
   - PhotoFlash是一個面向客戶的 Web 應用程序示例，使用戶可以與朋友共享單個照片和相冊。使用者可以針對允許檢視、留言和重新分享相片的人員設定精細的權限。帳戶擁有者也可以建立好友群組，並將相片整理成相簿。
   - DigitalPetStore 是一個示例應用程序，任何人都可以在其中註冊並成為客戶。客戶可以添加要出售的寵物，搜索寵物和下訂單。已添加寵物的顧客將被記錄為寵物主人。寵物主人可以更新寵物的詳細信息，上傳寵物圖片或刪除寵物清單。已下訂單的客戶會記錄為訂單擁有者。訂單擁有者可以取得訂單的詳細資訊或取消訂單。寵物店經理具有管理訪問權限。

     > **Note**
     >
     > [DigitalPet存放區] 範例原則存放區不包含原則範本。PhotoFlash和範TinyTodo例原則存放區包含原則範本。

   - TinyTodo是一個範例應用程式，可讓使用者建立 taks 和工作清單。清單擁有者可以管理和分享他們的清單，並指定誰可以檢視或編輯他們的清單。
3. 系統會根據您選擇的範例專案，自動產生範例原則存放區結構描述的命名空間。
4. 選擇建立原則存放區。

   您的原則存放區是使用您選擇的範例原則存放區的原則和結構描述來建立。如需可針對範例原則存放區建立的範本連結原則的詳細資訊，請參閱。[已驗證權限範例原則存放區的範例範本連結原則]

Empty policy store

使用空白原則儲存區組態方法建立原則存放區

1. 在 [開始選項] 區段中，選擇 [清空原則存放區]。
2. 選擇建立原則存放區。

建立的空白原則存放區不含結構描述，這表示不會驗證原則。如需更新原則存放區結構描述的詳細資訊，請參閱Amazon 驗證許可政策存儲模式。

如需為原則存放區建立原則的詳細資訊，請參閱創建 Amazon 驗證許可靜態政策和建立範本連結原則。

AWS CLI

若要使用建立空白原則存放區 AWS CLI。

您可以使用`create-policy-store`作業建立原則存放區。

> **Note**
>
> 您使用建立的原則存放區 AWS CLI 是空的。
>
> - 若要加入資料架構，請參閱Amazon 驗證許可政策存儲模式。
> - 若要新增原則，請參閱創建 Amazon 驗證許可靜態政策。
> - 若要新增策略範本，請參閱建立策略範本。

```
$ aws verifiedpermissions create-policy-store \
    --validation-settings "mode=STRICT"
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T17:41:29.103459+00:00",
    "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

AWS SDKs

您可以使用 `CreatePolicyStore` API 建立政策存放區。如需詳細資訊，請參閱 CreatePolicyAmazon 驗證許可 API 參考指南中的存放區。

API 連結的保單商店

在 Amazon 驗證許可主控台中建立新的政策存放區時，可以選擇使用 API Gateway 和身分來源設定選項。使用此選項，您可以建立 API 連結的政策存放區，這是一種授權模型，用於透過 Amazon Cognito

使用者集區或 OIDC 身分提供者 (IdP) 進行驗證的應用程式，並從 Amazon API Gateway 取得資料。若要開始使用，請參閱使用連線的 API 和身分識別提供者建立原則存放區。

主題

- 驗證權限如何授權 API 請求
- 新增以屬性為基礎的存取控制 (ABAC)
- API 連結原則存放區的考量
- 疑難排解 API 連結的原則存放區

> **Important**
>
> 您使用 [已驗證權限] 主控台中的 [使用 API Gateway 和身分識別來源] 選項建立的原則存放區不適用於立即部署至生產環境。使用您的初始原則存放區，完成授權模型，並將原則存放區資源匯出至 CloudFormation。使用 AWS Cloud Development Kit (CDK) 以程式設計方式將已驗證的許可部署到生產環境。如需詳細資訊，請參閱 轉移到生產環境 AWS CloudFormation。

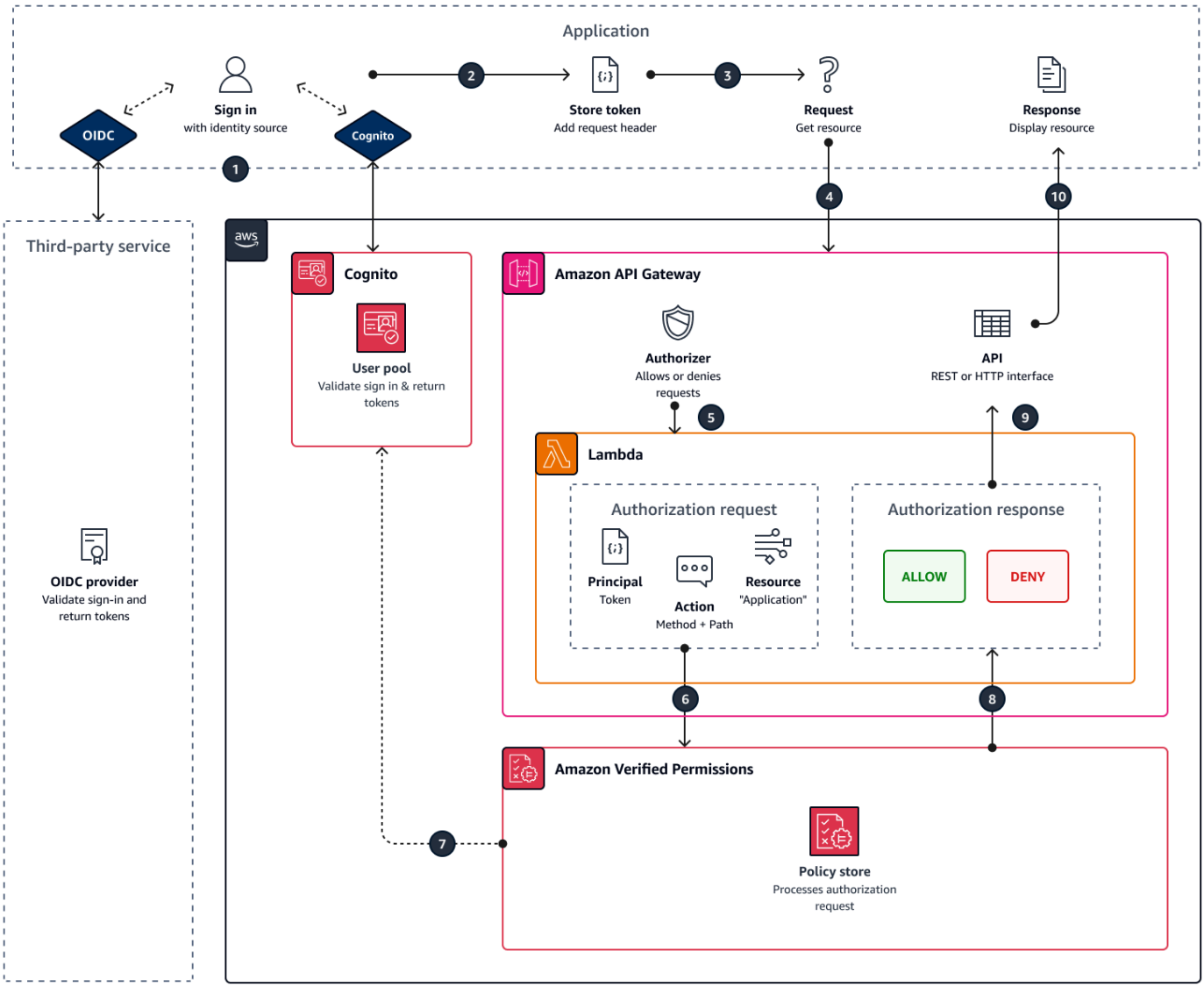
在連結至 API 和身分識別來源的原則存放區中，當應用程式向 API 發出要求時，應用程式會在授權標頭中顯示使用者集區權杖。原則存放區的身分識別來源提供已驗證權限的權杖驗證。令牌使用 IsAuthorizedWithTokenAPI 形成`principal`在授權請求。已驗證的權限會圍繞使用者的群組成員資格建立原則，如身分識別 (ID) 和存取權杖 (例如`cognito:groups`使用者集區) 中的群組宣告所示。您的 API 會在 Lambda 授權者中處理來自應用程式的權杖，並將其提交給已驗證的許可以進行授權決策。當您的 API 收到 Lambda 授權者的授權決定時，它會將請求傳遞給您的資料來源或拒絕該請求。

具有已驗證權限的身分識別來源和 API Gateway 授權的元件

- 可對使用者進行驗證和分組的 Amazon Cognito 使用者集區或 OIDC IdP。使用者的 Token 會填入群組成員資格，以及原則存放區中「已驗證權限」評估的主體或內容。
- 一個 API Gateway 其餘 API。例如，「已驗證的權限」會定義 API 路徑和 API 方法的動作`MyAPI::Action::get /photo`。
- 適用於您 API 的 Lambda 函數和 Lambda 授權器。Lambda 函數會從您的使用者集區接收不記名權杖，從已驗證的權限要求授權，然後將決策傳回給 API Gateway。「使用 Cognito 與 API Gateway 設定」工作流程會自動為您建立此 Lambda 授權器。
- 驗證權限原則存放區。原則存放區身分識別來源是您的使用者集區。原則存放區結構描述會反映 API 的設定，而政策會將使用者群組連結至允許的 API 動作。
- 使用 IdP 驗證使用者並將權杖附加至 API 要求的應用程式。

驗證權限如何授權 API 請求

當您建立新的原則存放區並選取 [使用 Cognito 和 API Gateway 設定] 選項時，[已驗證的權限] 會建立原則存放區結構描述和原則。結構描述和策略會反映 API 動作，以及您要授權執行動作的使用者集區群組。已驗證的權限也會建立 Lambda 函數和授權者。您必須在 API 中的方法上配置新的授權者。



1. 您的使用者透過 Amazon Cognito 或其他 OIDC IdP 使用您的應用程式登入。IdP 會以使用者的資訊發出 ID 和存取權杖。
2. 您的應用程序存儲 JWT。如需詳細資訊，請參閱 Amazon Cognito 開發人員指南中的搭配使用者集區使用權杖。
3. 您的使用者要求您的應用程式必須從外部 API 擷取的資料。

4. 您的應用程式會從 API Gateway 中的 REST API 要求資料。它附加一個 ID 或訪問令牌作為請求頭。
5. 如果您的 API 具有用於授權決策的緩存，則返回先前的響應。如果停用快取或 API 沒有目前的快取，API Gateway 會將請求參數傳遞給以權杖為基礎的 Lambda 授權者。
6. Lambda 函數會透過 IsAuthorizedWithTokenAPI 將授權要求傳送至「已驗證的權限」原則存放區。Lambda 函數傳遞授權決策的元素：
   a. 使用者的權杖做為主體。
   b. API 方法與 API 路徑相結合，例如`GetPhoto`，作為動作。
   c. `Application`作為資源的術語。
7. 已驗證的權限會驗證權杖。如需有關如何驗證 Amazon Cognito 權杖的詳細資訊，請參閱 Amazon Co gnito 開發人員指南中的使用 Amazon 驗證許可授權。
8. 「已驗證的權限」會根據原則存放區中的原則評估授權要求，並傳回授權決策。
9. Lambda 授權者會將 API Gateway 傳`Deny`回`Allow`或回應。
10. API 會傳回`ACCESS_DENIED`應用程式的資料或回應。您的應用程式會處理並顯示 API 要求的結果。

新增以屬性為基礎的存取控制 (ABAC)

具有 IdP 的典型驗證工作階段會傳回 ID 和存取權杖。您可以在應用程序請求中將這些令牌類型中的任何一種作為承載令牌傳遞給 API。根據您在建立原則存放區時的選擇，「已驗證的權限」會預期這兩種權杖類型的其中一種。這兩種類型都包含有關用戶組成員資格的信息。如需 Amazon Cognito 中權杖類型的詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用權杖搭配使用者集區。

建立原則存放區之後，您可以新增和擴充原則。例如，您可以在將群組新增至使用者集區時，將群組新增至原則。由於您的政策存放區已知道使用者集區以權杖呈現群組的方式，因此您可以針對具有新政策的任何新群組允許執行一組動作。

您也可以根據使用者屬性，將以群組為基礎的原則評估模型擴充為更精確的模型。用戶池令牌包含其他用戶信息，這些信息可以有助於授權決策。

身份證令牌

ID 令牌代表用戶的屬性，並具有最高級別的精細訪問控制。要評估電子郵件地址，電話號碼或自定義屬性（例如部門和經理），請評估 ID 令牌。

存取權杖

訪問令牌代表具有 OAuth 2.0 範圍的用戶權限。若要新增授權層或設定其他資源的要求，請評估存取權杖。例如，您可以驗證用戶是否位於適當的組中，並具有類似的範圍`PetStore.read`，通常授權對 API 的訪問權限。用戶池可以向具有資源服務器的令牌添加自定義範圍，並在運行時使用令牌自定義。

請參在結構描述和原則中使用身分識別來源閱在 ID 和訪問令牌中處理聲明的示例策略。

API 連結原則存放區的考量

當您在 [已驗證的權限] 主控台中建立 API 連結的原則存放區時，您正在為最終的生產部署建立測試。在移至生產環境之前，請為 API 和使用者集區建立固定組態。請考慮以下因素：

API Gateway 快取回應

在 API 連結的原則存放區中，「已驗證的權限」會建立 Lambda 授權器，其授權快取 TTL 為 120 秒。您可以在授權者中調整此值或關閉緩存。在啟用快取的授權者中，您的授權者每次都會傳回相同的回應，直到 TTL 到期為止。這可以通過等於請求階段的緩存 TTL 的持續時間來延長用戶池令牌的有效生命週期。

Amazon Cognito 群組可以重複使用

Amazon 驗證許可會根據使用者 ID 或存取權杖中的`cognito:groups`宣告來決定使用者集區使用者的群組成員資格。此宣告的值是使用者所屬之使用者集區群組易記名稱的陣列。您無法將使用者集區群組與唯一識別碼建立關聯。

您刪除和重新建立的使用者集區群組，其名稱與相同群組顯示在原則存放區中的相同名稱。當您從使用者集區中刪除群組時，請從原則存放區刪除群組的所有參照。

API 衍生的命名空間和結構描述 point-in-time

驗證權限會在某個時間點擷取您的 API：它只會在您建立原則存放區時查詢您的 API。當 API 的結構描述或名稱變更時，您必須更新原則存放區和 Lambda 授權者，或建立新的 API 連結政策存放區。已驗證的權限會從 API 的名稱衍生原則存放區命名空間。

Lambda 函數沒有 VPC 組態

已驗證許可為您的 API 授權者建立的 Lambda 函數未連線至 VPC。默認情況下。僅限於私有 VPC 的網路存取權的 API 無法與 Lambda 函數進行通訊，該函數使用已驗證的權限授權存取請求。

已驗證的權限部署授權者資源 CloudFormation

若要建立 API 連結的原則存放區，您必須將高權限的主體登入「已驗證的權限」 AWS 主控台。此使用者會部署跨多個 AWS 服務建立資源的 AWS CloudFormation 堆疊。此主體必須具有在已驗證許可、 IAM Lambda 和 API Gateway 中新增和修改資源的權限。最佳做法是，請勿與組織中的其他系統管理員共用這些認證。

如轉移到生產環境 AWS CloudFormation需已驗證權限建立的資源概觀，請參閱。

轉移到生產環境 AWS CloudFormation

API 連結原則存放區是一種快速建置 API Gateway API 授權模型的方法。它們被設計為作為應用程序授權組件的測試環境。建立測試原則存放區之後，請花時間精簡政策、結構描述和 Lambda 授權者。

您可以調整 API 的架構，需要對策存放區結構描述和政策進行等效調整。API 連結的政策存放區不會從 API 架構自動更新其結構描述 — 經過驗證的權限只會在您建立原則存放區時輪詢 API。如果您的 API 有足夠的變更，您可能必須使用新的原則存放區重複此程序。

當您的應用程式和授權模型已準備好部署到生產環境時，請整合您與自動化程序一起開發的 API 連結原則存放區。最佳作法是，建議您將原則存放區結構描述和原則匯出至可部署到其他 AWS 帳戶 和的 AWS CloudFormation 範本 AWS 區域。

API 連結政策存放區程序的結果為初始政策存放區和 Lambda 授權者。Lambda 授權者具有數個相依資源。已驗證的權限會在自動產 CloudFormation 生的堆疊中部署這些資源。若要部署到生產環境，您必須將原則存放區和 Lambda 授權者資源收集到範本中。API 連結的原則存放區由下列資源組成：

1. AWS::VerifiedPermissions::PolicyStore：將您的結構描述複製到`SchemaDefinition`物件。將字"元轉義為\"。
2. AWS::VerifiedPermissions::IdentitySource：從測試策略存儲中複製輸出GetIdentitySource 中的值，並根據需要進行修改。
3. 一或多個 AWS::VerifiedPermissions::Policy：將您的政策聲明複製到`Definition`物件。將字"元轉義為\"。
4. AWS:: Lambda:: 函數,: AWS:IAM: Role,::: 策略,AWS:IAM:: 授權者, AWS:ApiGateway: 授權者, AWS::Lambda::Permission: 從建立政策存放區時部署的「已驗證權限」堆疊中複製範本。

以下範本是原則存放區的範例。您可以將 Lambda 授權者資源從現有堆疊附加到此範本。

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
```

```
    "Resources": {
        "MyExamplePolicyStore": {
            "Type": "AWS::VerifiedPermissions::PolicyStore",
            "Properties": {
                "ValidationSettings": {
                    "Mode": "STRICT"
                },
                "Description": "ApiGateway: PetStore/test",
                "Schema": {
                    "CedarJson": "{\"PetStore\":{\"actions\":{\"get /pets\":
{\"appliesTo\":{\"principalTypes\":[\"User\"],\"resourceTypes\":[\"Application\"],
\"context\":{\"type\":\"Record\",\"attributes\":{}}}},\"get /\":{\"appliesTo\":
{\"principalTypes\":[\"User\"],\"resourceTypes\":[\"Application\"],\"context\":{\"type
\":\"Record\",\"attributes\":{}}}},\"get /pets/{petId}\":{\"appliesTo\":{\"context
\":{\"type\":\"Record\",\"attributes\":{}},\"resourceTypes\":[\"Application\"],
\"principalTypes\":[\"User\"]}},\"post /pets\":{\"appliesTo\":{\"principalTypes\":
[\"User\"],\"resourceTypes\":[\"Application\"],\"context\":{\"type\":\"Record\",
\"attributes\":{}}}}},\"entityTypes\":{\"Application\":{\"shape\":{\"type\":\"Record\",
\"attributes\":{}}},\"User\":{\"memberOfTypes\":[\"UserGroup\"],\"shape\":{\"attributes
\":{},\"type\":\"Record\"}},\"UserGroup\":{\"shape\":{\"type\":\"Record\",\"attributes
\":{}}}}}}"
                }
            }
        },
        "MyExamplePolicy": {
            "Type": "AWS::VerifiedPermissions::Policy",
            "Properties": {
                "Definition": {
                    "Static": {
                        "Description": "Policy defining permissions for testgroup
 cognito group",
                        "Statement": "permit(\nprincipal in PetStore::UserGroup::
\"us-east-1_EXAMPLE|testgroup\",\naction in [\n  PetStore::Action::\"get /\",
\n  PetStore::Action::\"post /pets\",\n  PetStore::Action::\"get /pets\",\n
 PetStore::Action::\"get /pets/{petId}\"\n],\nresource);"
                    }
                },
                "PolicyStoreId": {
                    "Ref": "MyExamplePolicyStore"
                }
            },
            "DependsOn": [
                "MyExamplePolicyStore"
            ]
```

```
        },
        "MyExampleIdentitySource": {
            "Type": "AWS::VerifiedPermissions::IdentitySource",
            "Properties": {
                "Configuration": {
                    "CognitoUserPoolConfiguration": {
                        "ClientIds": [
                            "1example23456789"
                        ],
                        "GroupConfiguration": {
                            "GroupEntityType": "PetStore::UserGroup"
                        },
                        "UserPoolArn": "arn:aws:cognito-idp:us-
east-1:123456789012:userpool/us-east-1_EXAMPLE"
                    }
                },
                "PolicyStoreId": {
                    "Ref": "MyExamplePolicyStore"
                },
                "PrincipalEntityType": "PetStore::User"
            },
            "DependsOn": [
                "MyExamplePolicyStore"
            ]
        }
    }
}
```

疑難排解 API 連結的原則存放區

使用此處的資訊可協助您診斷和修正建立 Amazon 驗證許可 API 連結政策存放區時的常見問題。

主題

- 我更新了我的政策，但授權決定沒有改變
- 我將 Lambda 授權器附加到我的 API，但它沒有生成授權請求
- 我收到了意想不到的授權決定，想要檢閱授權邏輯
- 我想要從我的 Lambda 授權者尋找記錄
- 我的 Lambda 授權者不存在
- 我的 API 位於私有 VPC 中，無法調用授權者
- 我想在我的授權模型中處理其他用戶屬性

- 我想要新增動作、動作前後關聯屬性或資源屬性

我更新了我的政策，但授權決定沒有改變

根據預設，「已驗證的權限」會設定 Lambda 授權者快取授權決策 120 秒。請在兩分鐘後再試一次，或停用授權者的快取。如需詳細資訊，請參閱 Amazon API 閘道開發人員指南中的啟用 API 快取以增強回應速度。

我將 Lambda 授權器附加到我的 API，但它沒有生成授權請求

若要開始處理請求，您必須部署附加授權者的 API 階段。如需詳細資訊，請參閱 Amazon API 閘道開發人員指南中的部署 REST API。

我收到了意想不到的授權決定，想要檢閱授權邏輯

API 連結的政策存放區程序會為您的授權者建立 Lambda 函數。驗證權限會自動將授權決策的邏輯構建到授權者函數中。您可以在建立原則存放區之後返回，以檢閱和更新函數中的邏輯。

若要從 AWS CloudFormation 主控台尋找 Lambda 函數，請選擇新政策存放區 [概觀] 頁面上的 [檢查部署] 按鈕。

您也可以在 AWS Lambda 控制台中找到您的功能。導覽至原則存放區中 AWS 區域 的主控台，然後搜尋前置字元為的函數名稱`AVPAuthorizerLambda`。如果您已建立多個 API 連結的原則存放區，請使用函數的上次修改時間，將它們與原則存放區建立關聯。

我想要從我的 Lambda 授權者尋找記錄

Lambda 函數會收集指標，並在 Amazon CloudWatch 中記錄其叫用結果。若要檢閱記錄，請在 Lambda 主控台中找到您的函數，然後選擇監控索引標籤。選取 [檢視 CloudWatch 記錄檔] 並檢閱記錄群組中的項目。

如需 Lambda 函數日誌的詳細資訊，請參閱AWS Lambda 開發人員指南 AWS Lambda中的搭配使用 Amazon CloudWatch 日誌。

我的 Lambda 授權者不存在

完成 API 連結政策存放區的設定後，您必須將 Lambda 授權器附加至您的 API。如果在 API Gateway 主控台中找不到授權者，則原則存放區的其他資源可能已失敗或尚未部署。API 連結的原則儲存區將這些資源部署在堆疊中 AWS CloudFormation 。

「已驗證的權限」會在建立程序結束時顯示含有「檢查」部署標籤的連結。如果您已離開此畫面，請前往 CloudFormation 主控台並搜尋最近的堆疊，尋找前置詞為的名稱。AVPAuthorizer-<policy store ID> CloudFormation 在堆疊部署的輸出中提供寶貴的疑難排解資訊。

如需疑難排解 CloudFormation 堆疊的說明，請參閱AWS CloudFormation 使用指南 CloudFormation中的疑難排解

我的 API 位於私有 VPC 中，無法調用授權者

已驗證的權限不支援透過 VPC 端點存取 Lambda 授權者。您必須在 API 和做為授權者的 Lambda 函數之間開啟網路路徑。

我想在我的授權模型中處理其他用戶屬性

API 連結的原則存放區程序會從使用者權杖中的群組宣告衍生「已驗證的權限」原則。若要更新授權模型以考量其他使用者屬性，請將這些屬性整合到您的策略中。

您可以將來自 Amazon Cognito 使用者集區的 ID 和存取權杖中的許多宣告對應至已驗證的許可政策陳述式。例如，大多數用戶在其 ID 令牌中都有email聲明。如需將宣告從身分來源新增至政策的詳細資訊，請參閱在結構描述和原則中使用身分識別來源。

我想要新增動作、動作前後關聯屬性或資源屬性

API 連結的政策存放區及其所建立的 Lambda 授權者都是資源。 point-in-time它們會在建立時反映 API 的狀態。策略存放區結構描述不會將任何前後關聯屬性指派給動作，也不會將任何屬性或父項指派給預設Application資源。

當您將動作 (路徑和方法) 新增至 API 時，您必須更新原則存放區以瞭解新動作。您還必須更新 Lambda 授權者，以處理新動作的授權請求。您可以從新的原則存放區重新開始，也可以更新現有的原則存放區。

若要更新現有的原則存放區，請找出您的功能。檢查自動產生函數中的邏輯，並加以更新，以處理新動作、屬性或前後關聯。然後編輯您的結構描述以包含新動作和屬性。

切換驗證權限原則存放區

AWS Management Console

切換原則存放區或建立其他原則存放區

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [目前原則存放區] 旁邊的 [切換]。
3. 您可以在現有的原則存放區之間切換或建立其他原則存放區。
   - 若要切換原則存放區，請選擇要切換至的原則存放區識別碼。
   - 若要建立新的原則存放區，請選擇 [建立新原則存放區]。請遵循中的說明進行建立驗證權限原則存放區

AWS CLI

切換原則存放區或建立其他原則存放區

AWS CLI不會維護「預設」原則存放區。相反地，大多數命AWS CLI令會使用`--policy-store-id`來指定每個命令要使用的原則存放區。

若要建立新的原則存放區，請使用create-policy-store命令。

刪除已驗證權限原則商店

AWS Management Console

刪除策略存放區

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 Settings (設定)。
3. 選擇 [刪除此原則存放區]。
4. 在文字方塊`delete`中鍵入，然後選擇 [刪除]。

AWS CLI

刪除策略存放區

您可以使用delete-policy-store作業刪除原則存放區。

```
$ aws verifiedpermissions delete-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

如果成功，此命令不會產生輸出。

Amazon 驗證許可政策存儲模式

結構描述是應用程式所支援之實體類型結構的宣告，以及您的應用程式可能在授權要求中提供的動作。

如需詳細資訊，請參閱 Cedar 政策語言參考指南中的 Cedar 結構描述格式。

> **Note**
>
> 在已驗證的權限中使用結構描述是可選的，但強烈建議在生產軟體中使用這些結構描述。當您建立新原則時，已驗證的權限可以使用結構描述來驗證範圍和條件中參照的實體和屬性，以避免可能導致系統行為混淆的原則中的錯字和錯誤。如果您啟動原則驗證，則所有新原則都必須符合綱要。

AWS Management Console

建立結構描述

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [結構描述]。
3. 選擇建立結構描述。

AWS CLI

若要提交新結構描述，或使用覆寫現有的綱要 AWS CLI。

您可以執行類似下列範例的 AWS CLI 命令來建立原則存放區。

請考慮包含下列 Cedar 內容的結構描述：

```
{
    "MySampleNamespace": {
        "actions": {
            "remoteAccess": {
                "appliesTo": {
                    "principalTypes": [ "Employee" ]
                }
            }
```

```
        },
        "entityTypes": {
            "Employee": {
                "shape": {
                    "type": "Record",
                    "attributes": {
                        "jobLevel": {"type": "Long"},
                        "name": {"type": "String"}
                    }
                }
            }
        }
    }
}
```

您必須首先將 JSON 轉義為單行字符串，並以其數據類型的聲明進行前面：cedarJson。下列範例會使用下列包含 JSON 結構描述逸出版本的schema.json檔案內容。

> **Note**
>
> 這裡的例子是為了可讀性而換行。您必須將整個檔案放在一行中，命令才能接受它。

```
{"cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {\"appliesTo
\":
{\"principalTypes\": [\"Employee\"]}}},\"entityTypes\": {\"Employee\": {\"shape\":
{\"attributes\": {\"jobLevel\": {\"type\": \"Long\"},\"name\": {\"type\": \"String
\"}},
\"type\": \"Record\"}}}}}"}
```

```
$ aws verifiedpermissions put-schema \
    --definition file://schema.json \
    --policy-store PSEXAMPLEabcdefg111111
{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "namespaces": [
        "MySampleNamespace"
    ],
    "createdDate": "2023-07-17T21:07:43.659196+00:00",
    "lastUpdatedDate": "2023-08-16T17:03:53.081839+00:00"
}
```

AWS SDKs

您可以使用 PutSchema API 建立政策存放區。如需詳細資訊，請參閱 Amazon 驗證許可 API 參考指南PutSchema中的。

在視覺模式中編輯資料架

當您在 [已驗證的權限] 主控台中選取 [結構描述] 時，視覺模式會顯示構成結構描述的 [實體] 類型和 [動作]。在這個頂層檢視或任何實體的詳細資料中，您可以選擇 [編輯結構描述] 開始更新您的結構描述。視覺模式不適用於某些結構描述格式，例如巢狀記錄。

視覺化結構描述編輯器從一系列圖表開始，這些圖表說明您的架構中實體之間的關係。選擇「展開」，可最大化結構描述的實體關聯性檢視。

動作圖

「動作」圖表檢視會列出您在原則存放區中設定的主參與者類型、它們有資格執行的動作，以及他們有資格對其執行動作的「資源」。實體之間的線條表示您能夠建立原則，讓主參與者對資源採取動作。如果您的動作圖表未指出兩個實體之間的關係，您必須先建立兩個實體之間的關係，才能在原則中允許或拒絕它。選取實體以查看屬性概觀，並向下鑽研以檢視完整詳細資訊。選擇 [動作篩選] | [資源類型] | [主參與者類型]，即可在檢視中查看僅包含其本身連線的實體。

實體類型圖

實體類型圖表著重於主參與者與資源之間的關係。如果您想要瞭解結構描述中複雜的巢狀父關係，請檢閱此圖表。將游標暫留在實體上，以向下鑽研其所具有的父項關係。

下圖是您的模式中的實體類型和操作的列表視圖。當您想要立即檢視特定動作或實體類型的詳細資訊時，清單檢視非常有用。選取要檢視詳細資訊的任何實體。

在視覺化模式中編輯已驗證的權限結構描述

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [結構描述]。
3. 選擇「視覺」模式。複查實體關係圖表，並規劃您要對結構描述進行的變更。您可以選擇按一個圖元進行篩選，以檢查其與其他圖元的個別連接。
4. 選擇編輯結構描述。

5. 在 [詳細資料] 區段中，輸入結構描述的命名空間。
6. 在 [實體類型] 區段中，選擇 [新增實體類型]。
7. 輸入實體的名稱。
8. (選擇性) 選擇「新增父項」以新增新實體所屬的父項實體。若要移除已新增至實體的父項，請選擇父項名稱旁邊的 [移除]。
9. 選擇新增屬性以將屬性新增至實體。輸入屬性名稱，然後為實體的每個屬性選擇屬性類型。驗證權限在根據結構描述驗證策略時，會使用指定的屬性值。選取每個屬性是否為必要。若要移除已新增至實體的屬性，請選擇屬性旁邊的 [移除]。
10. 選擇新增實體類型，將實體新增至結構描述。
11. 在「動作」區段中，選擇「新增動作」。
12. 輸入動作的名稱。
13. (選擇性) 選擇 [新增資源] 以新增動作套用的資源類型。若要移除已新增至動作的資源類型，請選擇資源類型名稱旁邊的 [移除]。
14. (選擇性) 選擇「新增主參與者」來新增套用動作的主參與者類型。欲移除已新增至動作的主參與者類型，請選擇主參與者類型名稱旁邊的「移除」(Remove)。
15. 選擇 [新增屬性] 以新增可新增至授權要求中動作前後關聯的屬性。輸入「屬性」名稱，並選擇每個屬性的「屬性」類型。驗證權限在根據結構描述驗證策略時，會使用指定的屬性值。選取每個屬性是否為必要。欲移除已新增至動作的屬性，請選擇屬性旁邊的「移除」。
16. 選擇新增動作。
17. 將所有實體類型和動作新增至結構描述之後，請選擇 [儲存變更]。

在 JSON 模式下編輯結構定義

若要在 JSON 模式中編輯已驗證的權限結構描述

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [結構描述]。
3. 選擇 JSON 模式，然後選擇 [編輯結構定義]。
4. 在「內容」欄位中輸入 JSON 結構定義的內容。在解決所有語法錯誤之前，您無法將更新儲存至結構定義。您可以選擇「格式化 JSON」，以建議的間距和縮排格式化結構描述的 JSON 語法。
5. 選擇儲存變更。

刪除結構描述

AWS Management Console

刪除已驗證的權限結構描述

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [結構描述]。
3. 選擇 [刪除綱要]。

AWS CLI

刪除已驗證的權限結構描述

沒有刪除結構描述命令。您可以在`cedarJson`欄位中使用具有空白結構描述的`put-schema`命令來刪除原則存放區中的結構描述。一個空的結構描述由一對大括號 '{}' 表示。

```
$ aws verifiedpermissions put-schema \
    --policy-store-id PSEXAMPLEabcdefg111111 \
    --definition cedarJson='{}'{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "namespaces": [],
    "createdDate": "2023-06-14T21:55:27.347581Z",
    "lastUpdatedDate": "2023-06-19T17:55:04.95944Z"
}
```

Amazon 驗證許可政策驗證模式

您可以在 [已驗證的權限] 中設定原則驗證模式，以控制是否針對原則存放區中的結構描述驗證原則變更。

> **Important**
>
> 當您開啟原則驗證時，會根據原則存放區中的結構描述驗證所有嘗試建立或更新原則或原則範本的嘗試。如果驗證失敗，「已驗證的權限」會拒絕要求。

AWS Management Console

若要設定原則存放區的原則驗證模式

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 選擇設定。
3. 在 [原則驗證模式] 區段中，選擇 [修改]。
4. 執行以下任意一項：
   - 若要啟動原則驗證並強制所有原則變更必須根據您的綱要驗證，請選擇嚴格 (建議) 圓鈕。
   - 若要關閉原則變更的原則驗證，請選擇 [關閉] 圓鈕。輸入`confirm`以確認將不再針對您的結構描述驗證原則的更新。
5. 選擇儲存變更。

AWS CLI

若要設定原則存放區的驗證模式

您可以使用UpdatePolicyStore作業並為ValidationSettings參數指定不同的值，來變更原則存放區的驗證模式。

```
$ aws verifiedpermissions update-policy-store \
    --validation-settings "mode=OFF",
    --policy-store-id PSEXAMPLEabcdefg111111
{
```

```
    "createdDate": "2023-05-17T18:36:10.134448+00:00",
    "lastUpdatedDate": "2023-05-17T18:36:10.134448+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "validationSettings": {
        "Mode": "OFF"
    }
}
```

如需詳細資訊，請參閱 Cedar 政策語言參考指南中的原則驗證。

Amazon 驗證許可政策

策略是允許或禁止主體對資源採取一或多個動作的陳述式。每個策略的評估與任何其他策略無關。如需有關 Cedar 原則結構化及評估方式的詳細資訊，請參閱 Cedar 政策語言參考指南中的 Cedar 針對架構進行原則驗證。

Important

當您撰寫參照主參與者、資源和動作的 Cedar 原則時，您可以定義用於每個元素的唯一識別元素。我們強烈建議您遵循以下最佳做法：

- 對所有主參與者和資源識別碼使用通用唯一識別碼 (UUID) 等值。

  例如，如果使用者`jane`離開公司，而您稍後讓其他人使用該名稱`jane`，則該新使用者會自動取得仍然參照之原則所授與的所有項目的存取權`User::"jane"`。Cedar 無法區分新用戶和舊用戶。這適用於主參與者和資源識別碼。一律使用保證唯一且絕不重複使用的識別碼，以確保您不會因為原則中存在舊識別碼而無意中授予存取權。

  當您為實體使用 UUID 時，我們建議您使用//註釋說明符和實體的「友好」名稱來跟隨它。這有助於使您的政策更容易理解。例如：校長 == 用戶:: "A1b2C3D4-E5F6-A1B2-C3D4-例子

- 請勿將個人識別資訊、機密或敏感資訊納入主參與者或資源的唯一識別碼中。這些識別碼包含在 AWS CloudTrail 追蹤中共用的記錄項目中。

Amazon 驗證許可中的實體格式

Amazon 驗證許可使用 Cedar 政策語言來建立政策。支援的原則語法和資料類型符合 Cedar 政策語言參考指南中 Cedar 主題支援的基本原則建構中概述的語法和資料類型，以及 Cedar 主題所支援的資料類型。不過，在提出授權要求時，「已驗證的權限」和「Cedar」在實體格式方面存在差異。

已驗證權限中實體的 JSON 格式與 Cedar 有以下幾種不同：

- 在已驗證的權限中，JSON 物件必須將其所有索引鍵值組包裝在 JSON 物件中，名`Record`稱為。
- 已驗證權限中的 JSON 清單必須包裝在金鑰名稱所在的 JSON 金鑰值組中，`Set`且值是 Cedar 的原始 JSON 清單。

- 對於StringLong、和Boolean類型名稱，Cedar 中的每個鍵值對都會由已驗證權限中的 JSON 物件取代。物件的名稱是原始金鑰名稱。在 JSON 物件中，有一個索引鍵值組，其中索引鍵名稱是標量值的類型名稱 (String、Long、或Boolean) 和值是來自 Cedar 實體的值。
- Cedar 實體和已驗證權限實體的語法格式有下列不同：

雪松格式	已驗證權限格式
uid	Identifier
type	EntityType
id	EntityId
attrs	Attributes
parents	Parents

下列範例顯示如何使用 Cedar 格式化清單中的實體。

```
[
  {
    "number": 1
  },
  {
    "sentence": "Here is an example sentence"
  },
  {
    "Question": false
  }
]
```

下列 exmple 顯示先前 Cedar 清單範例中的相同實體如何在已驗證的權限中格式化。

```
{
  "Set": [
    {
      "Record": {
        "number": {
          "Long": 1
        }
```

```
      }
    },
    {
      "Record": {
        "sentence": {
          "String": "Here is an example sentence"
        }
      }
    },
    {
      "Record": {
        "question": {
          "Boolean": false
        }
      }
    }
  ]
}
```

下列範例顯示 Cedar 實體如何格式化以評估授權要求中的原則。

```
[
    {
        "uid": {
            "type": "PhotoApp::User",
            "id": "alice"
        },
        "attrs": {
            "age": 25,
            "name": "alice",
            "userId": "123456789012"
        },
        "parents": [
            {
                "type": "PhotoApp::UserGroup",
                "id": "alice_friends"
            },
            {
                "type": "PhotoApp::UserGroup",
                "id": "AVTeam"
            }
        ]
    },
```

```
    {
        "uid": {
            "type": "PhotoApp::Photo",
            "id": "vacationPhoto.jpg"
        },
        "attrs": {
            "private": false,
            "account": {
                "__entity": {
                    "type": "PhotoApp::Account",
                    "id": "ahmad"
                }
            }
        },
        "parents": []
    },
    {
        "uid": {
            "type": "PhotoApp::UserGroup",
            "id": "alice_friends"
        },
        "attrs": {},
        "parents": []
    },
    {
        "uid": {
            "type": "PhotoApp::UserGroup",
            "id": "AVTeam"
        },
        "attrs": {},
        "parents": []
    }
]
```

下列範例顯示前一個 Cedar 範例中的相同實體如何在「已驗證的權限」中格式化。

```
[
    {
        "Identifier": {
            "EntityType": "PhotoApp::User",
            "EntityId": "alice"
        },
        "Attributes": {
```

```
            "age": {
                "Long": 25
            },
            "name": {
                "String": "alice"
            },
            "userId": {
                "String": "123456789012"
            }
        },
        "Parents": [
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "alice_friends"
            },
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "AVTeam"
            }
        ]
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::Photo",
            "EntityId": "vacationPhoto.jpg"
        },
        "Attributes": {
            "private": {
                "Boolean": false
            },
            "account": {
                "EntityIdentifier": {
                    "EntityType": "PhotoApp::Account",
                    "EntityId": "ahmad"
                }
            }
        },
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "alice_friends"
        },
```

```
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "AVTeam"
        },
        "Parents": []
    }
]
```

創建 Amazon 驗證許可靜態政策

您可以建立 Cedar 靜態政策，允許或拒絕主參與者針對應用程式的指定資源執行指定的動作。

AWS Management Console

若要建立靜態政策

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 Policies (政策)。
3. 選擇 [建立原則]，然後選擇 [建立靜態策略]
4. 在 [原則效果] 區段中，選擇當要求符合原則時，原則是 [允許] 還是 [禁止]。
5. 在「主參與者範圍」欄位中，選擇要套用原則的主參與者範圍。
   - 選擇特定主參與者，將保單套用至特定的主參與者。指定主參與者的實體類型和識別碼，以禁止採取原則中指定的動作。
   - 選擇主參與者群組，將原則套用至主參與者群組。在主參與者群組欄位中輸入主參與者群組名稱。
   - 選擇 [所有主參與者]，將原則套用至原則存放區中的所有主參與者。
6. 在 [資源範圍] 欄位中，選擇要套用原則的資源範圍。
   - 選擇 [特定資源]，將策略套用至特定資源。指定應套用策略之資源的實體類型和識別碼。
   - 選擇資源群組以將策略套用至資源群組。在資源群組欄位中輸入資源群組名稱。
   - 選擇 [所有資源]，將策略套用至原則存放區中的所有資源。
7. 在「動作範圍」段落中，選擇要套用原則的資源範圍。

- 選擇特定動作集，將策略套用至一組動作。選取要套用策略的動作旁邊的核取方塊。
- 選擇 [所有動作]，將原則套用至原則存放區中的所有動作。

8. 選擇下一步。
9. 在「政策」區段中，檢閱您的 Cedar 政策。您可以選擇 [格式]，以建議的間距和縮排格式化原則的語法。如需詳細資訊，請參閱 Cedar 政策語言參考指南中的 Cedar 中的基本原則建構。
10. 在 [詳細資料] 區段中，輸入原則的選擇性描述。
11. 選擇建立政策。

AWS CLI

若要建立靜態政策

您可以使用CreatePolicy作業建立靜態政策。下列範例會建立簡單的靜態原則。

```
$ aws verifiedpermissions create-policy \
    --definition "{ \"static\": { \"Description\": \"MyTestPolicy\", \"Statement\":
 \"permit(principal,action,resource) when {principal.owner == resource.owner};\"}}"
 \
    --policy-store-id PSEXAMPLEabcdefg111111
{
"Arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/
SPEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T20:33:01.730817+00:00",
    "lastUpdatedDate": "2023-05-16T20:33:01.730817+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC"
}
```

編輯 Amazon 驗證許可靜態政策

您可以在原則存放區中編輯現有的 Cedar 靜態政策。您只能直接更新靜態原則。您只能變更靜態政策的某些元素：

- 原則所`action`參照的。
- 條件子句，例如`when`和`unless`。

您無法變更靜態政策的下列元素：

- 將原則從靜態政策變更為範本連結政策。
- 從`permit`或變更靜態原則的效果`forbid`。
- 由靜態策略`principal`引用。
- 由靜態策略`resource`引用。

若要變更範本連結政策，您必須改為更新範本。如需詳細資訊，請參閱 編輯策略範本。

AWS Management Console

若要編輯靜態策略

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 Policies (政策)。
3. 選擇要編輯的靜態策略旁邊的圓鈕，然後選擇 [編輯]。
4. 在 [原則主體] 區段中，更新靜態原則的`action`或條件子句。您無法更新原則`resource`的效果`principal`、或。
5. 選擇更新政策。

> **Note**
>
> 如果在原則存放區中啟用原則驗證，則更新靜態原則會導致「已驗證的權限」針對原則存放區中的結構描述驗證原則。如果更新的靜態政策未通過驗證，則作業會失敗，並且不會儲存更新。

AWS CLI

若要編輯靜態策略

您可以使用UpdatePolicy作業來編輯靜態政策。下列範例會編輯簡單的靜態政策。

此範例使用檔案`definition.txt`來包含原則定義。

```
{
    "static": {
```

```
        "description":  "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
    }
}
```

以下命令引用該文件。

```
$ aws verifiedpermissions create-policy \
    --definition file://definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111

{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

檢視政策

AWS Management Console

檢視您的已驗證權限政策

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 Policies (政策)。此時會顯示您建立的所有策略。
3. 選擇搜尋文字方塊，依主參與者或資源來篩選策略。
4. 選擇策略旁邊的圓鈕，以顯示有關策略的詳細資訊，例如建立、更新策略的時間和策略內容。

5. 您可以選擇策略旁邊的圓鈕，然後選擇刪除來刪除策略。選擇 [刪除原則] 以確認刪除原則。

AWS CLI

列出原則存放區中所有可用的原則

您可以使用作業來檢視原則清GetPolicy單。下列範例會擷取包含靜態政策和範本連結原則的清單。

```
$ aws verifiedpermissions list-policies \
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "Policies": [
        {
            "createdDate": "2023-05-17T18:38:31.359864+00:00",
            "definition": {
                "static": {
                    "Description": "Grant everyone of janeFriends UserGroup access
 to the vacationFolder Album"
                }
            },
            "lastUpdatedDate": "2023-05-18T16:15:04.366237+00:00",
            "policyId": "SPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "STATIC",
            "resource": {
                "entityId": "publicFolder",
                "entityType": "Album"
            }
        },
        {
            "createdDate": "2023-05-22T18:57:53.298278+00:00",
            "definition": {
                "templateLinked": {
                    "policyTemplateId": "PTEXAMPLEabcdefg111111"
                }
            },
            "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
            "policyId": "TPEXAMPLEabcdefg111111",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyType": "TEMPLATELINKED",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
```

```
            },
            "resource": {
                "entityId": "VacationPhoto94.jpg",
                "entityType": "Photo"
            }
        }
    ]
}
```

若要檢視個別策略的詳細資料

您可以使用GetPolicy作業擷取原則的詳細資訊。下列範例會擷取範本連結原則的詳細資料。

```
$ aws verifiedpermissions get-policy \
    --policy-id TPEXAMPLEabcdefg111111
    --policy-store-id PSEXAMPLEabcdefg111111

{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/
TPEXAMPLEabcdefg111111",
    "createdDate": "2023-03-15T16:03:07.620867Z",
    "lastUpdatedDate": "2023-03-15T16:03:07.620867Z",
    "policyDefinition": {
        "templatedPolicy": {
            "policyTemplateId": "PTEXAMPLEabcdefg111111",
            "principal": {
                "entityId": "alice",
                "entityType": "User"
            },
            "resource": {
                "entityId": "Vacation94.jpg",
                "entityType": "Photo"
            }
        }
    },
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATELINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "Vacation94.jpg",
```

```
        "entityType": "Photo"
    }
}
```

Amazon 驗證許可示例政策

下列「已驗證權限」原則範例是根據 Cedar 原則語言參考指南中「範例結構描述」一節中 PhotoFlash 所述的假設應用程式所定義的結構描述為基礎。如需 Cedar 原則語法的詳細資訊，請參閱 Cedar 政策語言參考指南中的 Cedar 中的基本原則建構。

政策範例

- 允許存取個別實體
- 允許存取實體群組
- 允許存取任何實體
- 允許存取實體的屬性 (ABAC)
- 拒絕存取

允許存取個別實體

此範例顯示如何建立允許使用者檢視`alice`相片的原則`VacationPhoto94.jpg`。

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
);
```

允許存取實體群組

此範例顯示如何建立允許群組中的任`alice_friends`何人檢視相片的政策`VacationPhoto94.jpg`。

```
permit(
  principal in Group::"alice_friends",
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
```

```
);
```

此範例顯示如何建立允許使用者alice檢視相簿中任何相片的政策alice_vacation。

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource in Album::"alice_vacation"
);
```

此範例顯示如何建立政策，讓使用者alice檢視、編輯或刪除相簿中的任何相片alice_vacation。

```
permit(
  principal == User::"alice",
  action in [Action::"view", Action::"edit", Action::"delete"],
  resource in Album::"alice_vacation"
);
```

此範例顯示如何建立允許相簿alice中使用者存取權限的原則alice_vacation，其中admin是架構階層中定義的群組，其中包含檢視、編輯和刪除相片的權限。

```
permit(
  principal == User::"alice",
  action in PhotoflashRole::"admin",
  resource in Album::"alice_vacation"
);
```

此範例顯示如何建立允許相簿alice中使用者存取權限的原則alice_vacation，其中viewer是架構階層中定義的群組，其中包含檢視相片和留言權限的群組。政策中列出的第二個動作也會授與使alice用者edit權限。

```
permit(
  principal == User::"alice",
  action in [PhotoflashRole::"viewer", Action::"edit"],
  resource in Album::"alice_vacation"
)
```

允許存取任何實體

此範例顯示如何建立允許任何已驗證的主體檢視相簿的原則alice_vacation。

```
permit(
  principal,
  action == Action::"view",
  resource in Album::"alice_vacation"
);
```

此範例顯示如何建立政策，以允許使用者alice列出jane帳戶中的所有相簿、列出每個相簿中的相片，以及檢視帳戶中的相片。

```
permit(
  principal == User::"alice",
  action in [Action::"listAlbums", Action::"listPhotos", Action::"view"],
  resource in Account::"jane"
);
```

此範例顯示如何建立策略，以允許使用者alice對相簿中的資源執行任何動作jane_vaction。

```
permit(
  principal == User::"alice",
  action,
  resource in Album::"jane_vacation"
);
```

允許存取實體的屬性 (ABAC)

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。「已驗證權限」可讓屬性附加至主參與者、動作及資源。然後可以在原則的when和unless子句中參照這些屬性，這些原則會評估構成請求前後關聯之主參與者、動作和資源的屬性。

下列範例會使用在 Cedar 原則語言參考指南〈範例結構 PhotoFlash 描述〉一節中所述的假設應用程式中定義的屬性。

此範例顯示如何建立政策，允許HardwareEngineering部門中任何工作層級大於或等於 5 的主參與者檢視和列出相簿中的相片device_prototypes。

```
permit(
  principal,
  action in [Action::"listPhotos", Action::"view"],
  resource in Album::"device_prototypes"
)
```

```
when {
  principal.department == "HardwareEngineering" &&
  principal.jobLevel >= 5
};
```

此範例顯示如何建立允許使用者alice檢視任何檔案類型資源的策略JPEG。

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource
)
when {
  resource.fileType == "JPEG"
};
```

動作具有前後關聯屬性。您必須在授權要求中context傳遞這些屬性。此範例顯示如何建立允許使用者alice執行任何readOnly動作的策略。您也可以為結構描述中的動作設定appliesTo屬性。例如，當您想要確保使用者只能嘗試授權ViewPhoto類型的資源時，這會指定資源的有效動作PhotoFlash::Photo。

```
permit(
  principal == PhotoFlash::User::"alice",
  action,
  resource
) when {
    context has readOnly &&
    context.readOnly == true
};
```

不過，在結構描述中設定動作屬性的更好方法是將它們排列到功能性動作群組中。
例如，您可以建立名ReadOnlyPhotoAccess為ReadOnlyPhotoAccess並設定PhotoFlash::Action::"ViewPhoto"為作為動作群組成員的動作。此範例顯示如何建立原則，以授與 Alice 存取該群組中唯讀動作的權限。

```
permit(
  principal == PhotoFlash::User::"alice",
  action,
  resource
) when {
    action in PhotoFlash::Action::"ReadOnlyPhotoAccess"
```

```
};
```

此範例顯示如何建立策略，以允許所有主參與者對其具有owner屬性的資源執行任何動作。

```
permit(
  principal,
  action,
  resource
)
when {
  principal == resource.owner
};
```

此範例顯示如何建立策略，以便在主參與者的屬性符合資源的department屬性時，允許任何主參與者檢視任何資源。department

> **Note**
>
> 如果實體沒有在原則條件中提及的屬性，則當執行授權決策，且該實體的原則評估失敗時，將會忽略該原則。例如，任何沒有department屬性的主參與者都無法透過此原則授與對任何資源的存取權。

```
permit(
  principal,
  action == Action::"view",
  resource
)
when {
  principal.department == resource.owner.department
};
```

此範例顯示如何建立原則，如果主參與者是資源的主參與者，或者如果主參與者是資源的admins群組owner的一部分，則允許任何主參與者對資源執行任何動作。

```
permit(
  principal,
  action,
  resource,
)
```

```
when {
  principal == resource.owner |
  resource.admins.contains(principal)
};
```

拒絕存取

如果原則包含原則forbid的效果，則會限制權限，而不是授與權限。

> **Important**
>
> 在授權期間，如果同時強制執行permit和forbid政策，則優forbid先順序。

下列範例會使用在 Cedar 原則語言參考指南〈範例結構 PhotoFlash 描述〉一節中所述的假設應用程式中定義的屬性。

此範例顯示如何建立策略，拒絕使用者alicereadOnly對任何資源執行所有動作。

```
forbid (
  principal == User::"alice",
  action,
  resource
)
unless {
  action.readOnly
};
```

此範例顯示如何建立策略，以拒絕存取具有private屬性的所有資源，除非主參與者具有該資源的owner屬性。

```
forbid (
  principal,
  action,
  resource
)
when {
  resource.private
}
unless {
  principal == resource.owner
```

```
};
```

Amazon Verified Permissions Permissions

您可以在已驗證的權限中建立 Cedar 原則範本，以定義系統的存取控制規則。政策範本是 Cedar 政策，其中含有預留位置的`principal`,`resource`，或兩者兼而有之。原則範本可讓您定義一次原則，然後將其附加至多個主參與者和資源。策略範本的更新會反映在使用該範本的所有主參與者和資源中。如需詳細資訊，請參閱雪松政策在雪松政策語言參考指南。

我們建議您使用原則範本來建立可在整個應用程式中共用的原則。例如，您可以為編輯器建立原則範本，為使用該原則範本的主參與者和資源提供讀取、編輯和註解權限。

```
permit(
  principal == ?principal,
  action in [Action::"Read", Action::"Edit", Action::"Comment"],
  resource == ?resource
);
```

將主參與者指定為資源的編輯器時，您的應用程式可以使用範本實例化原則，以提供主參與者對資源執行讀取、編輯和註解動作的權限。

建立策略範本

AWS Management Console

建立策略範本

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的瀏覽窗格中，選擇 [原則範本]。
3. 選擇 [建立策略範本]。
4. 在 [詳細資料] 區段中，輸入原則範本說明。
5. 在 [原則範本內文] 區段中，使用預留位置，`?principal`並`?resource`允許根據此範本建立的策略自訂其授與的權限。您可以選擇 [格式]，以建議的間距和縮排格式化原則範本的語法。
6. 選擇 [建立策略範本]。

AWS CLI

建立策略範本

您可以使用CreatePolicyTemplate作業建立原則範本。下列範例會建立原則範本，其中包含主參與者的預留位置。

該文件template1.txt包含以下內容。

```
"VacationAccess"
permit(
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

```
$ aws verifiedpermissions create-policy-template \
    --description "Template for vacation picture access"
    --statement file://template1.txt
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-18T21:17:47.284268+00:00",
    "lastUpdatedDate": "2023-05-18T21:17:47.284268+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

建立範本連結原則

您可以建立範本連結的策略來連結至策略範本。範本連結的原則會與其原則範本保持連結。如果您變更政策範本中的政策聲明，任何與該範本連結的政策都會自動使用新的陳述式來處理從那一刻起所做的所有授權決定。

AWS Management Console

透過實例化原則範本來建立範本連結的原則

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 Policies (政策)。
3. 選擇 [建立原則]，然後選擇 [建立範本連結原則]。
4. 選擇要使用的策略範本旁邊的圓鈕，然後選擇下一步。

5. 輸入要用於此範本連結原則之特定實例的主參與者和資源。指定的值會顯示在 [原則陳述式] 預覽欄位中。

   > **Note**
   >
   > 「主參與者」和「資源」值的格式必須與靜態策略相同。例如，若要指定主參與者的AdminUsers群組，請鍵入Group::"AdminUsers"。如果您輸入AdminUsers，則會顯示驗證錯誤。

6. 選擇 [建立範本連結原則]。

   新的範本連結政策會顯示在 [原則] 下。

AWS CLI

透過實例化原則範本來建立範本連結的原則

您可以建立參照現有策略範本的範本連結策略，並為範本使用的任何預留位置指定值。

下列範例會建立使用具有下列陳述式之範本的範本連結原則：

```
permit(
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

它也會使用下列definition.txt檔案來提供definition參數的值：

```
{
    "templateLinked": {
        "policyTemplateId": "pt-4651be67-c128-4d22-8e67-9b068980c631",
        "principal": {
            "entityType": "User",
            "entityId": "alice"
        }
    }
}
```

輸出會顯示從範本取得的資源，以及從定義參數取得的主體

```
$ aws verifiedpermissions create-policy \
    --definition file://definition.txt
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-22T18:57:53.298278+00:00",
    "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATELINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "Photo"
    }
}
```

編輯策略範本

AWS Management Console

若要編輯您的政策範本

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的瀏覽窗格中，選擇 [原則範本]。主控台會顯示您在目前原則存放區中建立的所有原則範本。
3. 選擇原則範本旁邊的圓鈕，以顯示有關原則範本的詳細資訊，例如建立、更新原則範本的時間，以及原則範本內容的時間。
4. 選擇 [編輯] 以編輯您的政策範本。視需要更新「策略說明」和「策略內文」，然後選擇「更新策略範本」。
5. 您可以選擇策略樣板旁邊的圓鈕，然後選擇刪除來刪除策略範本。選擇確定以確認刪除原則範本。

AWS CLI

更新策略範本

您可以使用UpdatePolicy作業建立靜態政策。下列範例會以檔案中定義的新原則取代指定的原則主體，以更新指定的原則範本。

檔案內容template1.txt：

```
permit(
    principal in ?principal,
    action == Action::"view",
    resource in ?resource)
when {
    principal has department && principal.department == "research"
};
```

```
$ aws verifiedpermissions update-policy-template \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --description "My updated template description" \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-17T18:58:48.795411+00:00",
    "lastUpdatedDate": "2023-05-17T19:18:48.870209+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

已驗證權限範例原則存放區的範例範本連結原則

當您使用範例原則存放區方法在「已驗證的權限」中建立原則存放區時，系統會使用預先定義的原則、原則範本和您所選範例專案的結構描述來建立原則存放區。下列已驗證權限範本連結的原則範例可與範例原則存放區及其各自的原則、原則範本和結構描述搭配使用。

PhotoFlash範本連結政策範例

此範例顯示如何建立使用政策範本的範本連結政策，授予與個別使用者和相片的非私人共用相片的有限存取權。

> **Note**
>
> Cedar 政策語言認為一個實體in本身就是實體。因此`principal in User::"Alice"`，相當於`principal == User::"Alice"`.

```
permit (
 principal in PhotoFlash::User::"Alice",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Photo::"VacationPhoto94.jpg"
 );
```

此範例顯示如何建立使用原則範本的範本連結政策，授予與個別使用者和相簿之非私人共用相片的有限存取權。

```
permit (
 principal in PhotoFlash::User::"Alice",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Album::"Italy2023"
 );
```

此範例顯示如何建立使用政策範本的範本連結政策，授予與朋友群組和個別相片的非私人共享相片的有限存取權。

```
permit (
 principal in PhotoFlash::FriendGroup::"Jane::MySchoolFriends",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Photo::"VacationPhoto94.jpg"
 );
```

此範例顯示如何建立使用政策範本的範本連結政策，授予與朋友群組和相簿之非私人共享相片的有限存取權。

```
permit (
 principal in PhotoFlash::FriendGroup::"Jane::MySchoolFriends",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Album::"Italy2023"
 );
```

此範例顯示如何建立使用政策範本的範本連結政策，授予與朋友群組和個別相片的非私人共享相片的完整存取權。

```
permit (
 principal in PhotoFlash::UserGroup::"Jane::MySchoolFriends",
 action in PhotoFlash::Action::"SharePhotoFullAccess",
 resource in PhotoFlash::Photo::"VacationPhoto94.jpg"
 );
```

此範例顯示如何建立使用策略範本封鎖帳號的使用者的範本連結策略。

```
forbid(
 principal == PhotoFlash::User::"Bob",
 action,
 resource in PhotoFlash::Account::"Alice-account"
 );
```

DigitalPetStore

範 DigitalPetStore 例原則存放區不包含任何原則範本。您可以在建立DigitalPetStore範例原則存放區之後，在左側的導覽窗格中選擇 [原則]，以檢視原則存放區所包含的原則。

TinyToDo 範本連結政策範例

此範例顯示如何建立範本連結的原則，該原則範本使用可為個別使用者和工作清單提供檢視者存取權限的原則。

```
permit (
    principal == TinyTodo::User::"https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_h2aKCU1ts|5ae0c4b1-6de8-4dff-b52e-158188686f31|bob",
    action in [TinyTodo::Action::"ReadList", TinyTodo::Action::"ListTasks"],
    resource == TinyTodo::List::"1"
);
```

此範例顯示如何建立範本連結的原則，該原則範本使用該原則範本為個別使用者和工作清單提供編輯器存取權。

```
permit (
    principal == TinyTodo::User::"https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_h2aKCU1ts|5ae0c4b1-6de8-4dff-b52e-158188686f31|bob",
```

```
    action in [
        TinyTodo::Action::"ReadList",
        TinyTodo::Action::"UpdateList",
        TinyTodo::Action::"ListTasks",
        TinyTodo::Action::"CreateTask",
        TinyTodo::Action::"UpdateTask",
        TinyTodo::Action::"DeleteTask"
    ],
    resource == TinyTodo::List::"1"
);
```

透過身分供應商使用 Amazon 驗證許可

身分識別來源是 Amazon 驗證許可中外部身分識別提供者 (IdP) 的表示形式。身分識別來源提供使用者的資訊，該使用者透過與您的原則存放區具有信任關係的 IdP 進行驗證。當您的應用程式使用身分識別來源的 Token 發出授權要求時，您的原則存放區可以根據使用者屬性和存取權限做出授權決定。已驗證的權限身分識別來源透過直接連線至中央身分識別存放區和驗證服務來改善授權。

您可以使用具有已驗證權限的 OpenID Connect（OIDC）身份提供程序（IdPs）。您的應用程式可以使用 OIDC 身分 (ID) 產生授權要求，或存取 JSON 網頁權杖 (JWT)。透過 ID Token，「已驗證的權限」會將使用者 ID 和屬性宣告讀取為屬性型存取控制 (ABAC) 的主體。使用存取權杖，「已驗證的權限」會將使用者 ID 讀取為主體，以及其他宣告作為內容 使用這兩種 Token 類型，您可以將宣告對應`groups`至主體群組，並建立評估角色型存取控制 (RBAC) 的原則。

您可以將 Amazon Cognito 使用者集區或自訂的 OpenID Connect (OIDC) IdP 新增為您的身分識別來源。

主題

- 使用 Amazon Cognito 身分來源
- 使用 OIDC 身分識別來源
- 用戶端和受眾驗證
- JWT 的用戶端授權
- 建立 Amazon 驗證的許可身分來源
- 編輯 Amazon 驗證的許可身分來源
- 在結構描述和原則中使用身分識別來源

使用 Amazon Cognito 身分來源

經過驗證的許可與 Amazon Cognito 使用者集區密切合作。Amazon Cognito JWT 具有可預測的結構。驗證的權限可識別此結構，並從它包含的信息中獲得最大的好處。例如，您可以使用 ID 令牌或訪問令牌實現基於角色的訪問控制（RBAC）授權模型。

新的 Amazon Cognito 使用者集區身分來源需要下列資訊：

- 的 AWS 區域.
- 使用者集區 ID。

- 例如，您要與身分識別來源建立關聯的使用者實體類型`MyCorp::User`。
- 例如，您要與身分識別來源產生關聯的群組實體類型`MyCorp::UserGroup`。
- (選擇性) 您要授權對原則存放區發出要求的使用者集區中的用戶端 ID。

由於已驗證許可僅適用於相同的 Amazon Cognito 使用者集區 AWS 帳戶，因此您無法在其他帳戶中指定身分來源。[已驗證的權限] 會將實體前置詞 (您必須在針對使用者集區原則執行動作的原則中參照的身分識別碼-來源識別碼) 設定為使用者集區的 ID，例如。`us-west-2_EXAMPLE`

用戶池令牌聲明可以包含屬性，範圍，組，客戶端 ID 和自定義數據。Amazon Cognito JWT 能夠包含各種資訊，這些資訊可以在已驗證的許可中為授權決策做出貢獻。其中包含：

1. 具有`cognito:`前綴的用戶名和組聲明
2. 自訂使用者屬性 `custom: prefix`
3. 在運行時添加自定義聲明
4. OIDC 標準索賠，例如和 `sub email`

我們詳細介紹了這些聲明，以及如何在「已驗證權限」策略中對其進行管理，在中在結構描述和原則中使用身分識別來源。

> **Important**
>
> 雖然您可以在 Amazon Cognito 權杖到期前撤銷，但 JWT 被視為無狀態資源，且具有簽章和有效性。符合 JSON Web 令牌 RFC 7519 的服務預計將遠程驗證令牌，並且不需要向發行者驗證令牌。這意味著可以根據已撤銷或發行給稍後刪除的用戶的令牌來授予訪問權限已驗證的權限。為了減輕這種風險，我們建議您以最短的有效期限創建令牌，並在要刪除授權以繼續用戶會話時撤消刷新令牌。

已驗證權限中使用者集區身分識別來源的 Cedar 原則會針對包含字母數字和底線 (_) 以外的字元的宣告名稱使用特殊語法。這包括包含`:`字符（如`cognito:username`和）的用戶池前綴聲明`custom:department`。若要撰寫參照`cognito:username`或`custom:department`宣告的原則條件，請將它們分別寫為`principal["cognito:username"]`和`principal["custom:department"]`。

> **Note**
>
> 如果令牌包含帶有`cognito:`或`custom:`前綴的聲明以及帶有文字值的聲明名稱`custom`，`cognito`或者，具有的授權請求IsAuthorizedWithToken將失敗，並顯示`ValidationException`.

此範例顯示如何建立政策，以參考與主體相關聯的某些 Amazon Cognito 使用者集區宣告。

```
permit(
     principal == ExampleCo::User::"us-east-1_example|4fe90f4a-ref8d9-4033-
a750-4c8622d62fb6",
     action,
     resource == ExampleCo::Photo::"VacationPhoto94.jpg"
)
when {
     principal["cognito:username"]) == "alice" &&
     principal["custom:department"]) == "Finance"
};
```

如需有關對應宣告的詳細資訊，請參閱將 ID 令牌映射到模式。如需 Amazon Cognito 使用者授權的詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用 Amazon 驗證許可授權。

使用 OIDC 身分識別來源

您也可以將任何符合標準的 OpenID Connect (OIDC) IdP 設定為原則存放區的身分識別來源。OIDC 提供者與 Amazon Cognito 使用者集區類似：它們會產生 JWT 做為身分驗證的產品。若要新增 OIDC 提供者，您必須提供發行者 URL

新的 OIDC 身分識別來源需要下列資訊：

- 發行者網址。已驗證的權限必須能夠在此 URL 探索`.well-known/openid-configuration`端點。
- 您要在授權請求中使用的令牌類型。在這種情況下，您選擇了身份令牌。
- 例如，您要與身分識別來源建立關聯的使用者實體類型`MyCorp::User`。
- 例如，您要與身分識別來源產生關聯的群組實體類型`MyCorp::UserGroup`。
- 示例 ID 令牌或 ID 令牌中聲明的定義。

- 您要套用至使用者和群組實體 ID 的前置詞。在 CLI 和 API 中，您可以選擇此前置詞。例如，在您使用 [設定 API Gateway 和身分識別來源] 或 [引導式設定] 選項建立的原則存放區中，[已驗證的權限] 會指派簽發者名稱的前置詞減去`https://MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`。

[OIDC 身分識別來源的授權會使用與使用者集區身分識別來源相同的 API 作業：IsAuthorizedWithToken和 BatchIs AuthorizedWith Token。]

此範例顯示如何建立政策，以允許存取會計部門員工的年終報告、進行機密分類，以及不在衛星辦公室中。「已驗證的權限」會從主體 ID Token 中的宣告衍生出這些屬性。

```
permit(
     principal in MyCorp::UserGroup::"MyOIDCProvider|Accounting",
     action,
     resource in MyCorp::Folder::"YearEnd2024"
) when {
     principal.jobClassification == "Confidential" &&
     !(principal.location like "SatelliteOffice*")
};
```

用戶端和受眾驗證

當您將身分識別來源新增至原則存放區時，「已驗證的權限」會提供組態選項，以驗證 ID 和存取權杖是否如預期使用。此驗證會在處理`IsAuthorizedWithToken`和 `BatchIsAuthorizedWithToken` API 要求時進行。ID 和存取權杖之間以及 Amazon Cognito 和 OIDC 身分識別來源之間的行為有所不同。使用 Amazon Cognito 使用者集區提供者，已驗證的許可可可可可以同時驗證 ID 和存取權杖中的用戶端 ID。使用 OIDC 提供者，「已驗證的權限」可以驗證 ID Token 中的用戶端 ID，以及存取權杖中的對象。

用戶端識別碼是與提供者設定的 OAuth 或 OIDC 應用程式相關聯的識別碼。`1example23456789`對象是與目標應用程式的預定信賴憑證者或目標相關聯的 URL 路徑，例如`https://myapplication.example.com`。`aud`聲明並不總是與受眾有關。

已驗證的權限執行身分識別來源對象和用戶端驗證，如下

Amazon Cognito

Amazon Cognito ID 權杖具有包含[應用程式用戶端] ID 的`aud`宣告。訪問令牌有一個`client_id`聲明，其中還包含應用程序客戶端 ID。

當您在身分識別來源中輸入用戶端應用程式驗證的一或多個值時，「已驗證的權限」會將此應用程式用戶端 ID 清單與 ID Token aud 宣告或存取權杖client_id宣告進行比較。已驗證的許可不會驗證 Amazon Cognito 身分識別來源的重新聚會對象 URL。

OIDC

OIDC ID 令牌具有包含客戶端 ID 列表的aud聲明。訪問令牌具有包含令牌的受眾 URL 的aud聲明。訪問令牌還具有包含預期客戶端 ID 的client_id聲明。

您可以使用 OIDC 提供者輸入一或多個值以進行「對象」驗證。當您選擇 ID Token 的 Token 類型時，「已驗證權限」會驗證用戶端 ID，並檢查aud宣告中的用戶端 ID 中至少有一個成員是否符合對象驗證值。

「已驗證權限」會驗證對象的存取權杖，並檢查aud宣告是否符合對象驗證值。此訪問令牌值主要來自聲明，但如果沒有aud聲明存在，則可以來自cid或client_idaud聲明。請向您的 IdP 查詢正確的受眾聲明和格式。

ID 令牌受眾驗證值的示例為1example23456789。

存取權杖受眾驗證值的範例為https://myapplication.example.com。

JWT 的用戶端授權

您可能想要在應用程式中處理 JSON Web Token，並將其宣告傳遞至「已驗證的權限」，而不使用政策存放區身分識別來源。您可以從 JSON Web 令牌（JWT）中提取實體屬性，並將其解析為已驗證的權限。

此範例顯示如何從 OIDC IDP.[1] 呼叫已驗證的權限

```
async function authorizeUsingJwtToken(jwtToken) {

    const payload = await verifier.verify(jwtToken);

    var principalEntity = {
        entityType: "PhotoFlash::User", // the application needs to fill in the
 relevant user type
        entityId: payload["sub"], // the application need to use the claim that
 represents the user-id
    };
    var resourceEntity = {
        entityType: "PhotoFlash::Photo", //the application needs to fill in the
 relevant resource type
```

```
        entityId: "jane_photo_123.jpg", // the application needs to fill in the
 relevant resource id
    };
    var action = {
        actionType: "PhotoFlash::Action", //the application needs to fill in the
 relevant action id
        actionId: "GetPhoto", //the application needs to fill in the relevant action
 type
    };
    var entities = {
        entityList: [],
    };
    entities.entityList.push(...getUserEntitiesFromToken(payload));
    var policyStoreId = "PSEXAMPLEabcdefg111111"; // set your own policy store id

    const authResult = await client
        .isAuthorized({
        policyStoreId: policyStoreId,
        principal: principalEntity,
        resource: resourceEntity,
        action: action,
        entities,
        })
        .promise();

    return authResult;

}

function getUserEntitiesFromToken(payload) {
  let attributes = {};
  let claimsNotPassedInEntities = ['aud', 'sub', 'exp', 'jti', 'iss'];
  Object.entries(payload).forEach(([key, value]) => {
    if (claimsNotPassedInEntities.includes(key)) {
        return;
    }
    if (Array.isArray(value)) {
      var attibuteItem = [];
      value.forEach((item) => {
        attibuteItem.push({
          string: item,
        });
      });
      attributes[key] = {
```

```
        set: attibuteItem,
      };
    } else if (typeof value === 'string') {
      attributes[key] = {
        string: value,
      }
    } else if (typeof value === 'bigint' || typeof value ==='number') {
        attributes[key] = {
            long: value,
          }
    } else if (typeof value === 'bigint' || typeof value ==='number') {
        attributes[key] = {
            long: value,
          }
    } else if (typeof value === 'boolean') {
        attributes[key] = {
            boolean: value,
       }
    }

  });

  let entityItem = {
    attributes: attributes,
    identifier: {
      entityType: "PhotoFlash::User",
      entityId: payload["sub"], // the application need to use the claim that
 represents the user-id
    }
  };
  return [entityItem];
}
```

[1] 此代碼示例使用 aws-jw-驗證庫來驗證由 OID 兼容的 JWT 簽名。 IdPs

建立 Amazon 驗證的許可身分來源

下列程序會將身分識別來源新增至現有的原則存放區。新增身分識別來源之後，您必須將屬性新增至結構描述。

當您在 [已驗證的權限] 主控台中建立新的原則存放區時，也可以建立身分識別來源。在此過程中，您可以將身分識別來源 Token 中的宣告自動匯入實體屬性。選擇 [引導式設定] 或 [使用 API Gateway 和身分識別提供者設定] 選項。這些選項也會建立初始策略。

> **Note**
>
> 在您建立原則存放區之前，身分識別來源無法在左側的導覽窗格中使用。您建立的身分識別來源與目前的原則存放區相關聯。

當您使用 [已驗證權限 API] AWS CLI 或 CreateIdentity [來源] 中的 [建立身分識別來源] 建立識別來源時，可以省略主體實體類型。但是，空白實體類型會建立實體類型為的識別來源`AWS::Cognito`。此實體名稱與原則存放區結構描述不相容。若要將 Amazon Cognito 身分與您的政策存放區結構描述整合，您必須將主體實體類型設定為支援的政策存放區實體。

主題

- Amazon Cognito 身份來源
- OIDC 身分識別來源

Amazon Cognito 身份來源

AWS Management Console

若要建立 Amazon Cognito 使用者集區身分識別來源

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [身分識別來源]。
3. 選擇建立身分識別來源。
4. 在 Cognito 使用者集區詳細資料中，選取 AWS 區域 並輸入身分識別來源的使用者集區 ID。
5. 在 [主參與者] 組態中，選擇識別來源的 [主參與者] 類型。來自已連線 Amazon Cognito 使用者集區的身分將對應至選取的主體類型。
6. 如果您要對應使用者集區**`cognito:groups`**宣告，請在 [群組設定] 中選取 [使用 Cognito 群組]。選擇作為主參與者類型父項的實體類型。
7. 在用戶端應用程式驗證中，選擇是否驗證用戶端應用程式 ID。

- 若要驗證用戶端應用程式 ID，請選擇「僅接受具有相符用戶端應用程式 ID 的 為要驗證的每個用戶端應用程式 ID 選擇新增用戶端應用程式 ID。若要移除已新增的用戶端應用程式 ID，請選擇用戶端應用程式 ID 旁邊的 [移除]。
- 如果您不想驗證用戶端應用程式 ID，請選擇不驗證用戶端應用程式 ID。

8. 選擇建立身分識別來源。
9. 在您可以參考從 Cedar 政策中的身分識別或存取權杖擷取的屬性之前，您必須先更新結構描述，讓 Cedar 知道您的身分識別來源所建立的主體類型。結構描述的新增項目必須包含您要在 Cedar 原則中參照的屬性。如需將 Amazon Cognito 權杖屬性對應至 Cedar 主要屬性的詳細資訊，請參閱在結構描述和原則中使用身分識別來源。

   當您建立 API 連結的原則存放區時，已驗證的權限會查詢您的使用者集區中的使用者屬性，並建立結構描述，其中您的主參與者類型會填入使用者集區屬性。

AWS CLI

若要建立 Amazon Cognito 使用者集區身分識別來源

您可以使用「來源」作業建立身分識別CreateIdentity來源。下列範例會建立可從 Amazon Cognito 使用者集區存取已驗證身分的身分識別來源。

下列`config.txt`檔案包含 Amazon Cognito 使用者集區的詳細資訊，以供命令中的--configuration參數使用。`create-identity-source`

```
{
    "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"],
        "groupConfiguration": {
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

命令：

```
$ aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
```

```
    --principal-entity-type "User" \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

在您可以參考從 Cedar 政策中的身分識別或存取權杖擷取的屬性之前，您必須先更新結構描述，讓 Cedar 知道您的身分識別來源所建立的主體類型。結構描述的新增項目必須包含您要在 Cedar 原則中參照的屬性。如需將 Amazon Cognito 權杖屬性對應至 Cedar 主要屬性的詳細資訊，請參閱在結構描述和原則中使用身分識別來源。

當您建立 API 連結的原則存放區時，已驗證的權限會查詢您的使用者集區中的使用者屬性，並建立結構描述，其中您的主參與者類型會填入使用者集區屬性。

如需有關針對已驗證的使用者使用 Amazon Cognito 存取權和身分權杖的詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用 Amazon 驗證許可授權。

OIDC 身分識別來源

AWS Management Console

若要建立 OpenID Connect (OIDC) 身分識別來源

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [身分識別來源]。
3. 選擇建立身分識別來源。
4. 選擇「外部 OIDC 提供者」。
5. 在發行者網址中，輸入 OIDC 發行者的網址。例如，這是提供授權伺服器、簽署金鑰和其他提供者相關資訊的服務端點`https://auth.example.com`。您的發行者 URL 必須在上託管 OIDC 探索文件。`/.well-known/openid-configuration`
6. 在權杖類型中，選擇您希望應用程式提交以進行授權的 OIDC JWT 類型。如需詳細資訊，請參閱 在結構描述和原則中使用身分識別來源。
7. 在 [使用者和群組宣告] 中，選擇身分識別來源的使用者實體和使用者宣告。使用者實體是您原則存放區中的一個實體，您想要參考來自 OIDC 提供者的使用者。U ser 聲明是一種聲

明sub，通常來自您的 ID 或訪問令牌，該 ID 或訪問令牌包含要評估的實體的唯一標識符。來自連線 OIDC IdP 的識別會對應至選取的主參與者類型。

8. 在 [使用者和群組宣告] 中，選擇身分識別來源的群組實體和群組宣告。群組實體是使用者實體的父系。群組宣告會對應至此實體。Group 宣告是一項宣告groups，通常來自您的 ID 或存取權杖，其中包含要評估之實體的字串、JSON 或空格分隔的使用者群組名稱字串。來自連線 OIDC IdP 的識別會對應至選取的主參與者類型。
9. 在對象驗證中，輸入您希望政策存放區在授權請求中接受的用戶端 ID 或對象 URL（如果有）。
10. 選擇建立身分識別來源。
11. 更新您的結構描述，讓 Cedar 知道身分識別來源所建立的主體類型。結構描述的新增項目必須包含您要在 Cedar 原則中參照的屬性。如需將 Amazon Cognito 權杖屬性對應至 Cedar 主要屬性的詳細資訊，請參閱在結構描述和原則中使用身分識別來源。

    當您建立 API 連結的原則存放區時，已驗證的權限會查詢您的使用者集區中的使用者屬性，並建立結構描述，其中您的主參與者類型會填入使用者集區屬性。

AWS CLI

若要建立 OIDC 身分識別來源

您可以使用「來源」作業建立身分識別CreateIdentity來源。下列範例會建立可從 Amazon Cognito 使用者集區存取已驗證身分的身分識別來源。

下列config.txt檔案包含 OIDC IdP 的詳細資訊，以供指令的--configuration參數使用。create-identity-source此範例會為 ID 權杖建立 OIDC 身分識別來源。

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth.example.com",
        "tokenSelection": {
                "identityTokenOnly": {
                        "clientIds":["1example23456789"],
                        "principalIdClaim": "sub"
                },
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
```

```
        }
    }
}
```

下列config.txt檔案包含 OIDC IdP 的詳細資訊，以供指令的--configuration參數使用。create-identity-source此範例會為存取權杖建立 OIDC 身分識別來源。

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth.example.com",
        "tokenSelection": {
                "accessTokenOnly": {
                        "audiences":["https://auth.example.com"],
                        "principalIdClaim": "sub"
                },
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

命令：

```
$ aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
    --principal-entity-type "User" \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

在您可以參考從 Cedar 政策中的身分識別或存取權杖擷取的屬性之前，您必須先更新結構描述，讓 Cedar 知道您的身分識別來源所建立的主體類型。結構描述的新增項目必須包含您要在 Cedar 原則中參照的屬性。如需將 Amazon Cognito 權杖屬性對應至 Cedar 主要屬性的詳細資訊，請參閱在結構描述和原則中使用身分識別來源。

當您建立 API 連結的原則存放區時，已驗證的權限會查詢您的使用者集區中的使用者屬性，並建立結構描述，其中您的主參與者類型會填入使用者集區屬性。

編輯 Amazon 驗證的許可身分來源

您可以在建立身分識別來源之後編輯它的某些參數。如果您的原則存放區結構描述與您的身分識別來源屬性相符，請注意，您必須個別更新結構描述，以反映您對身分識別來源所做的變更。

主題

- Amazon Cognito 用者集區身分來源
- OpenID Connect (OIDC) 身分識別來源

Amazon Cognito 用者集區身分來源

AWS Management Console

若要更新亞馬遜認知使用者集區身分識別來源

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [身分識別來源]。
3. 選擇要編輯的身分識別來源 ID。
4. 選擇編輯。
5. 在 Cognito 使用者集區詳細資料中，選取 AWS 區域 並輸入身分識別來源的使用者集區 ID。
6. 在主參與者詳細資料中，您可以更新識別來源的主體類型。來自已連線 Amazon Cognito 使用者集區的身分將對應至選取的主體類型。
7. 如果您要對應使用者集區**`cognito:groups`**宣告，請在 [群組設定] 中選取 [使用 Cognito 群組]。選擇作為主參與者類型父項的實體類型。
8. 在用戶端應用程式驗證中，選擇是否驗證用戶端應用程式 ID。
   - 若要驗證用戶端應用程式 ID，請選擇「僅接受具有相符用戶端應用程式 ID 的 為要驗證的每個用戶端應用程式 ID 選擇新增用戶端應用程式 ID。若要移除已新增的用戶端應用程式 ID，請選擇用戶端應用程式 ID 旁邊的 [移除]。
   - 如果您不想驗證用戶端應用程式 ID，請選擇不驗證用戶端應用程式 ID。
9. 選擇儲存變更。

10. 如果您變更識別來源的主體類型，則必須更新結構描述以正確反映更新的主參與者類型。

您可以選擇身分識別來源旁邊的圓鈕，然後選擇刪除身分識別來源來刪除身分識別來源。在文字方塊delete中輸入，然後選擇 [刪除身分識別來源] 以確認刪除身分識別來源。

AWS CLI

若要更新亞馬遜認知使用者集區身分識別來源

您可以使用「來源」作業更新身分識別UpdateIdentity來源。下列範例會更新指定的身分識別來源，以使用不同的 Amazon Cognito 使用者集區。

下列config.txt檔案包含 Amazon Cognito 使用者集區的詳細資訊，以供命令中的--configuration參數使用。create-identity-source

```
{
    "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"],
        "groupConfiguration": {
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

命令：

```
$ aws verifiedpermissions update-identity-source \
    --update-configuration file://config.txt \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

如果您變更識別來源的主參與者類型，則必須更新結構描述以正確反映更新的主參與者類型。

OpenID Connect (OIDC) 身分識別來源

AWS Management Console

更新 OIDC 身分識別來源

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [身分識別來源]。
3. 選擇要編輯的身分識別來源 ID。
4. 選擇編輯。
5. 在 OIDC 提供者詳細資訊中，視需要變更發行者 URL。
6. 在 [將 Token 宣告對應至結構描述屬性] 中，視需要變更使用者和群組宣告與原則存放區實體類型之間的關聯。變更實體類型之後，您必須更新原則和結構描述屬性，才能套用至新的實體類型。
7. 在對象驗證中，新增或移除您要強制執行的對象值。
8. 選擇儲存變更。

您可以選擇身分識別來源旁邊的圓鈕，然後選擇刪除身分識別來源來刪除身分識別來源。在文字方塊delete中輸入，然後選擇 [刪除身分識別來源] 以確認刪除身分識別來源。

AWS CLI

更新 OIDC 身分識別來源

您可以使用「來源」作業更新身分識別UpdateIdentity來源。下列範例會更新指定的身分識別來源，以使用不同的 OIDC 提供者。

下列config.txt檔案包含 Amazon Cognito 使用者集區的詳細資訊，以供命令中的--configuration參數使用。create-identity-source

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth2.example.com",
        "tokenSelection": {
                "identityTokenOnly": {
                        "clientIds":["2example10111213"],
                        "principalIdClaim": "sub"
                },
```

```
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

命令：

```
$ aws verifiedpermissions update-identity-source \
    --update-configuration file://config.txt \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

如果您變更識別來源的主參與者類型，則必須更新結構描述以正確反映更新的主參與者類型。

在結構描述和原則中使用身分識別來源

您可能會發現要將身分識別來源新增至原則存放區，並將 Provider 宣告對應至您的原則存放區結構描述。您可以自動執行此程序或手動更新結構描述。使用者指南的這一節包含下列資訊：

- 何時可以自動將屬性填入策略存放區結構描述
- 如何在您的已驗證許可政策中使用 Amazon Cognito 和 OIDC 令牌聲明
- 如何手動建置身分識別來源的結構描述

透過引導式設定具有身分識別來源的 API 連結原則存放區和原則存放區不需要將身分識別 (ID) Token 屬性手動對應至結構描述。您可以為使用者集區或 OIDC Token 中的屬性提供「已驗證的權限」，並建立填入使用者屬性的結構描述。在 ID 令牌授權中，已驗證的權限將聲明映射到主體實體的屬性。在下列情況下，您可能需要手動將 Amazon Cognito 權杖對應至結構描述：

- 您已從範例建立空白原則存放區或原則存放區。
- 您希望將訪問令牌的使用擴展到基於角色的訪問控制（RBAC）之外。

- 您可以使用已驗證的權限 REST API、 AWS SDK 或 AWS CDK.

若要使用 Amazon Cognito 或 OIDC 身分識別提供者 (IdP) 做為已驗證許可政策存放區中的身分來源，您的結構描述中必須具有提供者屬性。如果您建立原則存放區的方式會從 ID Token 中的提供者資訊自動填入結構描述，您就可以編寫政策了。如果您建立沒有身分識別來源結構描述的原則存放區，則必須將 Provider 屬性新增至結構描述。您的模式必須對應於提供程序令牌創建的實體IsAuthorizedWithToken或BatchIsAuthorizedWith令牌 API 請求。然後，您可以使用提供者 Token 中的屬性來撰寫政策。

如需有關針對已驗證的使用者使用 Amazon Cognito ID 和存取權杖的詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用 Amazon 驗證許可授權。

主題

- 有關綱要對應的注意事項
- 將 ID 令牌映射到模式
- 映射訪問令牌
- Amazon Cognito 冒號分隔聲明的替代符號

有關綱要對應的注意事項

標記類型之間的屬性對應不同

在訪問令牌授權中，已驗證的權限將聲明映射到上下文。在 ID 令牌授權中，已驗證的權限將聲明映射到主要屬性。對於您在 [已驗證權限] 主控台中建立的原則存放區，只有空白和範例原則存放區會讓您沒有身分識別來源，並要求您在結構描述中填入 ID Token 授權的使用者集區屬性。存取權杖授權是以具有群組成員資格宣告的角色型存取控制 (RBAC) 為基礎，且不會自動將其他宣告對應至原則儲存架構。

不需要身份識別來源屬性

當您在 [已驗證的權限] 主控台中建立身分識別來源時，不會將任何屬性標記為必要。這樣可以防止缺少聲明導致授權請求中的驗證錯誤。您可以視需要將屬性設定為必要，但所有授權要求中都必須出現這些屬性。

RBAC 不需要架構中的屬性

身分識別來源的結構描述取決於您在新增身分識別來源時所建立的實體關聯。身分識別來源會將一個宣告對應至使用者實體類型，並將一個宣告對應至群組實體類型。這些實體對應是身分識別來源組態的

核心。有了這項最低限度資訊，您就可以撰寫原則，在角色型存取控制 (RBAC) 模型中，針對使用者可能是其成員的特定使用者和特定群組執行授權動作。將 Token 聲明添加到模式中擴展了策略存儲區的授權範圍。來自 ID 令牌的用戶屬性具有有關可以促進基於屬性的訪問控制（ABAC）授權的用戶的信息。來自訪問令牌的上下文屬性具有諸如 OAuth 2.0 範圍之類的信息，這些信息可以提供來自提供程序的其他訪問控制信息，但需要進行其他模式

[已驗證權限] 主控台中的 [使用 API Gateway 和身分識別來源設定] 以及 [引導式設定] 選項會將 ID Token 宣告指派給結構描述。訪問令牌聲明並非如此。若要將非群組存取權杖宣告新增至結構描述，您必須在 JSON 模式下編輯結構定義，並新增 CommonType 屬性。如需詳細資訊，請參閱 映射訪問令牌。

OIDC 群組聲明支援多種格式

新增 OIDC 提供者時，您可以選擇要對應至原則存放區中使用者群組成員資格的 ID 或存取權杖中的群組宣告名稱。已驗證的權限可識別以下列格式的群組宣告：

1. 字符串不帶空格："groups": "MyGroup"
2. 以空格分隔的清單："groups": "MyGroup1 MyGroup2 MyGroup3"。每個字串都是一個群組。
3. JSON (逗號分隔) 清單："groups": ["MyGroup1", "MyGroup2", "MyGroup3"]

> **Note**
>
> 驗證權限將空格分隔的組聲明中的每個字符串解釋為一個單獨的組。若要將具有空格字元的群組名稱解譯為單一群組，請取代或移除宣告中的空格。例如，格式化名My Group為的群組MyGroup。

選擇權杖類型

您的政策存放區與身分識別來源搭配使用的方式取決於身分識別來源組態中的一項重要決定：您是要處理 ID 還是存取權杖。使用 Amazon Cognito 身分識別供應商時，您可以在建立 API 連結政策存放區時選擇權杖類型。當您建立 API 連結的原則存放區時，您必須選擇是否要設定 ID 或存取權杖的授權。此資訊會影響已驗證權限套用至原則存放區的結構描述屬性，以及 API Gateway API 的 Lambda 授權器語法。使用 OIDC 提供者時，您必須在新增身分識別來源時選擇權杖類型。您可以選擇 ID 或存取權杖，而您的選擇將未選擇的權杖類型排除在您的政策存放區中處理。特別是如果您希望從「已驗證權限」控制台中的 ID 令牌聲明自動映射到屬性中受益，請在創建身份源之前儘早決定要處理的令牌類型。變更 Token 類型需要大量的努力來重構您的政策和結構描述。下列主題說明 ID 和存取權杖搭配原則存放區的使用方式。

Cedar 解析器需要一些字符的括號

策略通常會以類似的格式引用結構描述屬性principal.username。在大多數非字母數字字符（例如 :.，或/）可能出現在令牌聲明名稱中的情況下，「已驗證權限」無法解析principal.cognito:groups或context.ip-address之類的條件值。您必須改為使用格式或格式的括號標記法來格principal["cognito:username"]式context["ip-address"]化這些條件。底線字元_是宣告名稱中的有效字元，也是此需求的唯一非英數字元例外。

此類型主體屬性的部分範例結構描述如下所示：

```
"User": {
   "shape": {
      "type": "Record",
      "attributes": {
         "cognito:username": {
            "type": "String",
            "required": true
         },
         "custom:employmentStoreCode": {
            "type": "String",
            "required": true,
         },
         "email": {
            "type": "String",
            "required": false
         }
      }
   }
}
```

此類型的內容屬性的部分範例結構描述如下所示：

```
"GetOrder": {
   "memberOf": [],
   "appliesTo": {
      "resourceTypes": [
         "Order"
      ],
      "context": {
         "type": "Record",
         "attributes": {
            "ip-address": {
```

```
                "required": false,
                "type": "String"
            }
    }
    },
       "principalTypes": [
          "User"
       ]
    }
}
```

將根據此結構描述驗證之屬性的範例原則如下所示：

```
permit (
    principal in MyCorp::UserGroup::"us-west-2_EXAMPLE|MyUserGroup",
    action,
    resource
) when {
    principal["cognito:username"] == "alice" &&
    principal["custom:employmentStoreCode"] == "petstore-dallas" &&
    principal has email && principal.email == "alice@example.com" &&
    context["ip-address"] like "192.0.2.*"
};
```

將 ID 令牌映射到模式

驗證權限處理 ID 令牌聲明作為用戶的屬性：他們的姓名和標題，他們的組成員資格，他們的聯繫信息。ID 令牌在基於屬性的訪問控制（ABAC）授權模型中最有用。當您希望「已驗證權限」根據提出請求的人員分析資源存取時，請為身分識別來源選擇 ID Token。

Amazon Cognito ID 令牌

Amazon Cognito ID 令牌適用於大多數 OIDC 重新派對程式庫。他們通過其他索賠擴展了 OIDC 的功能。您的應用程式可以透過 Amazon Cognito 使用者集區身份驗證 API 操作或使用者集區託管的 UI 來驗證使用者。如需詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用 API 和端點。

Amazon Cognito ID 令牌中有用的聲明

cognito:username 和 *preferred_username*

使用者使用者名稱的變體。

sub

使用者的唯一使用者識別碼 (UUID)

具有*custom:*前綴的索賠

自定義用戶池屬性的前綴，例如custom:employmentStoreCode。

標準索償

標準 OIDC 聲明類似email和。phone_number有關更多信息，請參閱 OpenID Connect 核心 1.0 中的標準聲明，其中包含勘誤集 2。

cognito:groups

使用者的群組成員資格。在以角色為基礎的存取控制 (RBAC) 為基礎的授權模型中，此宣告會呈現您可以在原則中評估的角色。

短暫索賠

聲明不是用戶的屬性，但在運行時由用戶池前令牌生成 Lambda 觸發器添加。暫時性聲明類似於標準聲明，但不在標準範圍內，例如tenant或department。

在參考具有:分隔符號之 Amazon Cognito 屬性的政策中，請以格式principal["*cognito:username*"]參考屬性。角色宣告cognito:groups是此規則的例外狀況。已驗證的權限會將此宣告的內容對應至使用者實體的父項實體。

如需 Amazon Cognito 使用者集區 ID 權杖結構的詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用 ID 權杖。

以下示例 ID 令牌具有四種屬性類型中的每一種。其中包括 Amazon Cognito 專屬聲明cognito:username、自訂宣告custom:employmentStoreCode、標準聲明和暫時性索賠email。tenant

```
{
    "sub": "91eb4550-XXX",
    "cognito:groups": [
        "Store-Owner-Role",
        "Customer"
    ],
    "email_verified": true,
    "clearance": "confidential",
    "iss": "https://cognito-idp.us-east-2.amazonaws.com/us-east-2_EXAMPLE",
    "cognito:username": "alice",
```

```
    "custom:employmentStoreCode": "petstore-dallas",
    "origin_jti": "5b9f50a3-05da-454a-8b99-b79c2349de77",
    "aud": "1example23456789",
    "event_id": "0ed5ad5c-7182-4ecf-XXX",
    "token_use": "id",
    "auth_time": 1687885407,
    "department": "engineering",
    "exp": 1687889006,
    "iat": 1687885407,
    "tenant": "x11app-tenant-1",
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "email": "alice@example.com"
}
```

使用 Amazon Cognito 使用者集區建立身分識別來源時，您可以指定在授權請求中產生的已驗證許可的主體實體類型。`IsAuthorizedWithToken`然後，您的原則可以測試該主體的屬性，作為評估該請求的一部分。您的結構描述會定義身分識別來源的主體類型和屬性，然後您可以在 Cedar 原則中參照這些屬性。

您也可以指定要從 ID 權杖群組宣告衍生的群組實體類型。在授權要求中，已驗證的權限會將群組宣告的每個成員對應至該群組實體類型。在原則中，您可以將該群組實體參照為主參與者。

下列範例顯示如何反映 [已驗證權限] 結構描述中範例識別 Token 的屬性。如需編輯資料架構的更多資訊，請參閱〈〉在 JSON 模式下編輯結構定義。如果您的身分識別來源組態指定了主參與者類型`User`，則您可以包含類似下列範例的項目，讓 Cedar 可以使用這些屬性。

```
"User": {
   "shape": {
      "type": "Record",
      "attributes": {
         "cognito:username": {
            "type": "String",
            "required": false
         },
         "custom:employmentStoreCode": {
            "type": "String",
            "required": false
         },
         "email": {
            "type": "String"
         },
         "tenant": {
```

```
            "type": "String",
            "required": true
         }
      }
   }
}
```

更新結構描述以反映 Amazon Cognito 屬性後，您可以建立參考這些屬性的政策。

```
permit (
    principal in MyCorp::UserGroup::"us-west-2_EXAMPLE|MyUserGroup",
    action,
    resource
) when {
    principal["cognito:username"] == "alice" &&
    principal["custom:employmentStoreCode"] == "petstore-dallas" &&
    principal.tenant == "x11app-tenant-1" &&
    principal has email && principal.email == "alice@example.com"
};
```

OIDC 識別碼權杖

使用來自 OIDC 提供者的 ID 權杖與使用 Amazon Cognito ID 權杖大致相同。不同之處在於索賠。您的 IdP 可能會呈現標準 OIDC 屬性，或具有自訂結構描述。當您在 [已驗證權限] 主控台中建立新的原則存放區時，您可以使用範例 ID Token 新增 OIDC 身分識別來源，或者您可以手動將權杖宣告對應至使用者屬性。由於已驗證的權限不知道 IdP 的屬性結構描述，因此您必須提供此資訊。

如需詳細資訊，請參閱 建立驗證權限原則存放區。

以下是具有 OIDC 身分識別來源之原則存放區的範例結構描述。

```
"User": {
   "shape": {
      "type": "Record",
      "attributes": {
         "email": {
            "type": "String"
         },
         "email_verified": {
            "type": "Boolean"
         },
         "name": {
```

```
            "type": "String",
            "required": true
         },
         "phone_number": {
            "type": "String"
         },
         "phone_number_verified": {
            "type": "Boolean"
         }
      }
   }
}
```

下列政策適用於 OIDC 提供者中的群組成員。

```
permit (
    principal in MyCorp::UserGroup::"MyOIDCProvider|MyUserGroup",
    action,
    resource
) when {
    principal.email_verified == true && principal.email == "alice@example.com" &&
    principal.phone_number_verified == true && principal.phone_number like "+1206*"
};
```

映射訪問令牌

已驗證的權限處理除了組聲明作為動作的屬性或上下文屬性以外的訪問令牌聲明。除了群組成員資格外，IdP 中的存取權杖可能包含有關 API 存取的資訊。在使用基於角色的訪問控制（RBAC）的授權模型中，訪問令牌非常有用。依賴群組成員資格以外的存取權杖宣告的授權模型需要額外的工作架構組態。

映射 Amazon Cognito 問令牌

Amazon Cognito 訪問令牌具有可用於授權的聲明：

Amazon Cognito 訪問令牌中的有用聲明

`client_id`

OIDC 信賴方的用戶端應用程式識別碼。使用用戶端 ID，「已驗證的權限」可以驗證授權要求來自原則存放區的允許用戶端。在 machine-to-machine （M2M）授權中，請求系統使用客戶端密鑰授權請求，並提供客戶端 ID 和範圍作為授權證據。

scope

OAuth 2.0 範圍，表示令牌承載者的訪問權限。

cognito:groups

使用者的群組成員資格。在以角色為基礎的存取控制 (RBAC) 為基礎的授權模型中，此宣告會呈現您可以在原則中評估的角色。

短暫索賠

不是存取權限，但是由使用者集區前憑證產生 Lambda 觸發程序在執行階段新增的宣告。暫時性聲明類似於標準聲明，但不在標準範圍內，例如`tenant`或`department`。訪問令牌的自定義會為您的 AWS 賬單增加成本。

如需有關 Amazon Cognito 使用者集區存取權杖結構的詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用存取權杖。

傳遞至已驗證許可時，Amazon Cognito 存取權杖會對應至內容物件。訪問令牌的屬性可以使用引用`context.token.`*`attribute_name`*。以下示例訪問令牌包括`client_id`和`scope`聲明。

```
{
    "sub": "91eb4550-9091-708c-a7a6-9758ef8b6b1e",
    "cognito:groups": [
        "Store-Owner-Role",
        "Customer"
    ],
    "iss": "https://cognito-idp.us-east-2.amazonaws.com/us-east-2_EXAMPLE",
    "client_id": "1example23456789",
    "origin_jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "event_id": "bda909cb-3e29-4bb8-83e3-ce6808f49011",
    "token_use": "access",
    "scope": "MyAPI/mydata.write",
    "auth_time": 1688092966,
    "exp": 1688096566,
    "iat": 1688092966,
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN2222222",
    "username": "alice"
}
```

下列範例顯示如何反映 [已驗證權限] 結構描述中範例存取權杖的屬性。如需編輯資料架構的更多資訊，請參閱〈〉在 JSON 模式下編輯結構定義。

```
{
   "MyApplication": {
      "actions": {
         "Read": {
            "appliesTo": {
               "context": {
                  "type": "ReusedContext"
               },
               "resourceTypes": [
                  "Application"
               ],
               "principalTypes": [
                  "User"
               ]
            }
         }
      },
      ...
      ...
      "commonTypes": {
         "ReusedContext": {
            "attributes": {
               "token": {
                  "type": "Record",
                  "attributes": {
                     "scope": {
                        "type": "Set",
                        "element": {
                           "type": "String"
                        }
                     },
                     "client_id": {
                        "type": "String"
                     }
                  }
               }
            },
            "type": "Record"
         }
      }
   }
}
```

更新結構描述以反映 Amazon Cognito 屬性後，您可以建立參考這些屬性的政策。

```
permit(principal, action in [MyApplication::Action::"Read",
 MyApplication::Action::"GetStoreInventory"], resource)
when {
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
    context.token.scope.contains("MyAPI/mydata.write")
};
```

映射 OIDC 訪問令牌

來自外部 OIDC 提供者的大多數存取權杖都與 Amazon Cognito 存取權杖密切一致。傳遞至已驗證權限時，OIDC 存取權杖會對應至內容物件。訪問令牌的屬性可以使用引用`context.token.`*`attribute_name`*。以下示例 OIDC 訪問令牌包括示例基本聲明。

```
{
    "sub": "91eb4550-9091-708c-a7a6-9758ef8b6b1e",
    "groups": [
        "Store-Owner-Role",
        "Customer"
    ],
    "iss": "https://auth.example.com",
    "client_id": "1example23456789",
    "aud": "https://myapplication.example.com"
    "scope": "MyAPI-Read",
    "exp": 1688096566,
    "iat": 1688092966,
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN2222222",
    "username": "alice"
}
```

下列範例顯示如何反映 [已驗證權限] 結構描述中範例存取權杖的屬性。如需編輯資料架構的更多資訊，請參閱〈〉在 JSON 模式下編輯結構定義。

```
{
   "MyApplication": {
      "actions": {
         "Read": {
            "appliesTo": {
               "context": {
                  "type": "ReusedContext"
               },
```

```
                "resourceTypes": [
                   "Application"
                ],
                "principalTypes": [
                   "User"
                ]
             }
          }
       },
       ...
       ...
       "commonTypes": {
          "ReusedContext": {
             "attributes": {
                "token": {
                   "type": "Record",
                   "attributes": {
                      "scope": {
                         "type": "Set",
                         "element": {
                            "type": "String"
                         }
                      },
                      "client_id": {
                         "type": "String"
                      }
                   }
                }
             },
             "type": "Record"
          }
       }
    }
}
```

更新結構描述以反映 IdP 屬性之後，您可以建立參照屬性的策略。

```
permit(
    principal,
    action in [MyApplication::Action::"Read",
 MyApplication::Action::"GetStoreInventory"],
    resource
)
```

```
when {
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
    context.token.scope.contains("MyAPI-read")
};
```

Amazon Cognito 冒號分隔聲明的替代符號

啟動驗證許可時，Amazon Cognito 權杖的建議結構描述會宣告`cognito:groups`並`custom:store`轉換這些冒號分隔字串，以使用該字`.`元作為階層分隔符號。這種格式稱為點符號。例如，您的政策`principal.cognito.groups`中`cognito:groups`已成為對的引用。雖然您可以繼續使用此格式，但我們建議您使用括號標記來建置結構描述和原則。在這種格式中，您的政策`principal["cognito:groups"]`中將`cognito:groups`成為的引用。從「已驗證的權限」主控台為使用者集區 ID 權杖自動產生的結構描述使用括號標記法。

您可以在手動建立的結構描述和 Amazon Cognito 身分識別來源的政策中繼續使用點符號。對於任何其他類型的 OIDC IdP，您不能在綱要或原則中使用點符號或任何其他非英數字元。：

點符號的結構描述會將`:`字元的每個執行個體嵌套成`cognito`或`custom`初始片語的子項，如下列範例所示：

```
"CognitoUser": {
   "shape": {
      "type": "Record",
      "attributes": {
         "cognito": {
            "type": "Record",
            "required": true,
            "attributes": {
               "username": {
                  "type": "String",
                  "required": true
               }
            }
         },
         "custom": {
            "type": "Record",
            "required": true,
            "attributes": {
               "employmentStoreCode": {
                  "type": "String",
                  "required": true
               }
```

```
            }
         },
         "email": {
            "type": "String"
         },
         "tenant": {
            "type": "String",
            "required": true
         }
      }
   }
}
```

使用此格式的結構描述，您可以建立具有點標記法的原則，如下列範例所示：

```
permit(principal, action, resource)
when {
    principal.cognito.username == "alice" &&
    principal.custom.employmentStoreCode == "petstore-dallas" &&
    principal.tenant == "x11app-tenant-1" &&
    principal has email && principal.email == "alice@example.com"
};
```

為您的應用程式設計授權模型

當您準備在軟體應用程式中使用 Amazon 驗證許可服務時，第一步要立即編寫政策陳述式可能會很困難。這類似於在完全決定應用程式應該執行的動作之前撰寫 SQL 陳述式或 API 規格來開始開發應用程式的其他部分。相反地，您應該從使用者體驗開始，清楚瞭解使用者在應用程式 UI 中管理權限時應該看到的內容。然後，從這種經驗向後工作以達到實施方法。

當你做這項工作, 你會發現自己問的問題，如:

- 我的資源是什麼？ 他們彼此之間有關係嗎？ 例如，檔案是否位於資料夾中？
- 主參與者可以對每個資源執行哪些動作？
- 主體如何取得這些權限？
- 您是否希望使用者選擇預先定義的權限 (例如「管理員」、「操作員」或「ReadOnly」)，還是應該建立臨機操作政策聲明？ 或者兩者都是？
- 權限是否應該跨資源繼承，例如從父文件夾繼承權限的文件？
- 呈現用戶體驗需要哪些類型的查詢？ 例如，您是否需要列出主體可以存取的所有資源，以呈現該使用者的首頁？
- 使用者是否會不小心將自己鎖定在自己的資源之外？ 這是否需要避免？

本練習的最終結果稱為授權模型；它定義了主參與者、資源、動作，以及它們彼此之間的關聯性。產生此模型不需要 Cedar 或「已驗證權限」服務的獨特知識。取而代之的是，它首先是使用者體驗設計練習，就像其他任何其他練習一樣，並且可以顯示在成品中，例如介面模型、邏輯圖表，以及權限如何影響使用者在產品中看到的內容的整體描述中。Cedar 的設計靈活性足以滿足模型的客戶需求，而不是強迫模型不自然地彎曲以符合 Cedar 的實施。因此，對所需的用戶體驗有清晰的了解是達到最佳模型的最佳方法。

本節提供有關如何進行設計練習、需要注意的事項，以及成功使用已驗證權限的最佳作法集合的一般指引。

除了此處提供的準則外，請記得考慮 Cedar 政策語言參考指南中的最佳做法。

主題

- 沒有規範的「正確」模型
- 專注於 API 作業以外的資源
- 複合授權是正常的

- 多租戶考量
- 如果可能，請填入原則範圍
- 每個資源都存在於容器中
- 將主參與者與資源容器分開
- 不要在屬性中嵌入權限
- 偏好模型中的精細權限，並在使用者介面中彙總權限
- 考慮查詢授權的其他原因

沒有規範的「正確」模型

當你設計一個授權模型時，沒有單一的，唯一的正確答案。不同的應用程序可以有效地為類似的概念使用不同的授權模型，這是可以的。例如，考慮電腦檔案系統的表示方式。當您在類 Unix 的作業系統中建立檔案時，它不會自動繼承父資料夾的權限。相反地，在許多其他作業系統和大多數線上檔案共用服務中，檔案會繼承其父資料夾的權限。這兩個選項都是有效的，具體取決於應用程式最佳化的情況。

授權解決方案的正確性並非絕對性，但應該根據其提供客戶所需體驗的方式以及是否以預期的方式保護資源來檢視。如果您的授權模型提供了這一點，那麼它是成功的。

這就是為什麼以所需的使用者體驗開始設計是建立有效授權模型最有用的先決條件的原因。

專注於 API 作業以外的資源

在大多數面向消費者的應用程式中，使用權限是以應用程式支援的資源為主。例如，檔案共用應用程式可能會將權限表示為可對檔案或資料夾執行的動作。這是一個很好的，簡單的模型，抽象掉底層實現和後端 API 操作。

相比之下，其他類型的應用程式 (特別是 Web 服務) 通常會設計 API 作業本身的權限。例如，如果 Web 服務提供名為的 API`createThing()`，則授權模型可能會定義對應的權限，或`action`在 Cedar 中命名為`createThing`。這在許多情況下都有效，並且易於理解權限。若要呼叫`createThing`作業，您需要`createThing`動作權限。看起來很簡單，對吧？

您會發現，[已驗證權限] 主控台中的入門程序包含直接從 API 建置資源和動作的選項。這是一個有用的基準：您的策略存儲區和其授權的 API 之間的直接映射。

但是，這種以 API 為主的方法可能不如最佳化，因為 API 只是客戶真正嘗試保護的代理：基礎資料和資源。如果有多個 API 控制對相同資源的存取，則系統管理員可能難以理解這些資源的路徑，並據此管理存取。

例如，假設包含組織成員的使用者目錄。用戶可以組織成組，其中一個安全目標是禁止未經授權的方發現組成員資格。管理此使用者目錄的服務會提供兩個 API 作業：

- `listMembersOfGroup`
- `listGroupMembershipsForUser`

客戶可以使用其中一項作業來探索群組成員資格。因此，權限管理員必須記住，以協調對這兩項作業的存取。如果您以後選擇添加新的 API 操作來解決其他用例，例如以下內容，這將進一步複雜。

- `isUserInGroups`（用於快速測試用戶是否屬於一個或多個組的新 API）

從安全性的角度來看，此 API 會開啟第三個探索群組成員資格的路徑，從而中斷系統管理員精心設計的權限。

我們建議您忽略 API 語意，而是專注於基礎資料和資源及其關聯作業。將此方法套用至群組成員資格範例會產生抽象權限，例如`viewGroupMembership`，三個 API 作業中的每一個都必須參考這個權限。

API 名稱	許可
`listMembersOfGroup`	需要群組的`viewGroupMembership` 權限
`listGroupMembershipsForUser`	需要對用戶的`viewGroupMembership` 權限
`isUserInGroups`	需要對用戶的`viewGroupMembership` 權限

藉由定義這一個權限，系統管理員可以成功控制探索群組成員資格的存取權，無論現在還是永遠。作為權衡，每個 API 操作現在都必須記錄其可能需要的幾個權限，並且管理員在製作權限時必須查閱此文檔。必要時，這可能是一個有效的權衡，以滿足您的安全要求。

複合授權是正常的

當單一使用者活動 (例如按一下應用程式介面中的按鈕) 需要多個個別授權查詢來判斷是否允許該活動時，就會發生複合授權。例如，將檔案移至檔案系統中的新目錄可能需要三種不同的權限：從來源目錄刪除檔案的能力、將檔案新增至目標目錄的能力，以及可能觸碰檔案本身的能力 (視應用程式而定)。

如果您是設計授權模型的新手，則可能會認為每個授權決策都必須在單個授權查詢中解析。但這可能導致過於複雜的模型和令人費解的政策聲明。實際上，使用複合授權對於幫助您產生更簡單的授權模型很有用。精心設計的授權模型的一個衡量方法是，當您具有足夠分解的單個動作時，您的複合操作（例如移動文件）可以通過直觀的基元聚合來表示。

發生複合授權的另一種情況是，當多方參與授予權限的過程中。考慮一個組織目錄，其中使用者可以是群組的成員。一個簡單的方法是授予群組擁有者新增任何人的權限。但是，如果您希望用戶首先同意添加該怎麼辦？ 這引入了一個握手協議，其中用戶和組都必須同意成員資格。若要完成此操作，您可以引入另一個繫結至使用者的權限，並指定使用者是否可以新增至任何群組或特定群組。當呼叫者隨後嘗試將成員新增至群組時，應用程式必須強制執行權限的兩端：呼叫者具有將成員新增至指定群組的權限，並且要新增的個別使用者具有要新增的權限。何時N-方式握手存在，這是常見的觀察N複合授權查詢，以強制執行合約的每個部分。

如果您發現自己遇到了涉及多個資源的設計挑戰，而且還不清楚如何建立權限模型，則可能表明您擁有複合授權案例。在這種情況下，可以通過將操作分解為多個單獨的授權檢查來找到解決方案。

多租戶考量

您可能想要開發應用程式供多個客戶 (使用您應用程式的企業或租用戶) 使用，並將其與 Amazon 驗證許可整合。在開發授權模型之前，請先制定多租戶策略。您可以在一個共用原則存放區中管理客戶的政策，或為每個租用戶指派一個原則存放區。

1. 一個共用原則存放區

   所有租用戶共用單一原則存放區。應用程式會將所有授權要求傳送至共用原則存放區。

2. 每個租用戶原則儲存

   每個租用戶都有專用的原則存放區。應用程式會根據發出要求的承租人，查詢不同的原則存放區以進行授權決定。

這兩種策略都不會創建相對較大的授權請求，這些請求可能會對您的帳單產生影響。AWS那麼，如何，那麼，你應該如何設計你的方法呢？ 以下是可能有助於您的已驗證許可多租戶授權策略的常見條件。

租用戶原則隔離

將每個租用戶的政策與其他租戶隔離對於保護租戶數據很重要。當每個承租人都有自己的原則存放區時，他們每個人都有自己獨立的原則集。

授權流程

您可以使用每個租用戶原則存放區的要求中的原則存放區識別碼來識別發出授權要求的承租人。使用共用原則存放區時，所有要求都會使用相同的原則存放區 ID。

範本和結構描述管理

您的原則範本和原則存放區結構描述會在每個原則存放區中增加一定層級的設計和維護額外負荷。

全球政策管理

您可能想要將某些全域原則套用至每個承租人。管理全域原則的額外負荷層級會因共用和每個租用戶原則存放區模型而有所不同。

租戶下機

某些承租人會為您的結構描述和其案例特定的原則提供元素。當租用戶不再與您的組織處於使用中狀態，而您想要移除其資料時，工作量程度會隨其與其他租用戶隔離的程度而有所不同。

服務資源配額

已驗證的權限具有資源和要求率配額，可能會影響您的多租戶決策。如需配額的詳細資訊，請參閱資源配額。

比較共用原則存放區和每個租用戶原則存放

在共用和每個租用戶原則存放區模型中，每項考量都需要自己的時間和資源承諾。

考量事項	共用原則存放區中的工作量層級	每個租用戶原則存放區中的工作量
租用戶原則隔離	中等。Must include tenant identifiers in policies and authorization requests.	低。 Isolation is default behavior. Tenant-specific policies are inaccessible to other tenants.
授權流程	低。 All queries target one policy store.	中等。 Must maintain mappings between each tenant and their policy store ID.
範本和結構描述管理	低。 Must make one schema work for all tenants.	高。 Schemas and templates might be less complex

		individually, but changes require more coordination and complexity.
全球政策管理	低。 All policies are global and can be centrally updated.	高。 You must add global policies to each policy store in onboarding. Replicate global policy updates between many policy stores.
租戶下機	中等。 Must identify and delete only tenant-specific policies.	低。 Delete the policy store.
服務資源配額	高。 Tenants share resource quotas that affect policy stores like schema size, policy size per resource, and identity sources per policy store.	低。 Each tenant has dedicated resource quotas.

如何選擇

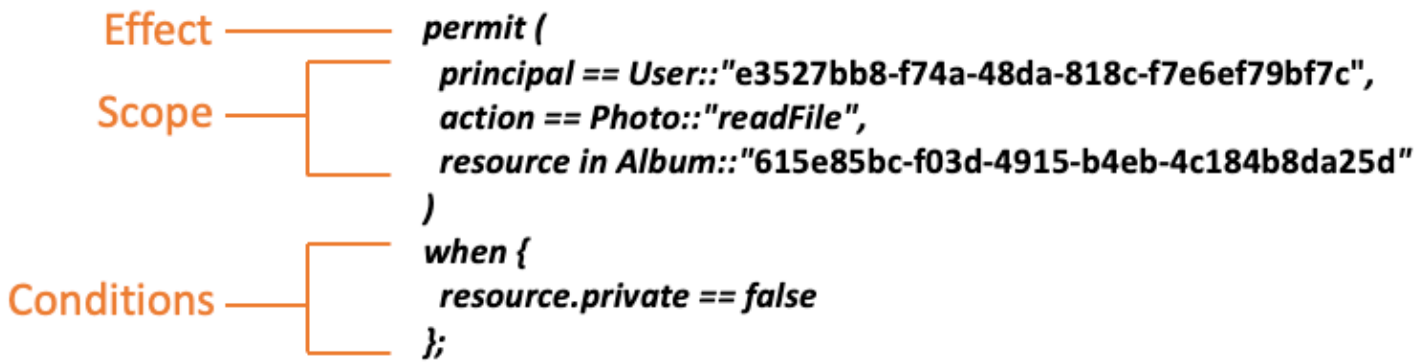
每個多租用戶應用程式都不同。在做出架構決策之前，請仔細比較這兩種方法及其考慮因素。

如果您的應用程式不需要承租人特定原則，而且使用單一身分識別來源，則所有租用戶的一個共用原則存放區可能是最有效的解決方案。這樣可以簡化授權流程和全域原則管理。使用一個共用原則存放區離線租用戶需要較少的工作，因為應用程式不需要刪除承租人特定原則。

但是，如果您的應用程式需要許多承租人特定原則，或使用多個身分識別來源，則每個租用戶原則存放區可能最有效。您可以使用將每個租用戶權限授與IAM每個原則存放區的原則來控制承租人原則的存取。離線承租人涉及刪除其原則存放區；在 shared-policy-store 環境中，您必須尋找並刪除承租人特定原則。

如果可能，請填入原則範圍

政策範圍是 Cedar 政策聲明之後的一部分`permit`或者`forbid`關鍵字和左括號之間。



我們建議您入入下列動作：principal和resource只要有可能。這可讓「已驗證的權限」建立原則的索引，以便更有效率地擷取，進而改善 如果您需要將相同的權限授與許多不同的主參與者或資源，建議您使用政策範本，並將其附加至每個主參與者和資源配對。

避免在中建立一個包含主參與者和資源清單的大型策略when子句。這樣做可能會導致您遇到可擴展性限制或操作挑戰。例如，若要從原則中的大型清單中新增或移除單一使用者，必須讀取整個原則、編輯清單、完整寫入新原則，以及在管理員覆寫另一個管理員的變更時處理並行錯誤。相反地，透過使用許多細微的權限，新增或移除使用者就像新增或移除套用至使用者的單一原則一樣簡單。

每個資源都存在於容器中

當您設計授權模型時，每個動作都必須與特定資源相關聯。使用諸如此類的動作viewFile，您可以應用它的資源很直觀：單個文件，或者可能是文件夾中的文件集合。但是，諸如之類的操作createFile不太直觀。建模建立檔案的功能時，它會套用到哪些資源？ 它不能是文件本身，因為該文件還不存在。

這是資源創建的廣義問題的一個例子。資源創建是一個引導問題。即使沒有資源存在，也必須有一種方法才能獲得創建資源的權限。解決方案是認識到每個資源都必須存在於某個容器中，並且它是容器本身充當權限的錨點。例如，如果系統中已存在資料夾，則建立檔案的能力可以建立為該資料夾的權限，因為這是具現化新資源所需權限的位置。

```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource == Folder::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

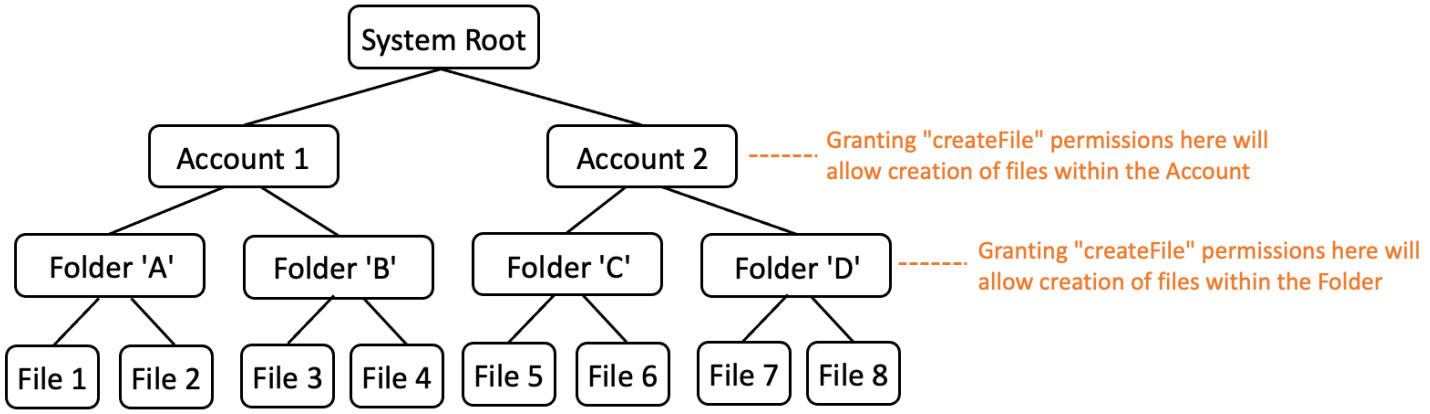
但是，如果沒有文件夾存在呢？ 也許這是一個全新的客戶帳戶，在尚未存在資源的應用程序中。在這種情況下，仍然有一個上下文可以通過詢問直觀地理解：客戶在哪裡可以創建新文件？ 您不希望他們

能夠在任何隨機客戶帳戶中創建文件。相反，有一個隱含的上下文：客戶自己的帳戶界限。因此，帳號本身代表用於建立資源的容器，並且可以在類似下列範例的原則中明確建模。

```
// Grants permission to create files within an account,
// or within any sub-folder inside the account.
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource in Account::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

但是，如果沒有帳戶怎麼辦？ 您可以選擇設計客戶註冊工作流程，以便在系統中建立新帳戶。如果是這樣，您將需要一個容器來容納程序可以在其中建立帳戶的最外層邊界。這個根級容器表示系統作為一個整體，可能被命名為「系統根」。但是，決定是否需要這樣做，以及如何命名它取決於您，應用程序所有者。

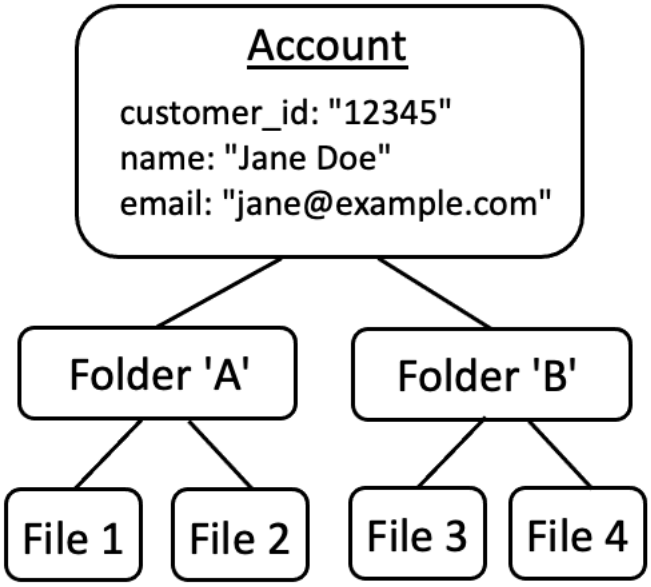
對於此範例應用程式，產生的容器階層會因此顯示如下：



這是一個範例階層。其他人也是有效的。要記住的是，資源創建始終發生在資源容器的上下文中。這些容器可以是隱含的，例如帳戶邊界，並且很容易忽略它們。在設計授權模型時，請務必注意這些隱含的假設，以便它們可以在授權模型中正式記錄和表示。

將主參與者與資源容器分開

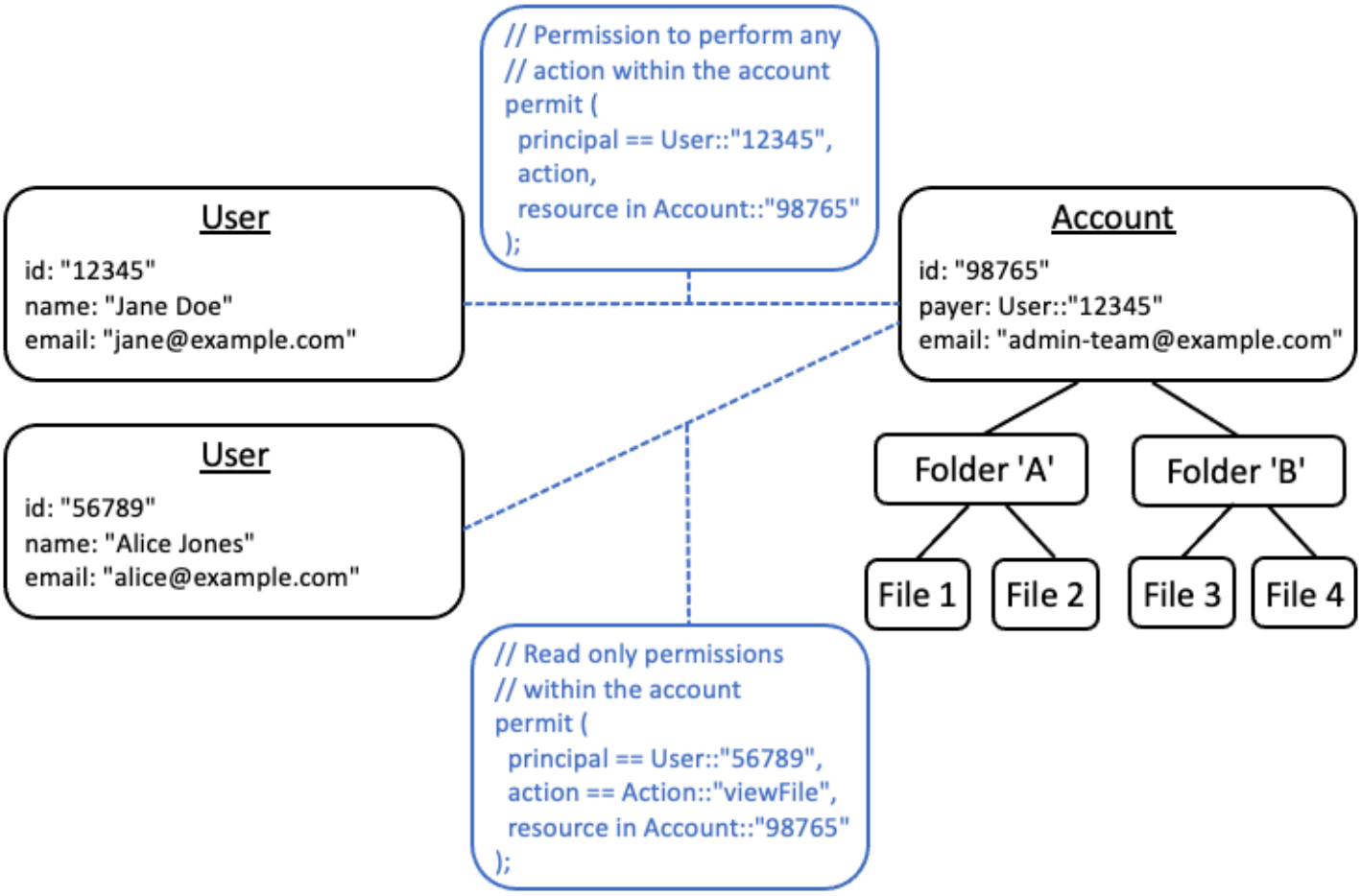
當您設計資源階層時，常見的傾向之一，特別是對於面向使用者的應用程式，就是使用客戶的使用者識別做為客戶帳戶內資源的容器。



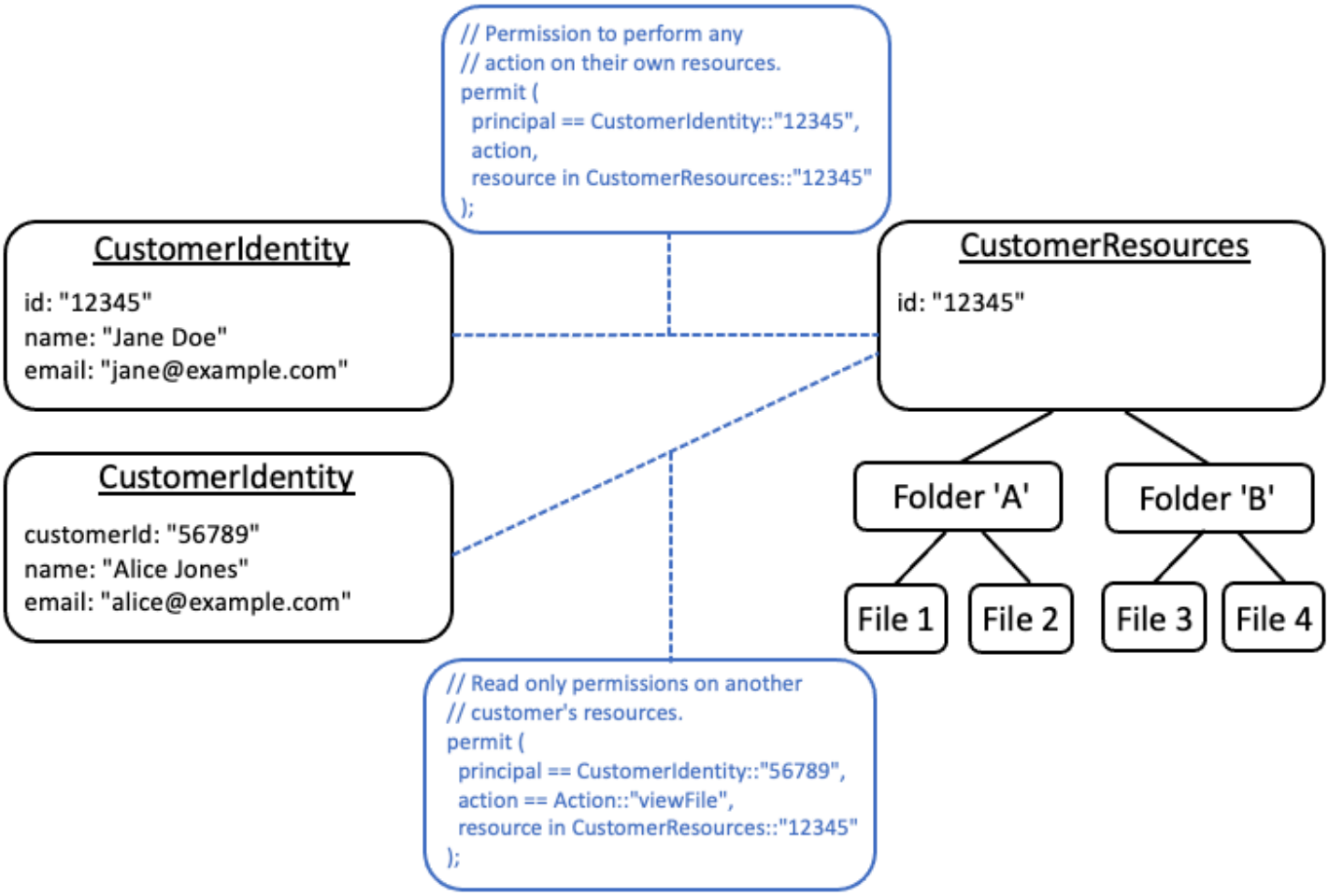
我們建議您將此策略視為反模式。這是因為在更豐富的應用程序中自然傾向於將訪問權限委派給其他用戶。例如，您可以選擇引入「家庭」帳戶，讓其他使用者可以共用帳號資源。同樣地，企業客戶有時候會想要將員工的多個成員指定為帳戶部分的操作員。您可能還需要將帳號的擁有權轉移給其他使用者，或將多個帳號的資源合併在一起。

使用使用者身分識別做為帳號的資源容器時，先前的案例會變得更加難以達成。更令人擔憂的是，如果其他人以這種方法授予對帳戶容器的訪問權限，他們可能無意中被授予修改用戶身份本身的訪問權限，例如更改 Jane 的電子郵件或登錄憑據。

因此，在可能的情況下，更具彈性的方法是將主參與者與資源容器分開，並使用諸如「管理員權限」或「擁有權」之類的概念來建立它們之間的連接模型。



如果您現有的應用程式無法採用此解耦模型，我們建議您在設計授權模型時盡可能地模仿它。例如，只有一個名為的概念的應用程序Customer封裝用戶身份，登錄憑據和他們擁有的資源，可以將其映射到包含一個邏輯實體的授權模型Customer Identity（包含姓名，電子郵件等）和一個單獨的邏輯實體Customer Resources或者Customer Account，作為其擁有之所有資源的父節點。兩個實體都可以共用相同Id，但與不同Type。



不要在屬性中嵌入權限

屬性最好用作輸入授權決定。不要使用屬性來表示權限本身，例如在用戶上聲明名為「許可的文件夾」的屬性：

```
// ANTI-PATTERN: comingling permissions into user attributes
{
    "id": "df82e4ad-949e-44cb-8acf-2d1acda71798",
    "name": "alice",
    "email": "alice@example.com",
    "permittedFolders": [
        "Folder::\"c943927f-d803-4f40-9a53-7740272cb969\"",
        "Folder::\"661817a9-d478-4096-943d-4ef1e082d19a\"",
        "Folder::\"b8ee140c-fa09-46c3-992e-099438930894\""
    ]
}
```

然後，在策略中使用屬性：

```
// ANTI-PATTERN
permit (
    principal,
    action == Action::"readFile",
    resource
)
when {
    resource in principal.permittedFolders
};
```

這種方法會將簡單授權模型 (其中特定主體可以存取特定資料夾) 轉換為具有隨附權衡的基於屬性的存取控制 (ABAC) 模型。這樣的權衡之一是，要快速確定誰擁有資源權限變得更加困難。在前面的範例中，若要判斷哪些人可以存取特定資料夾，必須重複查看每個使用者，以檢查其屬性中是否列出該資料夾，並在特別意識到有原則時授與存取權限。

這種方法的另一個風險是當權限打包在一個單一的內部時，擴展因素User名 如果用戶可以訪問很多東西，則其累計大小User記錄將增長，也許接近任何系統存儲數據的最大限制。

相反地，我們建議您使用多個個別原則來表示此案例，也許使用原則範本將重複的情況降到最低。

```
//BETTER PATTERN
permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
    action == Action::"readFile",
    resource in Folder::"c943927f-d803-4f40-9a53-7740272cb969"
);

permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
    action == Action::"readFile",
    resource in Folder::"661817a9-d478-4096-943d-4ef1e082d19a"
);

permit (
    principal == User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
    action == Action::"readFile",
    resource in Folder::"b8ee140c-fa09-46c3-992e-099438930894"
);
```

驗證的權限可以在授權評估期間有效地處理許多個別、精細的原則。隨著時間的推移，以這種方式建模事物更易於管理和可審核。

偏好模型中的精細權限，並在使用者介面中彙總權限

設計師後來經常後悔的一種策略是設計具有非常廣泛的行動的授權模型，例如Read和Write，後來意識到更細粒度的動作是必要的。客戶反饋更精細的存取控制，或者鼓勵最低權限權限的合規性和安全審核員驅動，可以驅動對更細微的細微性的需求。

如果未預先定義細微的權限，則可能需要進行複雜的轉換，才能將應用程式程式碼和原則陳述式修改為使用者更精細的權限。例如，先前針對課程粒度動作授權的應用程式程式碼將需要修改，才能使用精細的動作。此外，還需要更新原則以反映移轉：

```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    // action  ==  Action::"read",              -- coarse-grained permission --
 commented out
    action in [                        //        -- finer grained permissions
        Action::"listFolderContents",
        Action::"viewFile"
    ],
    resource in Account::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

為了避免這種昂貴的遷移，最好事先定義精細的權限。但是，如果您的最終使用者隨後被迫瞭解更多細微的權限，這可能會導致取捨，特別是如果大多數客戶對課程粒度控制感到滿意，例如Read和Write。為了獲得兩全其美，您可以將精細的權限分組到預先定義的集合中，例如Read和Write使用原則範本或動作群組等機制。透過使用這種方法，客戶只會看到課程粒度的權限。但在幕後，您已經通過將課程粒度權限建模為精細操作的集合，以確保應用程序面向未來。當客戶或稽核人員要求時，可能會公開精細的權限。

考慮查詢授權的其他原因

我們通常將授權檢查與用戶請求相關聯。檢查是判斷使用者是否具有執行該要求的權限的方法。但是，您也可以使用授權數據來影響應用程序界面的設計。例如，您可能想要顯示一個主畫面，其中只顯示一般使用者可以存取的資源清單。檢視資源的詳細資訊時，您可能希望介面只顯示使用者可以在該資源上執行的作業。

這些情況可能會將權衡引入授權模型中。例如，嚴重依賴基於歸因的訪問控制（ABAC）策略可能會使得快速回答「誰可以訪問什麼？」這個問題變得更加困難 這是因為回答該問題需要針對每個主參與者和資源檢查每個規則，以確定是否存在相符項目。因此，需要針對僅列出使用者可存取的資源進行最佳化的產品，可能會選擇使用以角色為基礎的存取控制 (RBAC) 模型。透過使用 RBAC，可以更容易地重複執行附加至使用者的所有原則，以判斷資源存取權。

測試台

「已驗證權限」測試工作台可讓您針對已驗證權限原則執行授權要求來測試和疑難排解這些原則。測試台會使用您指定的參數來決定原則存放區中的 Cedar 原則是否會授權要求。您可以在測試授權請求時在視覺模式和 JSON 模式之間切換。有關 Cedar 政策如何構建和評估的更多信息，請參閱 Cedar 政策語言參考指南中的 Cedar 中的基本原則構建。

> **Note**
>
> 當您使用已驗證的權限提出授權要求時，您可以在 [其他實體] 區段中提供主參與者和資源清單做為請求的一部分。但是，您無法包含有關操作的詳細信息。它們必須在結構描述中指定或從請求推斷出來。您無法在 [其他實體] 區段中放置動作。

有關測試台的視覺概述和演示，請參閱此視頻。

Visual mode

> **Note**
>
> 您必須在原則存放區中定義結構描述，才能使用測試工作台的視覺化模式。

若要在視覺化模式中測試原則

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [測試工作台]。
3. 選擇「視覺」模式。
4. 在「主參與者」段落中，從網要中的主參與者類型選擇「主參與者」採取動作。在文字方塊中輸入主參與者的識別碼。
5. (選擇性) 選擇「新增父項」，為指定的主參與者新增父項實體。若要移除已新增至主參與者的父項，請選擇父項名稱旁邊的「移除」(Remove)。
6. 為指定主參與者的每個屬性指定「屬性」值。測試台使用模擬授權請求中指定的屬性值。
7. 在「資源」段落中，選擇主參與者所作用的「資源」。在文字方塊中輸入資源的識別碼。

8. (選擇性) 選擇 [新增父系] 以新增指定資源的父項實體。若要移除已新增至資源的父項，請選擇父項名稱旁邊的 [移除]。
9. 為指定資源的每個屬性指定屬性值。測試台使用模擬授權請求中指定的屬性值。
10. 在「動作」段落中，從指定的主參與者與資源的有效動作清單中選擇主參與者正在執行的「動作」。
11. 為指定動作的每個屬性指定「屬性」值。測試台使用模擬授權請求中指定的屬性值。
12. (選擇性) 在「其他實體」區段中，選擇「新增實體」以新增要評估的授權決策的實體。
13. 從下拉式清單中選擇實體識別碼，然後輸入實體識別碼。
14. (選擇性) 選擇「新增父項」以新增指定實體的父項實體。若要移除已新增至實體的父項，請選擇父項名稱旁邊的 [移除]。
15. 為指定實體的每個屬性指定「屬性」值。測試台使用模擬授權請求中指定的屬性值。
16. 選擇 [確認] 以將實體新增至測試台。
17. 選擇 [執行授權要求]，以模擬原則存放區中 Cedar 原則的授權要求。測試工作台會顯示允許或拒絕要求的決定，以及有關已滿足的原則或評估期間遇到的錯誤的資訊。

JSON mode

若要在 JSON 模式下測試原則

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。選擇您的政策存放區。
2. 在左側的導覽窗格中，選擇 [測試工作台]。
3. 選擇 JSON 模式。
4. 在「要求詳細資訊」區段中，如果您已定義綱要，請從綱要中的主參與者類型中選擇「主參與者」採取動作。在文字方塊中輸入主參與者的識別碼。

   如果您沒有定義結構描述，請在主參與者採取動作文字方塊中輸入主參與者。
5. 如果您已定義結構描述，請從結構描述中的資源類型中選擇資源。在文字方塊中輸入資源的識別碼。

   如果您沒有定義結構描述，請在 [資源] 文字方塊中輸入資源。
6. 如果您已定義結構描述，請從指定主參與者和資源的有效動作清單中選擇「動作」(Action)。

   如果您尚未定義結構描述，請在「動作」文字方塊中輸入動作。
7. 在「內容」欄位中輸入要模擬之請求的內容。請求上下文是可用於授權決策的其他信息。

8. 在「實體」欄位中，輸入要針對授權決策評估的實體及其屬性的階層。
9. 選擇 [執行授權要求]，以模擬原則存放區中 Cedar 原則的授權要求。測試工作台會顯示允許或拒絕要求的決定，以及有關已滿足的原則或評估期間遇到的錯誤的資訊。

在 Amazon 驗證許可中實施授權

建立原則存放區、政策、範本、結構描述和授權模型之後，就可以開始使用 Amazon 驗證許可授權請求了。若要實作「已驗證的權限」授權，您必須將原則的組態 AWS 與應用程式的整合結合。若要將已驗證權限與您的應用程式整合，請新增 AWS SDK 並實作呼叫已驗證權限 API 的方法，並針對您的原則存放區產生授權決策。

具有已驗證權限的授權對於應用程序中的 UX 權限和 API 權限非常有用。

用戶體驗權

控制使用者對應用程式 UX 的存取。您可以允許使用者只檢視他們需要存取的確切表單、按鈕、圖形和其他資源。例如，當用戶登錄時，您可能想要確定他們的帳戶中是否可以看到「轉移資金」按鈕。您也可以控制使用者可以採取的動作。例如，在同一個銀行應用程序中，您可能想要確定是否允許您的用戶更改交易類別。

API 許可

控制使用者對資料的存取權。應用程序通常是分佈式系統的一部分，並從外部 API 引入信息。在「已驗證權限」允許顯示「轉移資金」按鈕的銀行應用程序示例中，當您的用戶啟動轉移時，必須做出更複雜的授權決策。「已驗證的權限」可以授權列出符合資格移轉目標的目的地帳戶的 API 請求，然後授權將轉移推送至其他帳戶的請求。

說明此內容的範例來自範例原則存放區。若要進行操作，請在您的測試環境中建立存放區範例原則存放區。DigitalPet

如需使用批次授權實作 UX 許可的端對端範例應用程式，請參閱 AWS Security Blog 上的大規模使用 Amazon 驗證許可進行大規模的精細授權。

用於授權的 API 操作

驗證權限 API 具有以下授權操作。

IsAuthorized

`IsAuthorized`API 操作是具有已驗證權限的授權請求的入口點。您必須提交主參與者、作業、資源、前後關聯及實體元素。已驗證的權限會根據您的原則存放區結構描述驗證請求中的實體。「已驗證的權限」接著會根據要求原則存放區中套用至要求中實體的所有原則評估您的要求。

IsAuthorizedWithToken

此IsAuthorizedWithToken作業會從 Amazon Cognito JSON 網路權杖 (JWT) 中的使用者資料產生授權要求。已驗證的許可可直接與 Amazon Cognito 搭配使用，做為政策存放區中的身分來源。「已驗證的權限」會將使用者 ID 或存取權杖中宣告的要求中的所有屬性填入主體。您可以從 Amazon Cognito 使用者集區中的使用者屬性或群組成員資格授權動作和資源。

您無法在IsAuthorizedWithToken要求中包含群組或使用者主體類型的相關資訊。您必須將所有主體資料填入您提供的 JWT 中。

BatchIs已授權

BatchIsAuthorized作業會針對單一 API 要求中的單一主體或資源處理多項授權決策。此作業會將要求分組成單一批次作業，以最大限度地減少配額使用量，並針對多達 30 個複雜巢狀動作傳回授權決策。透過單一資源的批次授權，您可以篩選使用者可對資源執行的動作。透過單一主參與者的批次授權，您可以篩選使用者可對其採取動作的資源。

BatchIsAuthorizedWith令牌

BatchIsAuthorizedWithToken作業會在一個 API 要求中針對單一主體處理多個授權決策。主體是由您的原則儲存區識別來源在 ID 或存取權杖中提供。此作業會將請求分組成單一批次作業，以最大限度地減少配額使用量，並針對每個動作和資源最多 30 個請求傳回授權決策。在您的政策中，您可以從 Amazon Cognito 使用者集區中的屬性或群組成員資格授權其存取權限。

就像使用一樣IsAuthorizedWithToken，您無法在要求中包含有關群組或使用者主體類型的BatchIsAuthorizedWithToken資訊。您必須將所有主體資料填入您提供的 JWT 中。

測試您的授權模型

若要瞭解部署應用程式時已驗證權限授權決策的影響，您可以在開發原則時評估原則，測試台並使用 HTTPS REST API 要求傳送至已驗證的權限。測試工作台是評估政策存放區中的授權要求和回應的工具。 AWS Management Console

驗證權限 REST API 是您開發中的下一個步驟，當您從概念理解轉移到應用程式設計。驗證權限 API 接受向區域服務端點提供IsAuthorizedIsAuthorizedWithToken、和BatchIs授權為已簽署 AWS API 請求的授權要求。若要測試您的授權模型，您可以使用任何 API 用戶端產生要求，並確認您的政策是否如預期傳回授權決策。

例如，您可以使用下列程序IsAuthorized在範例原則存放區中進行測試。

Test bench

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。從名稱為 [存放區] 的範例原則存放區建立原則DigitalPet存放區。
2. 在新原則存放區中選取「測試工作台」。
3. 從「已驗證權限 API」參考IsAuthorized中填入您的測試工作台請求。下列詳細資訊會複寫範例 4 中參照 DigitalPetStore 範例的條件。

   a. 設置愛麗絲為校長。對於主體採取行動，請選擇`DigitalPetStore::User`並輸入`Alice`。

   b. 設置愛麗絲作為客戶的角色。選擇「新增上層」`DigitalPetStore::Role`，選擇並輸入「客戶」。

   c. 將資源設定為順序「1234」。針對主參與者作用的資源，選擇`DigitalPetStore::Order`並輸入`1234`。

   d. 資`DigitalPetStore::Order`源需要一個`owner`屬性。將愛麗絲設置為訂單的擁有者。選擇`DigitalPetStore::User`並輸入 `Alice`

   e. 愛麗絲要求查看訂單。針對主體正在採取的動作，選擇`DigitalPetStore::Action::"GetOrder"`。
4. 選擇「執行授權請求」。在未修改的原則存放區中，此要求會產生`ALLOW`決定。請注意傳回決定的「滿意」原則。
5. 從左側導覽列選擇「策略」。檢閱具有「客戶角色-取得訂單」說明的靜態政策。
6. 請注意，已驗證的權限允許請求，因為主參與者是客戶角色，並且是資源的擁有者。

REST API

1. 在 https://console.aws.amazon.com/verifiedpermissions/ 開啟「已驗證的權限」主控台。從名稱為 [存放區] 的範例原則存放區建立原則DigitalPet存放區。
2. 記下新原則存放區的原則存放區 ID。
3. 從驗證權限 API 參考資料IsAuthorized中，複製參考DigitalPet存放區範例的範例 4 的要求內文。
4. 開啟 API 用戶端，並為您的政策存放區建立對區域服務端點的要求。如範例所示填入標頭。
5. 貼上範例要求主體，並將的值變更`policyStoreId`為您先前所記下的原則存放區識別碼。
6. 提交請求並複查結果。在預設的DigitalPet儲存區原則存放區中，此要求會傳回`ALLOW`決定。

您可以變更測試環境中的原則、結構描述和要求，以變更結果並產生更複雜的決策。

1. 以變更 [已驗證權限] 決定的方式變更要求。例如，將 Alice 的角色變更為`Employee`或將順序 1234 的`owner`屬性變更為`Bob`。
2. 以影響授權決策的方式變更政策。例如，使用「客戶角色-取得訂單」描述修改原則，以移除`User`必須是的擁有者的條件，`Resource`並修改`Bob`要求以檢視訂單。
3. 變更結構描述以允許政策做出更複雜的決策。更新要求實體，讓 Alice 能夠滿足新的需求。例如，編輯綱要以允`User`許成為`ActiveUsers`或的成員`InactiveUsers`。更新政策，以便只有作用中使用者可以檢視自己的訂單。更新要求實體，讓 Alice 是作用中或非作用中的使用者。

與應用程式和 AWS SDK 整合

若要在應用程式中實作 Amazon 驗證許可，您必須定義希望應用程式強制執行的政策和結構描述。有了您的授權模型並經過測試後，您的下一步就是從強制執行點開始產生 API 請求。若要這麼做，您必須設定應用程式邏輯來收集使用者資料，並將其填入授權要求。

應用程式如何使用已驗證權限授權要求

1. 收集有關當前用戶的信息。通常，用戶的詳細信息在經過身份驗證的會話的詳細信息中提供，例如 JWT 或 Web 會話 cookie。此使用者資料可能來自連結至您政策存放區的 Amazon Cognito 身分來源，或來自其他 OpenID Connect (OIDC) 提供者。
2. 收集使用者想要存取之資源的相關資訊。一般而言，當使用者選取要求您的應用程式載入新資產時，您的應用程式會收到有關資源的資訊。
3. 決定您的使用者想要採取的動作。
4. 針對使用者嘗試作業的主參與者、動作、資源和實體產生「已驗證權限」的授權要求。「已驗證的權限」會根據原則存放區中的原則評估要求，並傳回授權決策。
5. 您的應用程式會從已驗證的權限讀取允許或拒絕回應，並對使用者的要求強制執行決定。

已驗證的權限 API 作業內建於 AWS SDK 中。要在應用程序中包含「已驗證的權限」，請將所選語言的 AWS SDK 集成到應用程序包中。

若要深入了解並下載 AWS SDK，請參閱. Amazon Web Services

以下是各種 AWS SDK 中已驗證權限資源的文件連結。

- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto)
- AWS SDK for Ruby

以下 AWS SDK for JavaScript 範例來`IsAuthorized`源於使用 Amazon 驗證許可和 Amazon Cognito 簡化細粒度授權。

```
const authResult = await avp.isAuthorized({
    principal: 'User::"alice"',
    action: 'Action::"view"',
    resource: 'Photo::"VacationPhoto94.jpg"',
    // whenever our policy references attributes of the entity,
    // isAuthorized needs an entity argument that provides
    // those attributes
    entities: {
       entityList: [
         {
            "identifier": {
                "entityType": "User",
                "entityId": "alice"
            },
            "attributes": {
                "location": {
                    "String": "USA"
                }
            }
         }
       ]
    }
});
```

更多開發者資源

- Amazon 驗證許可研討
- Amazon 驗證許可-資源

- 使用 Amazon 驗證許可為 ASP.NET 核心應用程式實作自訂授權原則提供者
- 使用 Amazon 驗證許可為商業應用程式建立權益服務
- 使用 Amazon 驗證許可和 Amazon Cognito 簡化細粒度授權

添加上下文

前後關聯是與政策決策相關的資訊，但不是主參與者、動作或資源身分識別的一部分。您可能只想要允許來自一組來源 IP 位址的動作，或僅當您的使用者已使用 MFA 登入時才允許動作。您的應用程序可以訪問此上下文會話數據，並且必須將其填充到授權請求。驗證權限授權請求中的上下文數據必須在元素中進行 JSON 格式`contextMap`。

說明此內容的範例來自範例原則存放區。若要進行操作，請在您的測試環境中建立DigitalPetStore範例原則存放區。

下列前後關聯物件會根據範例DigitalPetStore原則存放區，針對應用程式宣告每種 Cedar 資料類型之一。

```
"context": {
   "contextMap": {
      "MfaAuthorized": {
         "boolean": true
      },
      "AccountCodes": {
         "set": [
            {
               "long": 111122223333
            },
            {
               "long": 444455556666
            },
            {
               "long": 123456789012
            }
         ]
      },
      "UserAgent": {
         "string": "My UserAgent 1.12"
      },
      "RequestedOrderCount": {
         "long": 4
      },
      "NetworkInfo": {
         "record": {
            "IPAddress": {
               "string": "192.0.2.178"
```

```
            },
            "Country": {
               "string": "United States of America"
            },
            "SSL": {
               "boolean": true
            }
         }
      },
      "approvedBy": {
         "entityIdentifier": {
            "entityId": "Bob",
            "entityType": "DigitalPetStore::User"
         }
      }
   }
}
```

授權上下文中的數據類型

Boolean

二進制`true`或`false`值。在此範例中，`true`for 的布林值`MfaAuthenticated`表示客戶在要求檢視其訂單之前已執行多重要素驗證。

設定

上下文元素的集合。集合成員可以是所有相同的類型，就像在這個例子中一樣，也可以是不同的類型，包括嵌套集合。在此範例中，客戶與 3 個不同的帳戶相關聯。

字串

字母、數字或符號的序列，以字`"`元括住。在此範例中，`UserAgent`字串代表客戶用來要求檢視其訂單的瀏覽器。

Long

整數。在範例中，`RequestedOrderCount`表示此請求屬於批次的一部分，該批次是由客戶要求檢視其過去四筆訂單而產生的。

記錄

屬性的集合。您必須在要求內容中宣告這些屬性。具有結構描述的原則存放區必須在結構描述中包含此實體和實體的屬性。在此範例中，`NetworkInfo`記錄包含有關使用者原始 IP、用戶端決定的 IP 地理位置，以及傳輸中加密的相關資訊。

EntityIdentifier

對在請求entities元素中聲明的實體和屬性的引用。在此範例中，使用者的訂單已由員工核准Bob。

若要在範例DigitalPetStore應用程式中測試此範例內容，您必須更新要求entities、原則存放區結構描述以及靜態政策，其說明為「客戶角色-取得訂單」。

修改 DigitalPetStore 以接受授權上下文

最初，不DigitalPetStore是一個非常複雜的策略存放區。它不包含任何預先設定的原則或前後關聯屬性來支援我們所呈現的前後關聯。若要使用此內容資訊評估授權要求範例，請對您的原則存放區和授權要求進行下列修改。

Schema

將下列更新套用至您的原則存放區結構描述，以支援新的前後關聯屬性。GetOrder在中更新actions，如下所示。

```
"GetOrder": {
   "memberOf": [],
   "appliesTo": {
      "resourceTypes": [
         "Order"
      ],
      "context": {
         "type": "Record",
         "attributes": {
            "UserAgent": {
               "required": true,
               "type": "String"
            },
            "approvedBy": {
               "name": "User",
               "required": true,
               "type": "Entity"
            },
            "AccountCodes": {
               "type": "Set",
               "required": true,
               "element": {
```

```
                "type": "Long"
             }
          },
          "RequestedOrderCount": {
             "type": "Long",
             "required": true
          },
          "MfaAuthorized": {
             "type": "Boolean",
             "required": true
          }
       }
    },
    "principalTypes": [
       "User"
    ]
 }
}
```

欲參照請求前後關聯NetworkInfo中指定的record資料類型，請在結構描述中建立 CommonType 建構，如下所示。commonType建構是一組共用的屬性，您可以套用至不同的圖元。

> **Note**
>
> 「已驗證的權限」視覺結構描述編輯器目前不支援commonType建構。當您將它們新增至結構描述時，您無法再以視覺化模式檢視結構定義。

```
"commonTypes": {
   "NetworkInfo": {
      "attributes": {
         "IPAddress": {
            "type": "String",
            "required": true
         },
         "SSL": {
            "required": true,
            "type": "Boolean"
         },
         "Country": {
            "required": true,
```

```
            "type": "String"
         }
      },
      "type": "Record"
   }
}
```

Policy

下列原則會設定必須由每個提供的前後關聯元素滿足的條件。它以現有靜態政策為基礎，其中包含「客戶角色-取得訂單」說明。此原則一開始只要求提出要求的主體是資源的擁有者。

```
permit (
    principal in DigitalPetStore::Role::"Customer",
    action in [DigitalPetStore::Action::"GetOrder"],
    resource
) when {
    principal == resource.owner &&
    context.MfaAuthorized == true &&
    context.UserAgent like "*My UserAgent*" &&
    context.RequestedOrderCount <= 4 &&
    context.AccountCodes.contains(111122223333) &&
    context.NetworkInfo.Country like "*United States*" &&
    context.NetworkInfo.SSL == true &&
    context.NetworkInfo.IPAddress like "192.0.2.*" &&
    context.approvedBy in DigitalPetStore::Role::"Employee"
};
```

我們現在要求擷取訂單的請求符合我們新增至要求的其他內容條件。

1. 使用者必須已使用 MFA 登入。
2. 使用者的網頁瀏覽器User-Agent必須包含字串My UserAgent。
3. 用戶必須要求查看 4 個或更少的訂單。
4. 用戶的帳戶代碼之一必須是111122223333。
5. 使用者的 IP 位址必須來自美國，必須位於加密工作階段，且其 IP 位址必須以開頭192.0.2.。
6. 員工必須已核准其訂單。在授權請求的entities元素中，我們將聲明具有Bob的作用的用戶Employee。

Request body

使用適當的結構描述和原則設定原則存放區之後，您可以將此授權要求提供給已驗證權限 API 作業IsAuthorized。請注意，entities區段包含的定義Bob，角色為的使用者Employee。

```
{
   "principal": {
      "entityType": "DigitalPetStore::User",
      "entityId": "Alice"
   },
   "action": {
      "actionType": "DigitalPetStore::Action",
      "actionId": "GetOrder"
   },
   "resource": {
      "entityType": "DigitalPetStore::Order",
      "entityId": "1234"
   },
   "context": {
     "contextMap": {
      "MfaAuthorized": {
       "boolean": true
      },
      "UserAgent": {
       "string": "My UserAgent 1.12"
      },
      "RequestedOrderCount":{
       "long": 4
      },
      "AccountCodes": {
       "set": [
        {"long": 111122223333},
        {"long": 444455556666},
        {"long": 123456789012}
        ]
      },
      "NetworkInfo": {
        "record": {
          "IPAddress": {"string": "192.0.2.178"},
          "Country": {"string": "United States of America"},
          "SSL": {"boolean": true}
        }
      },
      "approvedBy": {
```

```
        "entityIdentifier": {
         "entityId": "Bob",
         "entityType": "DigitalPetStore::User"
        }
      }
     }
    },
    "entities": {
      "entityList": [
        {
          "identifier": {
             "entityType": "DigitalPetStore::User",
             "entityId": "Alice"
          },
          "attributes": {
             "memberId": {
               "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
               }
          },
          "parents": [
             {
             "entityType": "DigitalPetStore::Role",
             "entityId": "Customer"
             }
          ]
        },
        {
          "identifier": {
             "entityType": "DigitalPetStore::User",
             "entityId": "Bob"
          },
          "attributes": {
             "memberId": {
               "string": "49d9b81e-735d-429c-989d-93bec0bcfd8b"
               }
          },
          "parents": [
             {
             "entityType": "DigitalPetStore::Role",
             "entityId": "Employee"
             }
          ]
        },
        {
```

```
         "identifier": {
            "entityType": "DigitalPetStore::Order",
            "entityId": "1234"
         },
         "attributes": {
            "owner": {
              "entityIdentifier": {
                "entityType": "DigitalPetStore::User",
                "entityId": "Alice"
              }
            }
         },
         "parents": []
       }
     ]
   },
   "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

亞馬遜驗證許可中的安全

雲端安全是 AWS 最重視的一環。身為 AWS 的客戶，您將能從資料中心和網路架構中獲益，這些都是專為最重視安全的組織而設計的。

安全是 AWS 與您共同肩負的責任。共同的責任模式將其稱為雲端的安全性和雲端中的安全性：

- 雲端本身的安全 – AWS 負責保護執行 AWS 雲端 內 AWS 服務的基礎設施。AWS 提供的服務，也可讓您安全使用。第三方稽核人員會定期測試和驗證我們安全性的有效性，作為 AWS 合規計劃的一部分。若要了解適用於 Amazon 驗證許可的合規計劃，請參閱AWS合規計劃範圍內的服務。
- 雲端內部的安全：您的責任取決於所使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您瞭解如何在使用已驗證權限時套用共用責任模型。下列主題說明如何設定「已驗證的權限」，以符合您的安全性與合規性目標。您還將學習如何使用其他AWS協助您監控和保護「已驗證權限」資源的服務。

主題

- 亞馬遜驗證許可中的數據保護
- Amazon 驗證許可的身分和存取管理
- 適用於 Amazon 驗證許可的合規驗證
- 亞馬遜驗證許可中的彈性

亞馬遜驗證許可中的數據保護

該AWS 共同責任模式適用於 Amazon 驗證許可中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您必須負責維護在此基礎設施上託管之內容的控制權。此內容包括您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

- 出於數據保護目的，我們建議您進行保護AWS 帳戶憑據並設置個別用戶AWS IAM Identity Center或者AWS Identity and Access Management(IAM). 如此一來，每個使用者都只會獲得授予完成其任務所必須的許可。
- 我們建議您通過以下方式保護您的數據：

  - 每個帳戶都使用多重要素驗證 (MFA)。
  - 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2。
  - 使用 AWS CloudTrail 設定 API 和使用者活動記錄。
  - 使用 AWS 加密解決方案，以及 AWS 服務 內的所有預設安全控制項。
  - 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
  - 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的詳細資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。
- 我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如 Name (名稱) 欄位。這包括當您使用已驗證權限或其他權限AWS 服務使用控制台，API，AWS CLI，或AWS軟體開發套件。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。
- 您的動作名稱不應包含任何敏感資訊。
- 我們還強烈建議您始終為實體（資源和主體）使用唯一，不可變和不可重複使用的標識符。在測試環境中，您可以選擇使用簡單的實體識別碼，例如`jane`或者`bob`對於類型的實體的名稱`User`。但是，在生產系統中，出於安全原因，使用無法重複使用的唯一值至關重要。我們建議您使用諸如通用唯一識別碼 (UUID) 之類的值。例如，考慮用戶`jane`誰離開了公司。後來，你讓別人使用這個名字`jane`。該新使用者可以自動存取仍然參照的原則所授予的所有項目`User::"jane"`。已驗證的權限和 Cedar 無法區分新使用者和先前的使用者。

  本指引同時適用於主參與者和資源識別碼。一律使用保證唯一且絕不重複使用的識別碼，以確保您不會因為原則中存在舊識別碼而無意中授予存取權。

- 確保您提供的字符串定義`Long`和`Decimal`值在每種類型的有效範圍內。另外，請確保您使用任何算術運算符不會導致有效範圍以外的值。如果超出範圍，則作業會導致溢位例外狀況。會忽略導致錯誤的原則，這表示 [允許] 原則可能會意外地無法允許存取，或 [禁止] 原則可能意外地無法封鎖存取。

資料加密

Amazon 驗證許可會自動加密所有客戶資料，例如政策AWS 受管金鑰，因此使用客戶管理的密鑰既不是必需的，也不支持。

Amazon 驗證許可的身分和存取管理

AWS Identity and Access Management (IAM) 可協助系統管理員安全地控制 AWS 資源存取權。 AWS 服務 IAM 管理員控制誰可以驗證（登錄）和授權（有權限）使用已驗證權限資源。 IAM 是一種您 AWS 服務 可以使用，無需額外費用。

主題

- 物件
- 使用身分驗證
- 使用政策管理存取權
- Amazon 驗證許可如何與 IAM
- Amazon 驗證許可的身分型政策範例
- 疑難排解 Amazon 驗證的許可身分和存取

物件

根據您在驗證權限中執行的工作，使用方式 AWS Identity and Access Management (IAM) 會有所不同。

服務使用者 — 如果您使用「已驗證權限」服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多「已驗證權限」功能來完成工作時，您可能需要其他權限。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取已驗證權限中的功能，請參閱疑難排解 Amazon 驗證的許可身分和存取。

服務管理員 — 如果您負責公司的「已驗證權限」資源，則可能擁有「已驗證權限」的完整存取權限。決定您的服務使用者應存取哪些已驗證權限功能和資源是您的工作。然後，您必須向 IAM 管理員提交請求，才能變更服務使用者的權限。檢閱此頁面上的資訊，以瞭解的基本概念 IAM。若要深入瞭解貴公司如何 IAM 透過已驗證權限使用，請參閱Amazon 驗證許可如何與 IAM。

IAM 系統管理員 — 如果您是 IAM 系統管理員，您可能想要瞭解如何撰寫原則以管理已驗證權限存取權限的詳細資訊。若要檢視您可以在中使用的已驗證權限基於身分的原則範例 IAM，請參閱。Amazon 驗證許可的身分型政策範例

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者或擔任 IAM 角色的身分驗證 (登入 AWS)。 AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。 AWS IAM Identity Center (IAM 身分中心) 使用者、貴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識別的範例。當您以同盟身分登入時，您的管理員先前會使用 IAM 角色設定聯合身分識別。當您使 AWS 用同盟存取時，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。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》 AWS 帳戶中的如何登入您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署要求的詳細資訊，請參閱使用IAM 者指南中的簽署 AWS API 要求。

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) 來增加帳戶的安全性。若要深入瞭解，請參閱使用AWS IAM Identity Center 者指南中的多因素驗證和使用多重要素驗證 (MFA) 的 AWS使用者指IAM 南。

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，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 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。如需需要您以 root 使用者身分登入的完整工作清單，請參閱《使用指南》中的〈需要 root 使用者認證的IAM 工作〉。

聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identi ty Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。 AWS Directory Service同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

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，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 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 使用者是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，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。如需詳細資訊，請參閱《使用指南》中的「IAM 定期輪換存取金鑰」以瞭解需要長期認證的使用案例。

IAM 群組是指定 IAM 使用者集合的身分識別。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 IAM Admin 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，但角色僅提供暫時憑證。若要進一步了解，請參閱《使用者指南》中的建立 IAM 使用IAM 者的時機 (而非角色)。

IAM 角色

IAM 角色是您 AWS 帳戶 中具有特定權限的身份。它類似 IAM 使用者，但不與特定的人員相關聯。您可以 AWS Management Console 透過切換角色來暫時擔任中的角色。 IAM 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需有關使用角色方法的詳細資訊，請參閱《使用指南》中的IAM 〈使用 IAM 角色〉。

IAM 具有臨時認證的角色在下列情況下很有用：

- 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。如需聯合角色的相關資訊，請參閱《使用指南》中的〈建立第三方身分識別提供IAM者的角色〉。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。
- 臨時 IAM 使用者許可 — IAM 使用者或角色可以假定某個 IAM 角色暫時取得特定任務的不同許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (受信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源（而不是使用角色作為代理）。若要瞭解跨帳戶存取角色與以資源為基礎的政策之間的差異，請參閱《IAM 使用指南》中的IAM 角色與以資源為基礎的政策有何不同。
- 執行於的應用程式 Amazon EC2— 您可以使用 IAM 角色來管理在 EC2 執行個體上執行並發出 AWS CLI 或 AWS API 請求的應用程式的臨時登入資料。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 執行個體並提供給其所有應用程式，請建立連接至執行個體的執

行個體設定檔。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暫時憑證。如需詳細資訊，請參閱《使用指南》中的使用 IAM 角色將權限授與在 Amazon EC2 執行個體上執行的應IAM 用程式。

若要了解是使用 IAM 角色還是 IAM 使用者，請參閱《使用者指南》中的建立 IAM 角色的時機 (而非使用IAM 者)。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS ，當與身分識別或資源相關聯時，會定義其權限。 AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需有關 JSON 政策文件結構和內容的詳細資訊，請參閱IAM 使用指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。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。若要授與使用者對所需資源執行動作的權限， IAM 管理員可以建立 IAM 策略。然後，系統管理員可以將 IAM 原則新增至角色，使用者可以擔任這些角色。

IAM 原則會定義動作的權限，不論您用來執行作業的方法為何。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 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。若要瞭解如何建立以身分識別為基礎的策略，請參閱《IAM 使用指南》中的〈建立 IAM 策略〉。

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。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶. 受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管策略或內嵌策略之間進行選擇，請參閱《IAM 使用手冊》中的「在受管策略和內嵌策略之間進行選擇」。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。以資源為基礎的政策範例包括 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。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的策略 IAM 中使用 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 和 Amazon VPC 是支援 ACL 的服務範例。 AWS WAF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原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 權限界限 — 許可界限是一項進階功能，您可以在其中設定以身分為基礎的政策可授予 IAM 實體 (IAM 使用者或角色) 的最大權限。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。如需有關權限界限的詳細資訊，請參閱《IAM 使用指南》中的IAM 實體的權限界限。
- 服務控制策略（SCP）— SCP 是 JSON 策略，用於指定中組織或組織單位（OU）的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。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 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需 Organizations 和 SCP 的詳細資訊，請參閱 AWS Organizations 使用者指南中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。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。若要瞭解如何在涉及多個原則類型時 AWS 決定是否允許要求，請參閱IAM 使用指南中的原則評估邏輯。

Amazon 驗證許可如何與 IAM

在您用 IAM 來管理「已驗證權限」的存取權限之前，請先了解哪些 IAM 功能可與「已驗證權限」搭配使用。

IAM 您可以搭配 Amazon 驗證許可使用的功能

IAM 特徵	已驗證權限支援
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	否
ACL	否
ABAC(政策中的標籤)	否
臨時憑證	是
主體許可	是
服務角色	否
服務連結角色	否

若要取得已驗證權限和其他 AWS 服務如何與大部分 IAM 功能搭配運作的高階檢視，請參閱《IAM 使用者指南》 IAM中的使用AWS 服務。

已驗證權限的身分型原則

支援身分型政策	是

身分型政策是可以連接到身分 (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。若要瞭解如何建立以身分識別為基礎的策略，請參閱《IAM 使用指南》中的〈建立 IAM 策略〉。

使用以 IAM 身分識別為基礎的策略，您可以指定允許或拒絕的動作和資源，以及允許或拒絕動作的條件。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。若要瞭解可在 JSON 政策中使用的所有元素，請參閱使用IAM 者指南中的 IAM JSON 政策元素參考資料。

已驗證權限的身分型原則範例

若要檢視已驗證權限身分型原則的範例，請參閱。Amazon 驗證許可的身分型政策範例

已驗證權限內的資源型政策

支援以資源基礎的政策	否

資源型政策是附加到資源的 JSON 政策文件。以資源為基礎的政策範例包括 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。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

若要啟用跨帳戶存取，您可以在以資源為基礎的策略中指定一個或多個帳戶中的一個或多個帳戶中的 IAM 實體作為主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主參與者和資源不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主參與者實體 (使用者或角色) 權限，才能存取資源。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 使用指南》IAM中的〈跨帳號資源存取〉。

已驗證權限的原則動作

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哪些內容。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。原則動作通常與關聯的 AWS API 作業具有相同的名稱。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。

若要查看已驗證許可動作清單，請參閱服務授權參考中由 Amazon 驗證許可定義的動作。

「已驗證權限」中的原則動作會在動作之前使用下列前置詞：

```
verifiedpermissions
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

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

您也可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 Get 文字的所有動作，請包含以下動作：

```
"Action": "verifiedpermissions:Get*"
```

若要檢視已驗證權限身分型原則的範例，請參閱。Amazon 驗證許可的身分型政策範例

已驗證權限的策略資源

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哪些內容。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": "*"
```

若要查看已驗證許可資源類型及其 ARN 的清單，請參閱服務授權參考中由 Amazon 驗證許可定義的資源類型。若要了解可以使用哪些動作指定每個資源的 ARN，請參閱 Amazon 驗證許可定義的動作。

已驗證權限的原則條件金鑰

支援服務特定政策條件金鑰	否

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。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，請使用邏輯OR運算來 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。如需詳細資訊，請參閱《IAM 使用指南》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條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《使用指南》中的AWS 全域條件內IAM 容索引鍵。

已驗證權限中的 ACL

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。

具有已驗證權限的 ABAC

支援 ABAC (政策中的標籤)	否

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。您可以將標籤附加至 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，請參閱什麼是 AB AC？ 在《IAM 使用者指南》中。若要檢視包含設定 ABAC 步驟的教學課程，請參閱《使用者指南》中的〈使用以屬性為基礎的存取控制 (ABAC)〉。IAM

使用具有已驗證權限的臨時憑

支援臨時憑證	是

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料搭配使用 AWS 服務 ，請參閱《IAM 使用指南》 IAM中的使用方式。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。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。如需有關切換角色的詳細資訊，請參閱《IAM 使用者指南》中的〈切換到角色 (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。

已驗證權限的跨服務主體權限

支援主體許可	是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。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 會使用主體呼叫的權限 AWS 服務，並結合要求

AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 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》。

已驗證權限的服務角色

支援服務角色	否

服務角色是服務假定代表您執行動作的IAM 角色。 IAM 管理員可以從中建立、修改和刪除服務角色 IAM。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。

已驗證權限的服務連結角色

支援服務連結角色。	否

服務連結角色是一種連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。 IAM 管理員可以檢視 (但無法編輯服務連結角色) 的權限。

如需有關建立或管理服務連結角色的詳細資訊，請參閱使用 IAM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。

Amazon 驗證許可的身分型政策範例

根據預設，使用者和角色沒有建立或修改已驗證權限資源的權限。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。 IAM 管理員必須建立 IAM 政策，授與使用者和角色權限，才能對所需的資源執行動作。管理員接著必須將這些政策連接至需要這些許可的使用者。

若要瞭解如何使用這些 JSON 政策文件範例來建立 IAM 身分型原則，請參閱使用IAM 者指南中的建立政 IAM 策。

有關由已驗證許可定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARN 格式，請參閱服務授權參考中的 Amazon 已驗證許可的動作、資源和條件金鑰。

主題

- 政策最佳實務

- 使用已驗證的權限控制台
- 允許使用者檢視他們自己的許可

政策最佳實務

以身分識別為基礎的政策會決定某人是否可以建立、存取或刪除您帳戶中的已驗證權限資源。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 受管原則並邁向最低權限權限 — 若要開始將權限授與使用者和工作負載，請使用可授與許多常見使用案例權限的AWS 受管理原則。它們可用在您的 AWS 帳戶. 建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需詳細資訊，請參閱AWS 《IAM 使用指南》中針對工作職能的AWS 受管理策略或受管理的策略。
- 套用最低權限權限 — 當您使用原則設定權限時， IAM 只授與執行工作所需的權限。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。如需有關使用套用權限 IAM 的詳細資訊，請參閱《使用指南》IAM中的IAM 《策略與權限》。
- 使用 IAM 策略中的條件進一步限制存取 — 您可以在策略中新增條件，以限制對動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 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 存取分析器驗證您的政 IAM 策，以確保安全和功能性的許可 — IAM Access Analyzer 會驗證新的和現有的政策，以便政策遵守 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，可協助您編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 IAM 存取分析器政策驗證。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。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，請參閱使用者指IAM 南中的設定 MFA 保護的 API 存取權。

如需有關中最佳作法的詳細資訊 IAM，請參閱《IAM 使用指南》 IAM中的「安全性最佳作法」。

使用已驗證的權限控制台

若要存取 Amazon 驗證許可主控台，您必須擁有最少一組許可。這些權限必須允許您列出並檢視您的 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。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。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。

若要確保使用者和角色仍可使用 [已驗證權限] 主控台，請同時將 [已驗證的權限] *ConsoleAccess* 或 [*ReadOnly* AWS 受管理的原則] 附加至實體。如需詳細資訊，請參閱《使用指南》中的〈將權限新增至IAM使用者〉。

允許使用者檢視他們自己的許可

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。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
    "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": "*"
        }
    ]
}
```

疑難排解 Amazon 驗證的許可身分和存取

使用下列資訊可協助您診斷及修正使用已驗證權限和時可能會遇到的常見問題 IAM。

主題

- 我沒有在已驗證權限中執行動作的授權
- 我沒有授權執行 iam：PassRole
- 我想允許我以外的人訪問我 AWS 帳戶 的已驗證權限資源

我沒有在已驗證權限中執行動作的授權

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 verifiedpermissions:*GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 verifiedpermissions: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 verifiedpermissions:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我沒有授權執行 iam：PassRole

如果您收到未獲授權執行iam:PassRole動作的錯誤訊息，則必須更新您的政策，以允許您將角色傳遞給「已驗證的權限」。

有些 AWS 服務 允許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。

當名為的 IAM 使用者marymajor嘗試使用主控台在已驗證許可中執行動作時，會發生下列範例錯誤。但是，動作請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 系統管理員。您的管理員提供您的簽署憑證。

我想允許我以外的人訪問我 AWS 帳戶 的已驗證權限資源

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：

- 若要瞭解已驗證權限是否支援這些功能，請參閱Amazon 驗證許可如何與 IAM。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱《使用者指南》中的另一個您擁有 AWS 帳戶 的 IAM IAM 使用者提供存取權限。
- 若要瞭解如何將資源存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用指南》中的提供第三方 AWS 帳戶 擁有的存取權。
- 若要瞭解如何透過聯合身分識別提供存取權，請參閱使用指南中的提供對外部驗證使用IAM 者的存取權 (身分聯合)。
- 若要瞭解針對跨帳號存取使用角色與以資源為基礎的政策之間的差異，請參閱《使用IAM 者指南》IAM中的〈跨帳號資源存取〉。

適用於 Amazon 驗證許可的合規驗證

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱AWS 服務 遵循規範計劃方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於您資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- 建構 HIPAA 安全性與合規性 Amazon Web Services— 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 用程式。

> **Note**
>
> 並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS 客戶合規指南 — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準 AWS 服務 與技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保護指引並對應至安全控制的最佳實務。
- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的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 提供了內部安全狀態的全面視圖 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 controls reference。
- Amazon GuardDuty — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。 GuardDuty 可協助您因應各種合規性需求，例如 PCI DSS，滿足特定合規性架構所規定的入侵偵測需求。
- AWS Audit Manager— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

亞馬遜驗證許可中的彈性

AWS 全球基礎架構是以 AWS 區域 與可用區域為中心建置的。AWS 區域 提供多個分開且隔離的實際可用區域，並以具備低延遲、高輸送量和高度備援特性的聯網相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，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。

當您建立「已驗證權限」原則存放區時，會在個人中建立AWS 區域，並會在組成該區域可用區域的資料中心之間自動複寫。目前，「已驗證的權限」不支援任何跨區域複寫。

如需 AWS 區域 與可用區域的詳細資訊，請參閱AWS全球基礎架構。

監控亞馬遜驗證許可

監控是維護 Amazon 驗證是以及其他AWS解決方案的可靠性、可用性、可用性、以及所不可或缺。AWS提供了以下列監控工具以監督，這些工具會在發生錯誤時回報，並自動適時採取動作：

- AWS CloudTrail 擷取您 AWS 帳戶發出或代表發出的 API 呼叫和相關事件，並傳送記錄檔案至您指定的 Simple Storage Service (Amazon S3) 儲存貯體。您可以找出哪些使用者和帳戶呼叫 AWS、發出呼叫的來源 IP 地址，以及呼叫的發生時間。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》。

使用記錄 Amazon 驗證許可 API 調用 AWS CloudTrail

Amazon 驗證許可與服務整合在一起AWS CloudTrail，該服務可提供使用者、角色或已驗證許可中的AWS服務所採取的動作記錄。 CloudTrail 將已驗證權限的所有 API 呼叫擷取為事件。擷取的呼叫包括來自已驗證權限主控台的呼叫，以及對已驗證權限 API 作業的程式碼呼叫。如果您建立追蹤，您可以啟用持續交付 CloudTrail 事件到 Amazon S3 儲存貯體，包括已驗證許可的事件。如果您未設定追蹤，您仍然可以在 [事件歷程記錄] 中檢視 CloudTrail 主控台中最近的事件。使用收集的資訊 CloudTrail，您可以判斷向「已驗證權限」提出的要求、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱使AWS CloudTrail用者指南。

已驗證的權限資訊 CloudTrail

CloudTrail 在您創建帳戶AWS 帳戶時啟用。當活動發生在已驗證的權限中時，該活動會與 CloudTrail 事件歷史記錄中的其他AWS服務事件一起記錄在事件中。您可以檢視、搜尋和下載 AWS 帳戶 的最新事件。如需詳細資訊，請參閱使用 CloudTrail 事件歷程記錄檢視事件。

如需您AWS 帳戶的事件的持續記錄 (包括已驗證權限的事件)，請建立追蹤。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您還可以設定其他AWS服務，以進一步分析 CloudTrail 記錄中收集的事件資料並採取行動。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務與整合
- 設定 Amazon SNS 通知 CloudTrail

- 從多個區域接收 CloudTrail 日誌文件並從多個帳戶接收 CloudTrail 日誌文件

所有已驗證許可動作均由記錄， CloudTrail 並記錄在 Amazon 驗證許可 API 參考指南中。例如，呼叫`CreateIdentitySourceDeletePolicy`、和`ListPolicyStores`動作會在 CloudTrail 記錄檔中產生項目。

每一筆事件或日誌項目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：

- 該請求是否使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 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 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

建立追蹤或事IsAuthorizedWithToken件資料存放區時，預設情況下不會記錄類似IsAuthorized和的資料事件。若要記錄資 CloudTrail 料事件，您必須明確新增要收集活動的支援資源或資源類型。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的資料事件。

瞭解已驗證的權限記錄檔項目

追蹤是一種組態，可讓事件以日誌檔的形式傳遞到您指定的 Amazon S3 儲存貯體。 CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求，包括有關請求的操作，動作的日期和時間，請求參數等信息。 CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

主題

- IsAuthorized
- BatchIsAuthorized
- CreatePolicyStore
- ListPolicyStores
- DeletePolicyStore
- PutSchema
- GetSchema
- CreatePolicyTemplate
- DeletePolicyTemplate
- CreatePolicy

- GetPolicy
- CreateIdentitySource
- GetIdentitySource
- ListIdentitySources
- DeleteIdentitySource

> **Note**
>
> 已修改資料隱私權範例的某些欄位。

IsAuthorized


```
{
    "eventVersion": "1.08",
    "userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLE_PRINCIPAL_ID",
  "arn": "arn:aws:iam::123456789012:role/ExampleRole",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2023-11-20T22:55:03Z",
    "eventSource": "verifiedpermissions.amazonaws.com",
    "eventName": "IsAuthorized",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
    "requestParameters": {
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "alice"
        },
        "action": {
            "actionType": "PhotoFlash::Action",
            "actionId": "ViewPhoto"
        },
        "resource": {
            "entityType": "PhotoFlash::Photo",
```

```
            "entityId": "VacationPhoto94.jpg"
        },
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "decision": "ALLOW"
    },
    "requestID": "346c4b6a-d12f-46b6-bc06-6c857bd3b28e",
    "eventID": "8a4fed32-9605-45dd-a09a-5ebbf0715bbc",
    "readOnly": true,
    "resources": [
        {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
}
```

BatchIsAuthorized

```
{
    "eventVersion": "1.08",
    "userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLE_PRINCIPAL_ID",
  "arn": "arn:aws:iam::123456789012:role/ExampleRole",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2023-11-20T23:02:33Z",
    "eventSource": "verifiedpermissions.amazonaws.com",
    "eventName": "BatchIsAuthorized",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
```

```
    "requestParameters": {
        "requests": [
            {
                "principal": {
                    "entityType": "PhotoFlash::User",
                    "entityId": "alice"
                },
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "ViewPhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            },
            {
                "principal": {
                    "entityType": "PhotoFlash::User",
                    "entityId": "annalisa"
                },
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "DeletePhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            }
        ],
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "results": [
            {
                "request": {
                    "principal": {
                        "entityType": "PhotoFlash::User",
                        "entityId": "alice"
                    },
                    "action": {
                        "actionType": "PhotoFlash::Action",
```

```
                        "actionId": "ViewPhoto"
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
                "decision": "ALLOW"
            },
            {
                "request": {
                    "principal": {
                        "entityType": "PhotoFlash::User",
                        "entityId": "annalisa"
                    },
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "DeletePhoto"
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
                "decision": "DENY"
            }
        ]
    },
    "requestID": "a8a5caf3-78bd-4139-924c-7101a8339c3b",
    "eventID": "7d81232f-f3d1-4102-b9c9-15157c70487b",
    "readOnly": true,
    "resources": [
        {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
```

```
}
```

CreatePolicyStore

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "validationSettings": {
      "mode": "OFF"
    }
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-22T07:43:33.962794Z",
    "lastUpdatedDate": "2023-05-22T07:43:33.962794Z"
  },
  "requestID": "1dd9360e-e2dc-4554-ab65-b46d2cf45c29",
  "eventID": "b6edaeee-3584-4b4e-a48e-311de46d7532",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

ListPolicyStores


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListPolicyStores",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "maxResults": 10
  },
  "responseElements": null,
  "requestID": "5ef238db-9f87-4f37-ab7b-6cf0ba5df891",
  "eventID": "b0430fb0-12c3-4cca-8d05-84c37f99c51f",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```


DeletePolicyStore


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
```

```
  "eventName": "DeletePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "1368e8f9-130d-45a5-b96d-99097ca3077f",
  "eventID": "ac482022-b2f6-4069-879a-dd509123d8d7",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

PutSchema

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T12:58:57Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "PutSchema",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
```

```
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T12:58:57.513442Z",
    "namespaces": "[some_namespace]",
    "createdDate": "2023-05-16T12:58:57.513442Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "631fbfa1-a959-4988-b9f8-f1a43ff5df0d",
  "eventID": "7cd0c677-733f-4602-bc03-248bae581fe5",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetSchema

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-25T01:12:07Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetSchema",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
```

```
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "a1f4d4cd-6156-480a-a9b8-e85a71dcc7c2",
  "eventID": "0b3b8e3d-155c-46f3-a303-7e9e8b5f606b",
  "readOnly": true,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "222222222222",
  "eventCategory": "Management"
}
```

CreatePolicyTemplate

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T13:00:24Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T13:00:23.444404Z",
    "createdDate": "2023-05-16T13:00:23.444404Z",
```

```
    "policyTemplateId": "PTEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "73953bda-af5e-4854-afe2-7660b492a6d0",
  "eventID": "7425de77-ed84-4f91-a4b9-b669181cc57b",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

DeletePolicyTemplate

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-25T01:11:48Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeletePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "5ff0f22e-6bbd-4b85-a400-4fb74aa05dc6",
```

```
  "eventID": "c0e0c689-369e-4e95-a9cd-8de113d47ffa",
  "readOnly": false,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "222222222222",
  "eventCategory": "Management"
}
```

CreatePolicy

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:42:30Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityType": "PhotoApp::Role",
```

```
        "entityId": "PhotoJudge"
    },
    "resource": {
        "entityType": "PhotoApp::Application",
        "entityId": "PhotoApp"
    },
    "lastUpdatedDate": "2023-05-22T07:42:30.70852Z",
    "createdDate": "2023-05-22T07:42:30.70852Z"
  },
  "requestID": "93ffa151-3841-4960-9af6-30a7f817ef93",
  "eventID": "30ab405f-3dff-43ff-8af9-f513829e8bde",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetPolicy

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:29Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetPolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
```

```
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "23022a9e-2f5c-4dac-b653-59e6987f2fac",
  "eventID": "9b4d5037-bafa-4d57-b197-f46af83fc684",
  "readOnly": true,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```


CreateIdentitySource


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-19T01:27:44Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreateIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "configuration": {
      "cognitoUserPoolConfiguration": {
```

```
        "userPoolArn": "arn:aws:cognito-idp:000011112222:us-east-1:userpool/us-
east-1_aaaaaaaaaa"
      }
    },
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
  },
  "responseElements": {
    "createdDate": "2023-07-14T15:05:01.599534Z",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-07-14T15:05:01.599534Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
  "requestID": "afcc1e67-d5a4-4a9b-a74c-cdc2f719391c",
  "eventID": "f13a41dc-4496-4517-aeb8-a389eb379860",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

GetIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:31Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
```

```
  "eventName": "GetIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "7a6ecf79-c489-4516-bb57-9ded970279c9",
  "eventID": "fa158e6c-f705-4a15-a731-2cdb4bd9a427",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

ListIdentitySources

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T20:05:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListIdentitySources",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
```

```
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "95d2a7bc-7e9a-4efe-918e-97e558aacaf7",
  "eventID": "d3dc53f6-1432-40c8-9d1d-b9eeb75c6193",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

DeleteIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeleteIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
```

```
  "requestID": "d554d964-0957-4834-a421-c417bd293086",
  "eventID": "fe4d867c-88ee-4e5d-8d30-2fbc208c9260",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

使用創建 Amazon 驗證許可資源 AWS CloudFormation

Amazon 驗證許可與整合 AWS CloudFormation，這項服務可協助您建立資源模型和設定 AWS 資源，以減少建立和管理資源和基礎設施的時間。您可以建立描述您想要的所有 AWS 資源 (例如原則存放區) 的範本，並為您 AWS CloudFormation 佈建和設定這些資源。

使用時 AWS CloudFormation，您可以重複使用範本，以一致且重複地設定「已驗證權限」資源。描述您的資源一次，然後在多個區域中一遍又一遍地佈建相同 AWS 帳戶 的資源。

> **Important**
>
> Amazon Cognito 可身份與 Amazon 驗證許可無法在所有相 AWS 區域 同的情況下使用。如果您收到有 AWS CloudFormation 關 Amazon Cognito 身分識別的錯誤，例如`Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient`，我們建議您在可使用 Amazon Cognito 身分的地理位置建立 Amazon Cognito 使用者集區和用戶端。 AWS 區域 建立已驗證權限身分識別來源時，請使用此新建立的使用者集區。

已驗證的權限和 AWS CloudFormation 範本

若要佈建和設定已驗證權限及相關服務的資源，您必須瞭解AWS CloudFormation 範本。範本是以 JSON 或 YAML 格式化的文本檔案。這些範本說明您要在 AWS CloudFormation 堆疊中佈建的資源。如果您不熟悉 JSON 或 YAML，可以使用 AWS CloudFormation 設計工具來協助您開始 AWS CloudFormation 使用範本。如需詳細資訊，請參閱什麼是 AWS CloudFormation 設計師？ 在《AWS CloudFormation 使用者指南》中。

已驗證的權限支援在中建立身分識別來源、原則、原則存放區和原則範本 AWS CloudFormation。如需詳細資訊，包括已驗證許可資源的 JSON 和 YAML 範本範本範例，請參閱AWS CloudFormation 使用者指南中的 Amazon 已驗證許可資源類型參考。

AWS CDK 構建

這 AWS Cloud Development Kit (AWS CDK) 是一個開放原始碼軟體開發架構，用於在程式碼中定義雲端基礎架構，並透過 AWS CloudFormation. 構造，或可重複使用的雲組件，可用於創建 AWS CloudFormation 模板。然後，您可以使用這些範本來部署您的雲端基礎架構。

若要深入瞭解並下載 AWS CDK，請參閱 AWS Cloud Development Kit。

以下是「已驗證權限」 AWS CDK 資源 (例如建構) 的文件連結。

- Amazon 驗證許可 L2 CDK 構造

進一步了解 AWS CloudFormation

若要進一步了解 AWS CloudFormation，請參閱下列資源：

- AWS CloudFormation
- AWS CloudFormation 使用者指南
- AWS CloudFormation API 參考
- AWS CloudFormation 指令行介面使用者指南

使用介面端點存取 Amazon 已驗證的權量 (AWS PrivateLink)

您可以使用AWS PrivateLink在 VPC 與 Amazon 驗證的權限之間建立 VPC 和 Amazon 驗證的私有連線。您可以如在 VPC 中一樣存取「已驗證的權限」，無需使用網際網路閘道、NAT 裝置、VPN 連線或AWS Direct Connect連線。VPC 中的執行個體無需公有 IP 地址，即可存取已驗證的權限。

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「已驗證的權限」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。

已驗的考量

為已驗證的權限設定介面端點之前，請先檢閱AWS PrivateLink指南中的考量事項。

「已驗證的權限」支援透過介面端點呼叫其所有 API 動作。

已驗證的權限不支援 VPC 端點政策。依預設，允許透過介面端點完整存取「已驗證的權限」。或者，您也可以將安全群組與端點網路介面相關聯，以控制透過介面端點傳輸至已驗證權的流量。

建立已驗證的介面端點

您可使用 Amazon VPC 主控台或 AWS Command Line Interface (AWS CLI) 來為已驗證許可建立 Amazon VPC 主控台的介面端點。如需詳細資訊，請參閱 《AWS PrivateLink 指南》中的建立介面端點。

使用以下服務名稱為「已驗證的權限」建立的介面端點：

```
com.amazonaws.region.verifiedpermissions
```

如果您為介面端點啟用私有 DNS，則可以使用其預設的區域 DNS 名稱向已驗證權限發出 API 要求。例如：`verifiedpermissions.us-east-1.amazonaws.com`。

Amazon 驗證許可的配額

您的每項 AWS 服務都 AWS 帳戶 有預設配額 (先前稱為限制)。除非另有說明，否則每個配額都是區域特定規定。您可以請求提高某些配額，而其他配額無法提高。

若要檢視已驗證權限的配額，請開啟「Service Quotas」主控台。在功能窗格中，選擇 [AWS 服務]，然後選取 [驗證權限]。

若要請求增加配額，請參閱 Service Quotas 使用者指南中的請求提高配額。如果 Service Quotas 中尚未提供配額，請使用增加服務配額表單。

您 AWS 帳戶 有下列與已驗證權限相關的配額。

主題

- 資源配額
- 階層配額
- 每秒作業的配額

資源配額

名稱	預設	可調整	描述
每個帳戶每個區域的政策存放	每個受支援的區域：1,000	是	策略存放區的數目上限。
每個原則存放區的原則範	每個受支援的區域：40	是	原則存放區中原則範本的數目上限。
每個原則存放區的身分識	1	否	您可以為原則存放區定義的身分識別來源數目上限。
授權請求大小 [1]	1 MB	否	授權要求的大小上限。
政策大小	10,000 位元組	否	個別策略的大小上限。

名稱	預設	可調整	描述
綱要大小	100,000 位元組	否	原則存放區之結構描述的大小上限。
每個資源的策略大小	20 萬字節	否	參考特定資源之所有策略的大小上限。

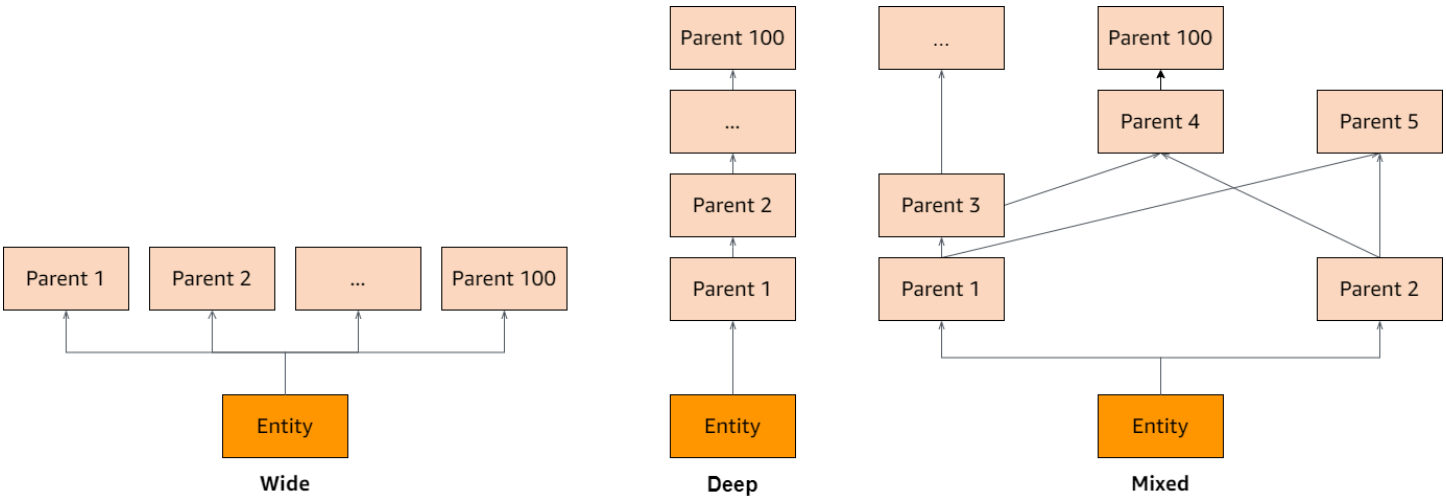
[1] IsAuthorized和的授權要求配額相同IsAuthorizedWithToken。

[2] 與單一資源相關的所有策略總大小不得超過 200,000 個位元組。對於範本連結的策略，原則範本的大小只會計算一次，再加上用於實體化每個範本連結策略的每組參數大小。

階層配額

名稱	預設	可調整	描述
每位校長的傳遞父母	100	否	每個主體的轉移父項數目上限。
每個動作的傳遞父母	100	否	每個動作的轉移父項數目上限。
每個資源的傳遞父項	100	否	每個資源的轉移父項數目上限。

下圖說明如何針對實體 (主參與者、動作或資源) 定義傳遞父項。



每秒作業的配額

當應用程式要求超過 API 作業的配額 AWS 區域 時，已驗證的權限會將要求限制至服務端點。當您超過每秒要求的配額，或者您嘗試同時寫入作業時，已驗證的權限可能會傳回例外狀況。您可以在「Service Quotas」中檢視目前的 RPS 配額。若要防止應用程式超出作業的配額，您必須針對重試和指數輪詢進行最佳化。有關詳情，請參閱使用輪詢模式重試和管理和監控工作負載中的 API 節流。

名稱	預設	可調整	描述
BatchIsAuthorized 每個帳戶每個區域的每秒要求數	每個受支援的區域：30	是	每秒 BatchIsAuthorized 要求的最大數目。
BatchIsAuthorizedWithToken 每個帳戶每個區域的每秒要求數	每個受支援的區域：30	是	每秒 BatchIsAuthorizedWithToken 要求的最大數目。
CreatePolicy 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 CreatePolicy 要求的最大數目。
CreatePolicyStore 每個帳戶每個區域的每秒要求數	每個受支援的區域：1	否	每秒 CreatePolicyStore 要求的最大數目。
CreatePolicyTemplate 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 CreatePolicyTemplate 要求的最大數目。

名稱	預設	可調整	描述
DeletePolicy 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 DeletePolicy 要求的最大數目。
DeletePolicyStore 每個帳戶每個區域的每秒要求數	每個受支援的區域：1	否	每秒 DeletePolicyStore 要求的最大數目。
DeletePolicyTemplate 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 DeletePolicyTemplate 要求的最大數目。
GetPolicy 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 GetPolicy 要求的最大數目。
GetPolicyTemplate 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 GetPolicyTemplate 要求的最大數目。
GetSchema 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 GetSchema 要求的最大數目。
IsAuthorized 每個帳戶每個區域的每秒要求數	每個受支援的區域：200	是	每秒 IsAuthorized 要求的最大數目。
IsAuthorizedWithToken 每個帳戶每個區域的每秒要求數	每個受支援的區域：200	是	每秒 IsAuthorizedWithToken 要求的最大數目。
ListPolicies 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 ListPolicies 要求的最大數目。
ListPolicyStores 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 ListPolicyStores 要求的最大數目。
ListPolicyTemplates 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 ListPolicyTemplates 要求的最大數目。
PutSchema 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 PutSchema 要求的最大數目。

名稱	預設	可調整	描述
UpdatePolicy 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 UpdatePolicy 要求的最大數目。
UpdatePolicyTemplate 每個帳戶每個區域的每秒要求數	每個受支援的區域：10	是	每秒 UpdatePolicyTemplate 要求的最大數目。

Amazon 驗證許可使用者指南的文件歷史記錄

下表說明已驗證權限的說明文件版本。

變更	描述	日期
OIDC 身分識別來源	您現在可以從 OpenID Connect（OIDC）身份提供商授權用戶。	2024年6月8日
使用身份源令牌進行 Batch 授權	您現在可以在單一 `BatchIsAuthorizedWithToken` API 請求中從 Amazon Cognito 使用者集區授權使用者。	2024年4月5 日
使用 API Gateway 建立原則存放區	您現在可以從現有的 API 和 Amazon Cognito 使用者集區建立政策存放區。	2024年4月1日
上下文概念和範例	已新增具有已驗證權限之授權要求中內容的相關資訊	2024年2月1日
授權概念與範例	新增有關具有已驗證權限之授權要求的資訊	2024年2月1日
AWS CloudFormation 整合	已驗證的權限支援在中建立身分識別來源、原則、原則存放區和原則範本 AWS CloudFormation。	2023 年 6 月 30 日
初始版本	Amazon 驗證許可使用者指南的初始版本	2023 年 6 月 13 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。