歡迎學習模組 8：確保使用者和應用程式訪問安全。

模組概覽

章節

1. 架構需求
2. 賬戶用戶和 IAM
3. 組織用戶
4. 用戶聯合身份管理
5. 多個帳戶

演示

- EC2 實例設定檔

活動

- **檢查** IAM **策略**

實驗

- 挑戰實驗：使用 IAM 控制 AWS 帳戶訪問


知識考核

aws

本模組包含以下章節：

1. 架構需求
2. 賬戶用戶和 IAM
3. 組織用戶
4. 用戶聯合身份管理
5. 多個帳戶

本模組還包括：

- 一個演示，向您展示常用功能。將一個 IAM 角色（該角色授予訪問 Amazon Web Services (AWS) 其他服務的許可權）附加到 Amazon Elastic Compute Cloud (Amazon EC2) 實例
- 一項活動，要求您分析 AWS Identity and Access Management (IAM) 策略文檔，以確定策略允許或拒絕哪些操作
- 一個挑戰實驗，您將在實驗中使用 IAM 配置適合咖啡館使用案例的用戶、組和訪問策略

最後，您需要完成一個知識考核，以測試您對本模組中涵蓋的關鍵概念的理解程度。

模組目標

學完本模組後，您應該能夠：

- 說明 AWS Identity and Access Management (IAM) 用戶、組和角色的用途
- 描述如何允許在架構中使用用戶聯合以提高安全性
- 瞭解 AWS Organizations 服務控制策略 (SCP) 如何提高架構內的安全性
- 描述如何管理多個 AWS 帳戶
- 配置 IAM 用戶

學完本模組後，您應該能夠:
- 說明 AWS Identity and Access Management (IAM) 用戶、組和角色的用途
- 描述如何允許在架構中使用用戶聯合以提高安全性
- 瞭解 AWS Organizations 服務控制策略 (SCP) 如何提高架構內的安全性
- 描述如何管理多個 AWS 帳戶
- 配置 IAM 用戶


第 1 節：架構需求
模組 8：確保使用者和應用程式訪問安全
aws
© 2023，Amazon Web Services, Inc. 或其聯屬公司。保留所有權利。

介紹第 1 節：架構需求

咖啡館業務需求

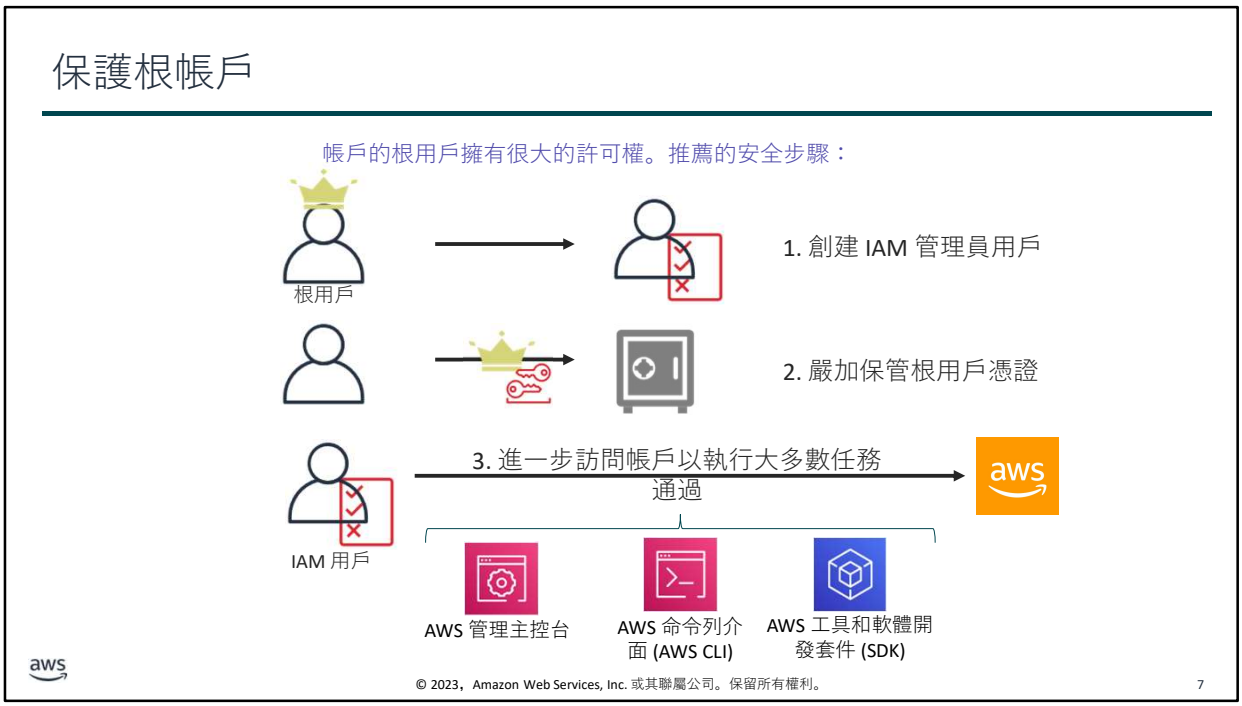
咖啡館需要定義使用者和系統在雲資源中應具有的存取權限級別，然後在整個 AWS 帳戶中實施這些存取控制。

咖啡館必須定義使用者和系統在其雲資源中應具有的存取權限級別。然後，他們必須在其整個 AWS 帳戶中實施這些存取控制。

現在，咖啡館的規模擴張，有專門的團隊成員負責在 AWS 上構建、維護或訪問應用程式（例如開發人員或資料庫管理員）。到目前為止，他們還沒有開始根據每個使用者的角色和責任明確定義他們應具有的存取權限級別。

在本模組中，您將瞭解 IAM，它可以提供您在滿足這些新業務要求時所需的功能。

介紹第 2 節：帳戶用戶和 IAM。



當您第一次創建 AWS 帳戶時，首先創建*根用戶*。此使用者可以使用用於創建帳戶的電子郵寄地址登錄 AWS 管理主控台。

AWS 帳戶根用戶具有對帳戶中所有資源的完全存取權限，包括帳單資訊、使用者資料中的個人資料以及在帳戶的任何 AWS 服務中創建的所有資源。您無法控制 AWS 帳戶根用戶憑證的許可權。

AWS 強烈建議您不要在與 AWS 的日常交互中使用根用戶憑證，而是應創建一個或多個 IAM 用戶。將根用戶憑證保存在安全的位置。對於大多數正在進行的帳戶訪問和管理任務，您可以使用 IAM 用戶憑證。

AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM)

- 安全地控制個人和組對 AWS 資源的存取權限
- 與其他 AWS 服務集成
- 聯合身份管理
- 精細許可權
- 支持多重身份驗證

AWS Identity and Access Management 也稱為 IAM。這項服務允許您配置對 AWS 資源的精細存取控制。IAM 允許您向用戶和組授予唯一的安全憑證，從而實現安全最佳實踐。這些憑證指定他們可以訪問哪些 AWS 服務 Application Programming Interface (API) 和資源。IAM 默認已啟用安全保護。用戶無法訪問 AWS 資源，除非被明確授予許可權。

IAM 已集成到大多數 AWS 服務中。您可以在 AWS 管理主控台中集中定義存取控制，這些控制將在整個 AWS 環境中生效。

您可以使用現有身份系統通過 IAM 向您的員工和應用程式授予對 AWS 管理主控台和 AWS 服務 API 的存取權限。 AWS 支援從公司系統聯合身份，如 Microsoft Active Directory 和標準的身份提供商。IAM 還支持多重身份驗證 (MFA)。如果已啟用 MFA 且 IAM 使用者嘗試登錄，則系統將提示他們輸入身份驗證代碼。身份驗證代碼將傳送到 AWS MFA 設備。MFA 設備可以是硬體 MFA 設備，也可以是使用者通過運行在用戶智慧手機上的應用程式訪問的虛擬 MFA 設備，如 Google Authenticator。

您可以創建具有類似於 AWS 帳戶根用戶的許可權的帳戶。但是，最好創建管理帳戶，僅授予所需的帳戶許可權。遵循最低許可權原則。例如，思考您的資料庫管理員 (DBA) 是否應該能夠預置 EC2 實例。如果不能，則相應地預置帳戶。

IAM 組件：回顧

IAM 用戶

在 AWS 帳戶中定義。使用憑證通過程式設計方式或 AWS 管理主控台進行身份驗證。

IAM 組

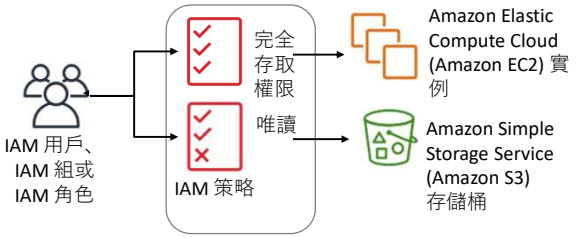
獲得相同授權的 IAM 使用者集合。

IAM 策略

定義可以訪問哪些資源以及對每個資源的訪問級別。

IAM 角色

授予臨時存取權限以供發出 AWS 服務請求的機制。可由人員、應用程式或服務*代入*。



要瞭解如何使用 IAM 保護您的 AWS 帳戶，必須瞭解四個 IAM 元件的角色和功能。

*IAM 用戶*是 AWS 帳戶中定義的人員或應用程式，它們必須對 AWS 產品進行 API 調用。每個用戶在 AWS 帳戶內都必須具有唯一的名稱（名稱中不含空格）和一組不得與其他用戶共用的安全憑證。這些憑證不同於 AWS 帳戶根用戶的安全憑證。每個用戶在一個且僅能在一個 AWS 帳戶中定義。

*IAM 組*是 IAM 使用者的集合。您可使用 IAM 組簡化為多個用戶指定和管理許可權的方式。

*IAM 策略*是一個定義許可權的文檔，可確定用戶在 AWS 帳戶中可以執行和不可以執行的操作。

*IAM 角色*是一個工具，用於授予對 AWS 帳戶中特定 AWS 資源的臨時存取權限。

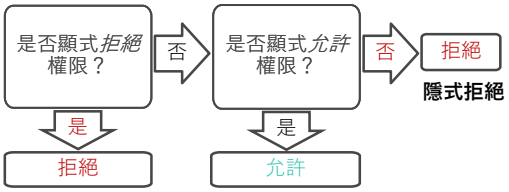
IAM 許可權

IAM 策略

許可權在 IAM 策略中指定：

- 文檔採用 JavaScript 物件標記法 (JSON) 格式
- 它定義允許使用哪些資源和操作
- 最佳實踐 – 遵循最低許可權原則
- 兩種策略類型 –
  - 基於身份：附加到 IAM 主體
  - 基於資源：附加到 AWS 資源

IAM 在收到請求時如何確定許可權：



在 IAM 中，許可權在 IAM 策略文檔中定義。策略允許您對授予給主體的許可權進行微調。示例主體包括 IAM 用戶、IAM 角色或其他 AWS 服務。

在 IAM 確定是否允許許可權時，它會先檢查是否存在任何適用的*顯式拒絕策略*。如果不存在顯式拒絕策略，IAM 會繼續檢查是否存在任何適用的*顯式允許策略*。如果不存在顯式拒絕或顯式允許策略，IAM 將恢復為默認設置並拒絕訪問。這一過程稱為*隱式拒絕*。僅當所請求的操作*不是*顯式拒絕，而*是*顯式允許時，才允許用戶執行操作。

在制定 IAM 策略時，可能很難確定是否會向 IAM 實體授予對資源的存取權限。IAM 策略模擬器是一個非常有用的工具，可用於對 IAM 策略進行測試和故障排除。

策略存儲為 JavaScript 物件標記法 (JSON) 文檔。它們可以作為*基於身份的策略*附加到主體，也可以作為*基于資源的策略*附加到資源。

基於身份的策略與基於資源的策略

基於身份的策略

- 附加到用戶、組或角色
- 策略類型
  - AWS 託管的策略
  - 客戶託管的策略
  - 內聯策略

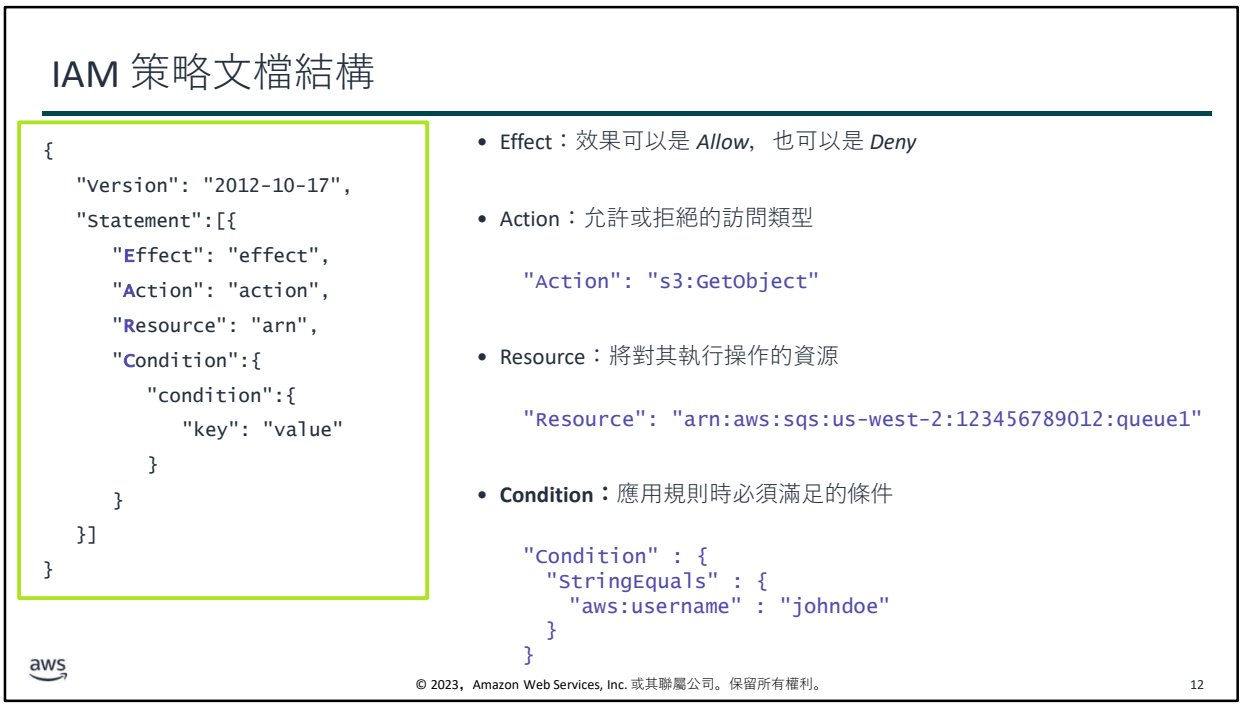
基於資源的策略

- 附加到 AWS 資源
  - 示例：附加到 Amazon S3 存儲桶
- 始終是內聯策略

*基於身份的策略*是可以附加到主體（也稱為"身份"，例如 IAM 用戶、角色或組）的許可權策略。這些策略可以*控制該身份在什麼條件下可以針對哪些資源執行哪些操作*。

基於身份的策略可以進一步歸類為 AWS 託管策略、客戶託管策略或內聯策略。*AWS 託管策略*由 AWS 創建和管理，可以附加到您的 AWS 帳戶中的多個用戶、組和角色。如果您剛開始使用策略，建議您從使用 AWS 託管策略開始。*客戶託管策略*是由您在 AWS 帳戶中創建和管理的策略。與 AWS 託管策略相比，客戶託管策略可以更精確地控制您的策略。您可以通過視覺化編輯器或直接創建 JSON 策略文檔來創建和編輯 IAM 策略。*內聯策略*是由您創建和管理的策略，直接嵌入在單個用戶、組或角色中。

*基於資源的策略*是附加到資源（如 Amazon Simple Storage Service (Amazon S3) 存儲桶）的 JSON 策略文檔。這些策略可以*控制特定主體在什麼條件下可以針對該資源執行哪些操作*。基於資源的策略是內聯策略，沒有基於資源的託管策略。



IAM 策略在 AWS 中存儲為 JSON 文檔。基於身份的策略是附加到用戶或角色的策略文檔。基於資源的策略是附加到資源的策略文檔。策略文檔包括一個或多個單獨語句。每個語句都包含有關單個許可權的資訊。如果策略包含多個語句，則 AWS 會在評估它們時跨語句應用邏輯“或”。

以下是 IAM 策略文檔中的常見元素：

- **Version** – 指定要使用的策略語言的版本。作為最佳實踐，請使用最新的 2012-10-17 版本。
- **Statement** – 使用此主要策略元素作為以下元素的容器。可以在一個策略中包含多個語句。
- **Effect** – 使用 Allow 或 Deny 來指示策略是允許還是拒絕訪問。
- **Principal** – 如果創建基於資源的策略，您必須指示要允許或拒絕訪問的帳戶、用戶、角色或聯合身份用戶。如果您要創建 IAM 許可權策略以附加到用戶或角色，則不能包含此元素。主體隱式代表該用戶或角色。
- **Action** – 包括策略允許或拒絕的操作列表。
- **Resource** – 如果創建 IAM 許可權策略，您必須指定操作適用的資源列表。如果您創建基於資源的策略，則此元素是可選的。
- **Condition**（可選）– 指定策略在哪些情況下授予許可權。

ARN 和萬用字元

- 使用 Amazon Resource Name (ARN) 格式標識資源
  - 語法 – `arn:partition:service:region:account:resource`
  - 示例 – `"Resource": "arn:aws:iam::123456789012:user/mmajor"`
- 您可以使用萬用字元 (*) 來授予針對特定 AWS 服務的所有操作
  - 示例 –
    - `"Action": "s3:*"`
    - `"Action": "iam:*AccessKey*"`

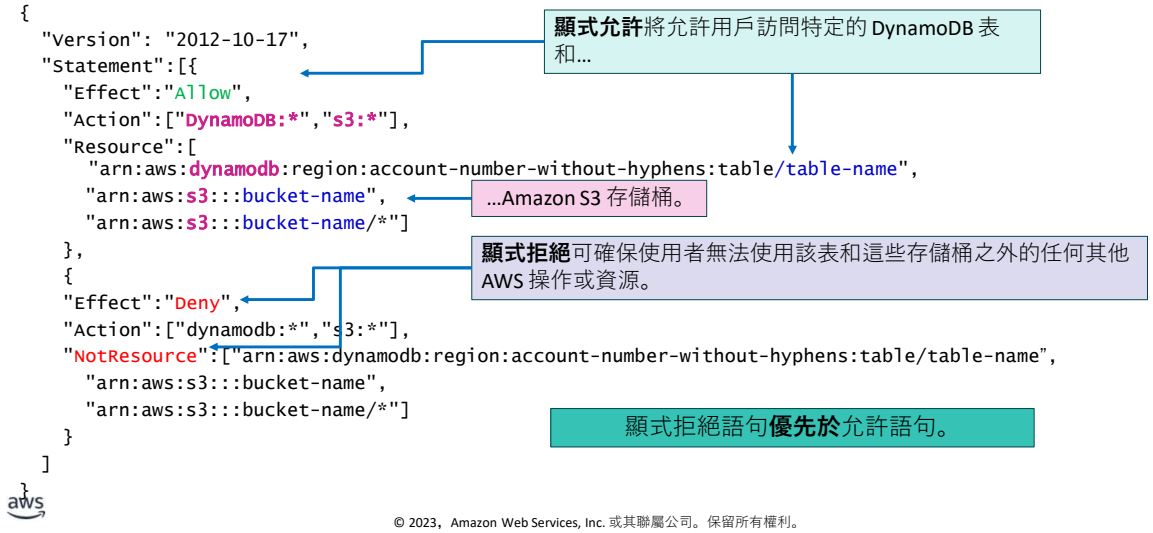
對於基於身份的（IAM 許可權）策略，您必須指定操作適用的資源列表。*Resource* 元素指定語句涵蓋的一個或多個物件。語句必須包含 Resource 或 NotResource 元素。

大多數資源都有易記名稱（例如，名為 *Bob* 的用戶或名為 *Developers* 的組）。不過，許可權策略語言要求您使用以下 *Amazon Resource Name (ARN)* 格式指定資源。

每項服務都有自己的一套資源。雖然您總是使用 ARN 來指定資源，但資源的 ARN 細節取決於服務和資源。有關如何指定資源的資訊，請根據編寫的語句所涉及的資源，參閱相應資源所屬服務的服務文檔。

您還可以在 IAM 策略文檔中使用萬用字元，例如在 ARN 或 Action 中。您可以使用萬用字元 (*)。星號 (*) 表示 0 個或多個字元的任意組合。例如，“*Action*”值“*s3:**”適用於所有 S3 操作。還可以使用萬用字元 (*) 作為操作名稱的一部分。例如，“*Action*”值“*iam:*AccessKey**”適用於包含字串 *AccessKey* 的所有 IAM 操作，包括 CreateAccessKey、DeleteAccessKey、ListAccessKeys、和 UpdateAccessKey。

IAM 策略示例



如前所述，IAM 策略文檔以 JSON 格式編寫。

此示例 IAM 策略將僅授予用戶訪問以下資源的許可權：
- 名稱用 *table-name* 表示的 Amazon DynamoDB 表。
- AWS 帳戶內名稱用 *bucket-name* 表示的 S3 存儲桶及其包含的所有物件。

該 IAM 策略還包含顯式拒絕 ("Effect":"Deny") 元素。該 *NotResource* 元素有助於確保使用者不能使用策略中指定的操作和資源以外的任何其他 DynamoDB 或 S3 操作或資源，即使在其他策略中已授予相關許可權也不例外。顯式拒絕語句優先於允許語句。

活動：檢查 IAM 策略

照片由 Pixabay 提供（摘自 Pexels）。

在此講師指導的活動中，會為您提供示例 IAM 策略。對於每項策略，您需要回答有關該策略是允許還是拒絕特定操作的問題。講師引導對每個問題進行討論，一次揭曉一個正確答案。

活動：IAM 策略分析 (1/3)

考慮此 IAM 策略，然後回答問題。

```
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": [
         "iam:Get*",
         "iam:List*"
      ],
      "Resource": "*"
   }
}
```

1. 此策略授予您對哪種 AWS 服務的存取權限？
2. 它是否允許您創建 IAM 用戶、組、策略或角色？
3. 轉到 https://docs.aws.amazon.com/IAM/latest/UserGuide/，然後在左側巡覽列中展開 *Reference （參考）> Policy Reference （策略參考）> Actions, Resources, and Condition Keys （操作、資源和條件鍵）*。選擇 *Identity and Access Management*。滾動到 Actions Defined by Identity And Access Management （Identity And Access Management 定義的操作）列表。

   至少指出 iam:Get* 操作允許的三個特定操作。

查看 IAM 策略文檔示例。講師現在將向您提出一系列問題，來評估您是否理解此策略將允許和拒絕哪些操作。

活動：IAM 策略分析 (1/3) – 答案

考慮此 IAM 策略，然後回答問題。

```
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": [
         "iam:Get*",
         "iam:List*"
      ],
      "Resource": "*"
   }
}
```

1. 此策略授予您對哪種 AWS 服務的存取權限？
   - 答案：IAM 服務。
2. 它是否允許您創建 IAM 用戶、組、策略或角色？
   - 答案：否。存取權限僅限於 *get* 和 *list* 請求。它實際上授予的是唯讀許可權。
3. 轉到 https://docs.aws.amazon.com/IAM/latest/UserGuide/，然後在左側巡覽列中展開 *Reference （參考）> Policy Reference （策略參考）> Actions, Resources, and Condition Keys（操作、資源和條件鍵）*。選擇 *Identity and Access Management*。滾動到 Actions Defined by Identity And Access Management（Identity And Access Management 定義的操作）列表。

   至少指出 iam:Get* 操作允許的三個特定操作。
   - 答案：iam:Get* 允許許多特定的操作，包括 GetGroup、GetPolicy、GetRole 等。

答案揭曉。

活動：IAM 策略分析 (2/3)

考慮此 IAM 策略，然後回答問題。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

1. 該策略是否允許您隨時無條件地終止任何 EC2 實例？
2. 您是否有權從任何地方進行終止實例調用？
3. 如果您從分配的 IP 地址為 *192.0.2.243* 的伺服器進行調用，您能否終止實例？

分析第二個 IAM 策略檔示例。第一部分顯示對資源的操作 ec2:TerminateInstance 實施“允許”效果。第二部分顯示在 NotIpAddress aws:SourceIp 192.0.2.0/24 和 203.0.113.0/24 條件下，對資源的操作 ec2:TerminateInstances 實施“拒絕”效果。講師現在將再次向您提出一系列問題，來評估您是否理解此策略將允許和拒絕哪些操作。

活動：IAM 策略分析 (2/3) – 答案

考慮此 IAM 策略，然後在顯示問題時作答。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

1. 該策略是否允許您隨時無條件地終止任何 EC2 實例？
   - 答案：否。第一個語句對象允許。但是，第二個語句物件應用了一個條件。
2. 您是否有權從任何地方進行終止實例調用？
   - 答案：否。您只能從 *aws:SourceIp* 中指定的兩個 IP 位址範圍之一發出請求。
3. 如果您從分配的 IP 地址為 *192.0.2.243* 的伺服器進行調用，您能否終止實例？
   - 答案：能，因為 192.0.2.0/24 無類域間路由 (CIDR) IP 地址範圍包括從 192.0.2.0 到 192.0.2.255 的 IP 地址。CIDR 到 IP 範圍工具等資源可用於計算 CIDR 塊的範圍。

答案揭曉。

內容說明：JSON 格式的策略文檔示例。顯示一個包含兩個部分的語句區域。第一部分顯示對全部資源的操作 EC2:TerminateInstance 實施“允許”效果。第二部分顯示在 NotIpAddress aws:SourceIp 192.0.2.0/24 和 203.0.113.0/24 條件下，對全部資源的操作 EC2:TerminateInstances 實施“拒絕”效果。**內容說明結束。**

活動：IAM 策略分析 (3/3)

考慮此 IAM 策略，然後回答問題。

```
{
  "Version": "2012-10-17",
  "Statement":[{
      "Condition": {
        "StringNotEquals": {
           "ec2:InstanceType": [
              "t2.micro",
              "t2.small“
           ]
        }
      },
      "Resource": "arn:aws:ec2:*:*:instance/*"
      "Action": [
         "ec2:RunInstances",
         "ec2:StartInstances“
      ],
     "Effect": "Deny“
   }
]
```

1. 該策略允許哪些操作？
2. 假設該策略包含一個額外的語句物件，如本例所示：

```
{
     "Effect": "Allow",
     "Action": "ec2:*",
     "Resource": "*"
}
```

   該策略將如何限制此額外語句授予您的存取權限？

3. 如果該策略同時包含左側的語句和問題 2 中的語句，您是否能終止帳戶中存在的 m3.xlarge 實例？

觀察第三個、也是最後一個 IAM 策略文檔示例。講師現在將再次向您提出一系列問題，來評估您是否理解此策略將允許和拒絕哪些操作。

活動：IAM 策略分析 (3/3)

考慮此 IAM 策略，然後回答問題。

```
{
  "Version": "2012-10-17",
  "Statement":[{
      "Condition": {
        "StringNotEquals": {
           "ec2:InstanceType": [
              "t2.micro",
              "t2.small"
           ]
        }
      },
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Action": [
         "ec2:RunInstances",
         "ec2:StartInstances"
      ],
     "Effect": "Deny"
   }
]
```

1. 該策略允許哪些操作？
   - 答案：它不允許您執行任何操作（效果是*拒絕*）。
2. 假設該策略包含一個額外的語句物件，如本例所示：

   ```
   {
        "Effect": "Allow",
        "Action": "ec2:*",
        "Resource": "*"
   }
   ```

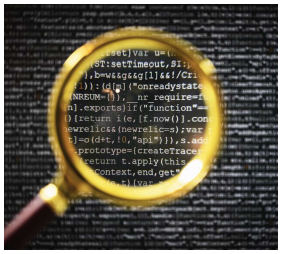
   該策略將如何限制此額外語句授予您的存取權限？
   - 答案：您將具有對 Amazon EC2 服務的完全存取權限。但是，您只能啟動實例類型為 t2.micro 或 t2.small 的 EC2 實例。
3. 如果該策略同時包含左側的語句和問題 2 中的語句，您是否能終止帳戶中存在的 m3.xlarge 實例？
   - 答案：是。

答案揭曉。

AWS CloudTrail

AWS CloudTrail

- 記錄和監控使用者活動
- 提供 AWS 帳戶的事件歷史記錄
  - 通過 AWS 管理主控台、 SDK、 AWS CLI 進行的操作
  - 提高用戶和資源活動的可見性
  - 默認情況下免費提供 90 天的事件歷史記錄
- 可以確定以下資訊
  - *誰*訪問過您的帳戶
  - *何時*以及從*何處*訪問的
  - 他們對 AWS 服務執行了*哪些*操作
- 用於以下用途的有用工具
  - 執行安全分析
  - 發現哪些調用被阻止（例如，被 IAM 策略阻止）



AWS CloudTrail 是一項可實現您的 AWS 帳戶的監管、合規性檢查和審計的服務。借助 CloudTrail，您可以持續監控並保留與 AWS 基礎設施中的操作相關的帳戶活動。它可提供帳戶活動的事件歷史記錄，包括通過 AWS 管理主控台、AWS SDK 和命令列工具執行的操作。事件歷史記錄可以簡化安全性分析、資源更改跟蹤和故障排除工作。

您可以通過捕捉特定時段內在您的 AWS 帳戶中所發生更改的全面歷史記錄，發現並解決安全性和運行問題。您可以識別調用 AWS 的用戶和帳戶、發起調用的源 IP 地址以及執行調用的時間。借助 CloudTrail，您能夠跟蹤並自動應對威脅 AWS 資源安全性的帳戶活動。

通過與 Amazon EventBridge（以前稱為 Amazon CloudWatch Events）的集成，您可以定義在檢測到可能導致安全性漏洞的事件時要運行的工作流。例如，您可以創建一個工作流，以在 CloudTrail 記錄將導致某個 S3 存儲桶被公開的 API 調用時將特定策略添加到該存儲桶。

CloudTrail 可記錄每個操作的重要資訊，包括請求的發出方、使用的服務、執行的操作、操作的參數，以及 AWS 服務返回的回應元素。該服務還有助於企業滿足必須遵守的合規性和審計要求。

第 2 節要點

- 避免使用帳戶根用戶執行常見任務。相反，請創建和使用 IAM 用戶憑證。
- 用於訪問 AWS 帳戶資源的權限在一個或多個 IAM 策略文檔中定義。
  - 將 IAM 策略附加到 IAM 用戶、組或角色。
- 當 IAM 確定許可權時，顯式拒絕將始終覆蓋任何允許語句。
- 在授予存取權限時，遵循最低許可權原則是最佳實踐。

本模組中這節內容的要點包括：

- 避免使用帳戶根用戶執行常見任務。相反，請創建和使用 IAM 用戶憑證。
- 用于訪問 AWS 帳戶資源的許可權在一個或多個 IAM 策略文檔中定義。
  - 將 IAM 策略附加到 IAM 用戶、組或角色。
- 當 IAM 確定許可權時，顯式拒絕將始終覆蓋任何允許語句。
- 在授予存取權限時，遵循最低許可權原則是最佳實踐。


第 3 節：組織使用者
模組 8：確保使用者和應用程式訪問安全
aws
© 2023，Amazon Web Services, Inc. 或其聯屬公司。保留所有權利。

介紹第 3 節：組織使用者。

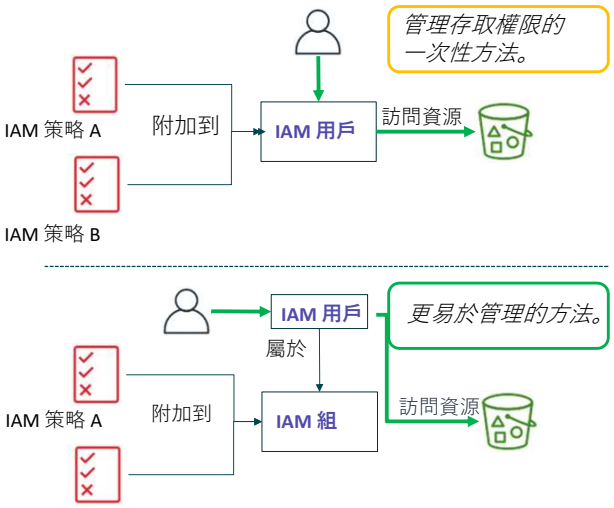
IAM 組

使用 IAM 組向多個用戶授予相同的存取權限。

- 組中的所有用戶將繼承分配到該組的許可權
  - 使多個用戶的存取權限管理變得更加輕鬆

提示：結合多種方法可以精細地管理個人存取權限

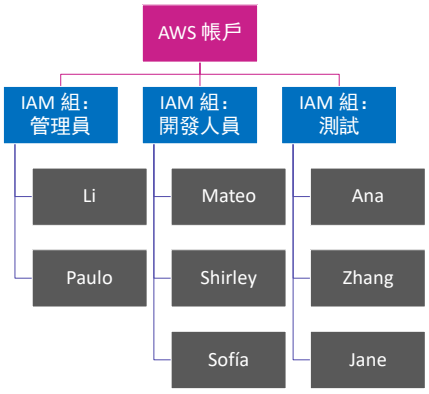
- 將用戶添加到組，以根據工作職能應用標準存取權限
- 可選擇將額外策略附加到需要例外情況的使用者



IAM 組是 IAM 使用者的集合。組是一種便利的方法，能讓您更加輕鬆地管理使用者集合的許可權，而不必分別管理每個用戶的許可權。

將組成員作為一個簡單列表來進行管理：
- 向組添加用戶或將用戶從組中刪除。
- 一個用戶可以屬於多個組。
- 一個組無法從屬於其他組。
- 可以通過使用存取控制策略授予組許可權。
- 組沒有安全憑證，也無法直接訪問 Web 服務，它們存在的目的只是為了更輕鬆地管理用戶許可權。

IAM 組示例



提示：創建反映工作職能的組

- 在雇用新的開發人員之後，將他們添加到*開發人員*組
  - 立即繼承已授予其他開發人員的相同存取權限
- 如果 Ana 擔任新的開發人員角色 –
  - 將她從*測試*組中移除
  - 將她添加到*開發人員*組
- 用戶可以屬於多個組
  - 但是，將應用最嚴格的策略

通常，您需要創建反映工作職能的組。例如，您可以為管理員創建一個組，為開發人員創建另一個組，為執行測試功能的團隊再創建一個組。

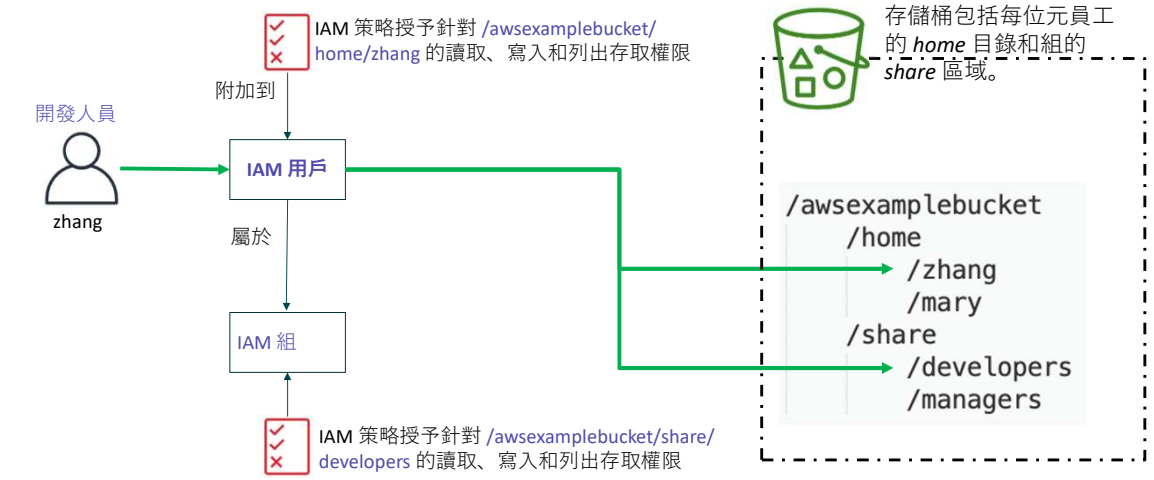
然後，您可以將一個或多個策略檔附加到每個組，然後向組中添加用戶。憑藉其組成員資格，用戶會擁有分配給他們所在的一個或多個組的存取權限。

在雇用新的開發人員之後，您可以將他們添加到現有開發人員組。他們將獲得與其他開發人員相同的存取權限。

如果某個人員，例如 Ana（如示例中所示）在企業中擔任了一個新的角色，則可以將其從*測試*組中移除，然後將其添加到*開發人員*組中。或者，如果 Ana 將執行這兩項職能，您可以將她留在*測試*組中，然後將她添加到*開發人員*組中。

如果您發現開發人員需要訪問帳戶中的其他資源，則可以更新*開發人員*組的策略或添加策略。該組的所有成員都將獲得這種額外的存取權限級別。通過組功能，可以更加輕鬆地在團隊之間維護一致的存取權限。

Amazon S3 上的 IAM 使用案例



此示例演示如何在 S3 存儲桶上配置 IAM 許可權。

awsexamplebucket 有兩個主要目錄。*home* 目錄為每個使用者提供子目錄，讓他們可以在其中存儲個人工作成果。在 *share* 目錄包含的子目錄中，不同團隊都可以在其中存儲內容。

如果新的團隊成員 *zhang* 以開發人員身份加入企業，您可以採取三項操作來向其授予適當的存取權限。

首先，將 *zhang* 添加到面向開發人員的 IAM 組中。請注意，該組附加有一個 IAM 策略，用於授予針對 */awsexamplebucket/share/developers* 的存取權限。

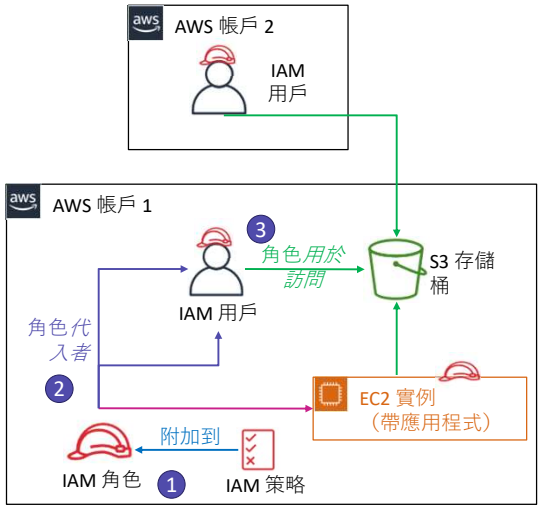
接下來，在 Amazon S3 中創建 */awsexamplebucket/home/zhang* 目錄。

最後，附加一個 IAM 策略，用於將對 */awsexamplebucket/home/zhang* 目錄的存取權限直接授予 *zhang* IAM 用戶。Zhang 的存取權限將包括從該組授予的許可權以及直接附加到 IAM 用戶主體的許可權。

介紹第 4 節：用戶聯合身份管理。

IAM 角色

- IAM 角色特性
  - 提供*臨時*安全憑證
  - 不是唯一地與一個人相關聯
  - 可由人員、應用程式或服務*代入*
  - 通常用於委派存取權限
- 使用案例
  - 為 AWS 資源提供對 AWS 服務的存取權限
  - 為經過外部身份驗證的用戶提供存取權限
  - 為第三方提供存取權限
  - 切換角色以訪問以下位置的資源 –
    - 您的 AWS 帳戶
    - 任何其他 AWS 帳戶（跨帳戶訪問）



IAM 角色使您能夠定義一組許可權，以便訪問使用者或服務所需的資源。但是，許可權並不附加到 IAM 用戶或組。而是附加到角色，而角色由使用者或服務代入。

當使用者代入角色時，系統會暫時忘記使用者之前所具有的許可權。AWS 返回臨時安全憑證，隨後使用者或應用程式可以使用這些憑證以程式設計方式對 AWS 發出請求。

通過使用 IAM 角色，您不必與每個需要訪問資源的實體共用長期安全憑證（例如創建 IAM 用戶）。

對於像 Amazon EC2 這樣的服務，應用程式或 AWS 服務可以在運行時以程式設計方式代入角色。

代入該角色的主體也可以是來自其他 AWS 帳戶的 IAM 用戶、組或角色，包括不歸您所有的帳戶。

通過創建用於外部帳戶訪問的角色，您無需為協力廠商管理用戶名和密碼。如果您不再希望某人或某個系統具有存取權限，可以修改或刪除該角色。因此，您不需要為企業之外的人員創建和管理帳戶。

演示：EC2 實例設定檔

現在，講師可能會選擇演示如何將 IAM 角色附加到 EC2 實例。此角色將 AWS 資源存取權限授予應用程式。

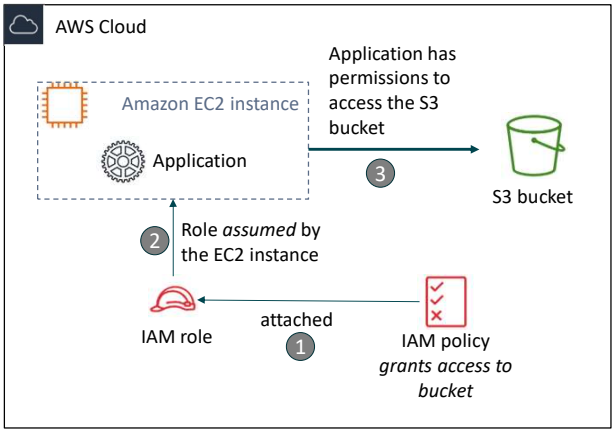
Summary: EC2 instance profile demonstration

Scenario:

- An application that runs on an EC2 instance needs access to an S3 bucket

Solution:

- Define an IAM policy that grants access to the S3 bucket
- Attach the policy to a role
- Allow the EC2 instance to assume the role



This diagram illustrates the educator-led demonstration.

- An application runs on an EC2 instance, and that application needs access to the S3 bucket.
- An administrator creates an *IAM role*.
- Then, they create an *IAM policy* that grants read-only access to the specified S3 bucket. The policy also includes a trust policy that allows the EC2 instance to assume the role and retrieve the temporary credentials.
- Next, they attach the IAM policy to the role.

When the application runs on the instance, it can assume the role and use the role's temporary credentials to access the bucket.

With this architecture, the administrator does not need to directly grant the application developer permission to access the bucket, and the developer never needs to share or manage credentials.

授予代入角色的許可權

AWS Security Token Service (AWS STS)

- 要使 IAM 使用者、應用程式或服務代入某個角色，您必須授予切換到該角色的許可權
- AWS Security Token Service (AWS STS)
  - 允許您請求臨時、有限許可權憑證的 Web 服務
  - 憑證可供 IAM 用戶使用，也可供您驗證的用戶（聯合身份用戶）使用
- 示例策略 – 允許 IAM 用戶代入某個角色

```
{
   "Version": "2012-10-17",
   "Statement": {
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource": "arn:aws:iam::123456789012:role/Test*"
   }
}
```


AWS Security Token Service 也稱為 *AWS STS*。它是一項 Web 服務，使 IAM 使用者、聯合身份使用者或應用程式能夠代入他們需要的 IAM 角色。

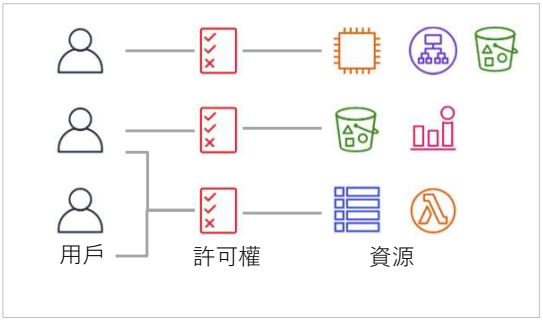
成功調用 AWS STS API 的 AssumeRole 操作後，Web 服務將返回 IAM 使用者或經過聯合身份驗證的用戶所請求的臨時受限許可權憑證。通常，AssumeRole 操作用於跨帳戶訪問或聯合身份訪問。

本示例策略允許 IAM 用戶代入在 AWS 帳號 123456789012 中定義的任何角色，只要角色名稱以 *Test* 開頭。

基於*角色*的存取控制 (RBAC)

傳統的存取控制方法：

- 根據工作職能授予使用者特定許可權（例如資料庫管理員）
- 為每個許可權組合創建不同的 IAM 角色
- 通過為每個新資源添加存取權限來更新許可權（持續更新策略可能會非常耗時）



現在，您要考慮兩種不同的存取控制方法：基於角色的存取控制 (RBAC) 和基於屬性的存取控制 (ABAC)。您首先將瞭解 RBAC。

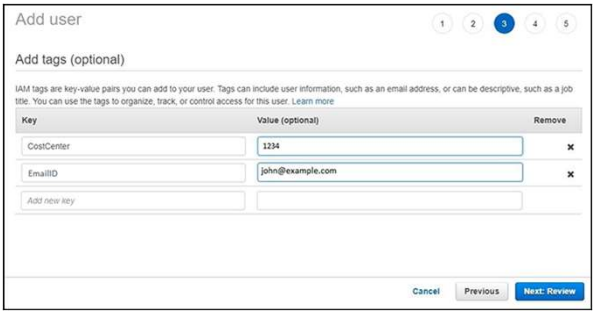
RBAC 一直在本地部署和雲中使用。使用這個模型，您可以向使用者授予對一組許可權的顯式存取權限。假設您擁有資料庫管理員、網路系統管理員和開發人員。如果您的一個或多個網路系統管理員也是開發人員，則您不會創建新策略來授予這些許可權。您會將這些用戶添加到兩個角色中。

這種方法很常見，而且具有許多優點。但是，在此模型中，維護許可權的人員可能會發現，每次創建新資源時，他們必須不斷更新許可權檔來添加對特定角色的存取權限。例如，每當有人創建新資源並希望允許用戶訪問該資源時，他們都必須使用 ARN 更新策略。

最佳實踐：標記

- 標籤由名稱和（可選）值組成
  - 可應用於您的 AWS 帳戶中的資源
  - 很多不同的 API 操作返回標籤鍵及值
- 定義*自定義*標簽
- 多種實際用途
  - 賬單、篩選視圖、存取控制等
- 應用於 EC2 實例的示例標籤：
  - Name = web server
  - Project = unicorn
  - Stack = dev

- 標籤還可以應用於 IAM 用戶或 IAM 角色，例如 –



在考慮第二種許可權控制方法之前，您應該瞭解 AWS 中的標記功能。

AWS 使客戶能夠以*標簽*的形式將中繼資料分配給其 AWS 資源和身份。每個標籤都是一個簡單的標記，由一個客戶定義的鍵和一個可選值組成。利用標籤可以更輕鬆地管理、搜索和篩選資源。

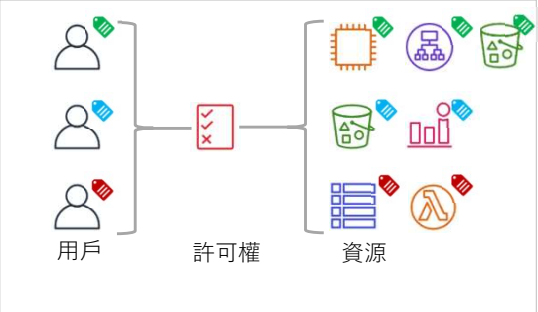
標籤有許多實際用途。例如，您可以創建*技術標籤*，來標識某項資源是 Web 伺服器、是特定專案的一部分、是特定環境（測試、開發或生產）的一部分等。您還可以創建*業務標簽*，用來標識應為此資源或此資源所屬專案付費的部門或成本中心。最後，您還可以設置*安全標籤*，例如用於資源支援的特定資料機密級別的識別字。

您最多可以為每個資源創建 個標簽。對於每個資源，每個標籤鍵都必須是唯一的，每個標籤鍵只能有一個值。標簽鍵和值區分大小寫。

您還可以向 IAM 用戶和 IAM 角色添加標籤。標籤是您接下來將學習的第二種存取控制方法的一個重要組成部分。

基於*屬性*的存取控制 (ABAC)

- 高度可擴展的存取控制方法
  - 屬性是鍵或鍵值對，例如標籤
  - 示例屬性 –
    - Team = Developers
    - Project = Unicorn
- 使用 ABAC 的許可權（策略）規則比使用 RBAC 的許可權（策略）規則更容易維護
- 益處
  - 根據屬性自動應用許可權
  - *無需*對每個新用戶或資源進行許可權更新，即可實現精細許可權
  - 完全可審計



現在，您已經瞭解了標記功能，您將瞭解第二種存取控制方法：基於屬性的存取控制 (ABAC)。

通過 ABAC，您可以使用屬性創建隨企業規模而擴展的常規許可權規則。

在此模型中，IAM 使用者具有您所創建和應用的屬性，例如一個或多個標籤。

資源還具有同樣應用于資源的屬性，例如匹配標籤。

通過 RBAC 方法，編寫許可權變得相對簡單。策略用來檢查應用於 IAM 使用者的屬性是否也被應用於他們要訪問的資源。在創建新的 IAM 用戶和新的帳戶資源時，您要將正確的標籤應用于用戶和資源。

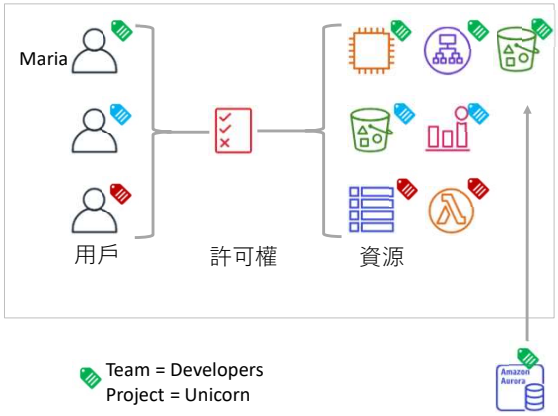
通過 ABAC 方法，您可以向開發人員授予對其專案資源的存取權限，但無需在策略檔中指定資源。

您可以想像一下 ABAC 訪問管理方法的可擴展性如何。您無需修改許可權設置。使用正確的標籤創建資源或用戶時，許可權將自動應用。

將 ABAC 應用於您的企業

如何將 ABAC 應用於您的企業：

1. 設置身份的存取控制屬性
2. 需要新資源的屬性
3. 根據屬性配置許可權
4. 測試
   a) 創建新資源
   b) 驗證權限是否自動應用



要將 ABAC 應用於您的企業，第一步是創建身份，例如 IAM 用戶或 IAM 角色。這些身份必須具有將用於存取控制目的的屬性。例如，您可以將 *Team = Developers* 和 *Project = Unicorn* 標籤應用於 *Maria* 用戶。

接下來，需要為新資源提供屬性。您應該創建強制實施規則的策略。例如，您可以要求在創建任何資源時將 *Project* 屬性和 *Team* 屬性應用于資源。

第三步，根據屬性配置存取權限。例如，假設 IAM 用戶具有 *Project = Unicorn* 和 *Team = Developers* 標籤。如果該用戶嘗試訪問在這兩個相同標籤上具有匹配值的資源，該策略將允許訪問。否則，該策略將拒絕訪問。

第四步，測試配置。例如，您可以嘗試創建 Amazon Aurora 資料庫實例但不要包含必需的標籤。創建操作應該會失敗。再次嘗試創建資料庫實例但包含必需的標籤。這次，您應該能夠成功創建資源。最後，您可以嘗試以 *Maria* 使用者身份訪問資料庫實例。您應該可以成功訪問。但是，如果您嘗試以沒有匹配標籤的其他使用者身份訪問資料庫實例，則您的訪問應該會遭到拒絕。

外部驗證的用戶

聯合身份

- 由 AWS 帳戶外的系統完成的使用者身份驗證
  - 示例：公司目錄
- 這種方法允許通過現有身份進行訪問而無需創建 IAM 用戶

聯合身份選項

1. AWS STS
   - 公有身份服務提供者 (IdP)
   - 自訂身份代理應用程式
2. 安全斷言標記語言 (SAML)
3. Amazon Cognito

IdP 身份驗證概覽

使用者通過應用程式訪問身份代理 | 身份代理對使用者進行身份驗證 | 從 AWS STS 請求臨時憑證 | 臨時憑證返回到應用程式

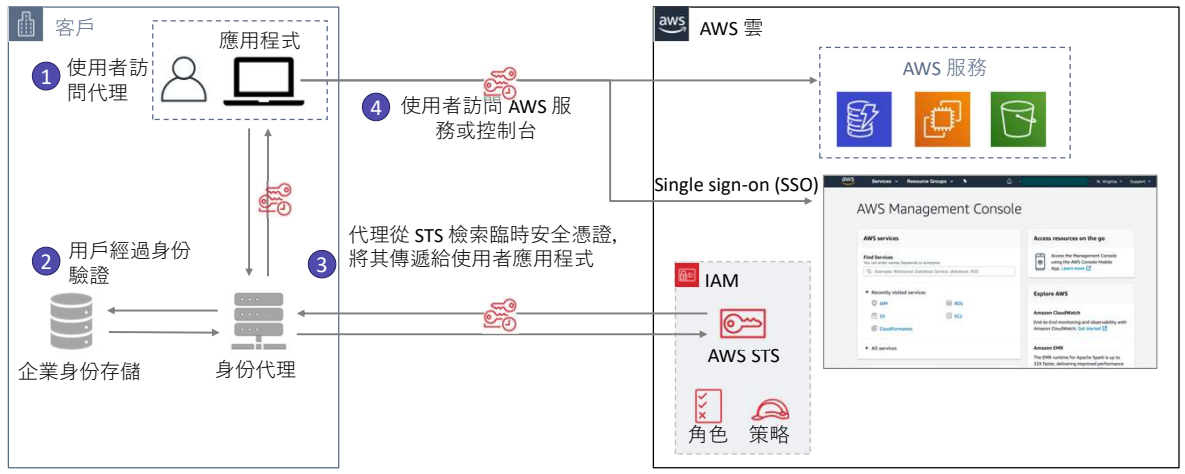
現在，您將瞭解一個新的主題：外部驗證的用戶。

IAM 支持聯合身份，用於實現對 AWS 管理主控台或 AWS API 的委託訪問。通過聯合身份，外部身份可以安全訪問您 AWS 帳戶中的資源，而*無需*創建 IAM 用戶。

該圖顯示了使用*身份提供商 (IdP)* 為使用者或應用程式創建臨時憑證時發生的四個主要步驟。

聯合身份可以通過三種方法來實現。第一種方法是使用公司 IdP（例如 Microsoft Active Directory）或自訂身份代理應用程式。每個選項都使用 AWS STS。第二種方法是創建使用安全斷言標記語言 (SAML) 的集成。第三種方法是使用 Web 身份提供商，例如 Amazon Cognito。接下來幾張幻燈片將討論這三種方法。

使用身份代理的聯合身份

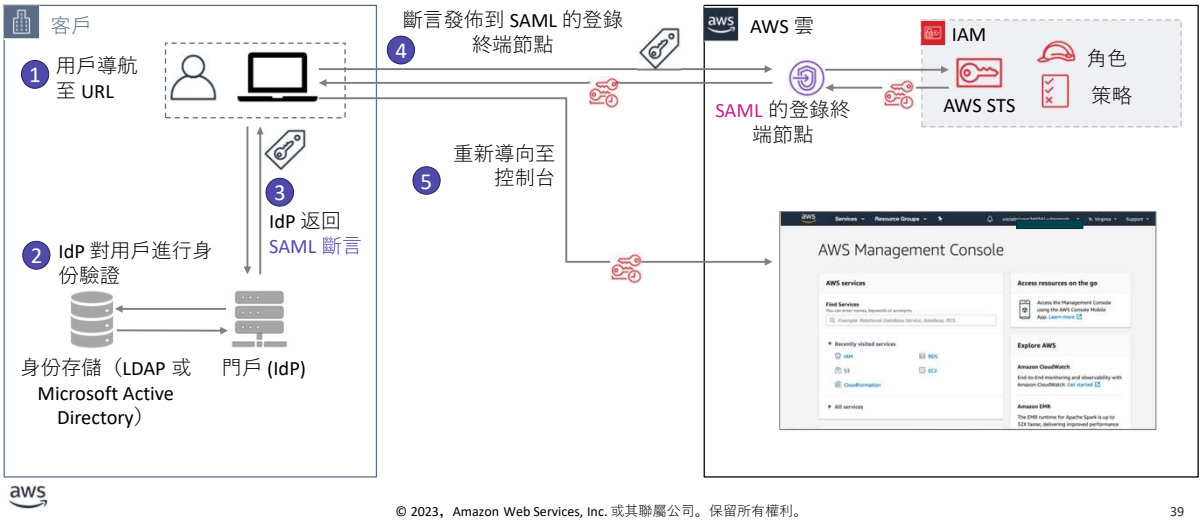


現在，您將學習如何使用身份代理來完成聯合身份。

此過程包括以下步驟：

1. 用戶訪問應用程式。使用者輸入其使用者 ID 和密碼並提交
2. 身份代理接收身份驗證請求。然後，它與公司身份存儲（可能是 Microsoft Active Directory 或輕量級目錄訪問協定 (LDAP) 伺服器）進行通信。
3. 如果身份驗證請求成功，身份代理將向 AWS STS 發出請求。請求的內容是為使用者應用程式檢索臨時 AWS 安全憑證。
4. 用戶應用程式接收臨時 AWS 安全憑證，並將使用者重新導向到 AWS 管理主控台。用戶無需使用另一組憑證集即可直接登錄 AWS。此過程是 Single-Sign On (SSO) 實現的一個示例。如果 IAM 策略文檔允許，使用者應用程式還可以使用這些臨時 AWS 安全憑證訪問 AWS 服務。

使用 SAML 的聯合身份



現在，您將瞭解實現聯合身份驗證的第二個選項。此方法使用 *SAML* 開放標準在 IdP 和服務提供者之間交換身份驗證和授權資料。

此過程包括以下步驟：

1. 您企業內的使用者導航至網路中的內部門戶。該門戶也充當 IdP，處理您的企業與 AWS 之間的 SAML 信任。
2. IdP 根據身份存儲（可能是 LDAP 伺服器或 Microsoft Active Directory）對用戶的身份進行身份驗證。
3. 門戶從 IdP 接收身份驗證回應作為 *SAML 斷言*。
4. 用戶端將 SAML 斷言發佈到 SAML 的 AWS 登錄終端節點。該終端節點與 AWS STS 通信，並且它調用 AssumeRoleWithSAML 操作來請求臨時安全憑證並構建登錄 URL。
5. 客戶端收到臨時 AWS 安全憑證。客戶端被重新導向到 AWS 管理主控台，並使用臨時 AWS 安全憑證進行身份驗證。

Amazon Cognito

Amazon Cognito

Amazon Cognito 是一項完全託管式的服務。

- 它為 Web 和移動應用程式提供身份驗證、授權和用戶管理
- Amazon Cognito 提供 Web 聯合身份
  - 它們可以用作身份代理，支援與 OpenID Connect (OIDC) 相容的 IdP
- 聯合身份
  - 用戶使用社交身份提供商（Amazon、Facebook、Google）或 SAML 登錄
- 用戶池
  - 您可以使用使用者設定檔身份驗證權杖維護目錄

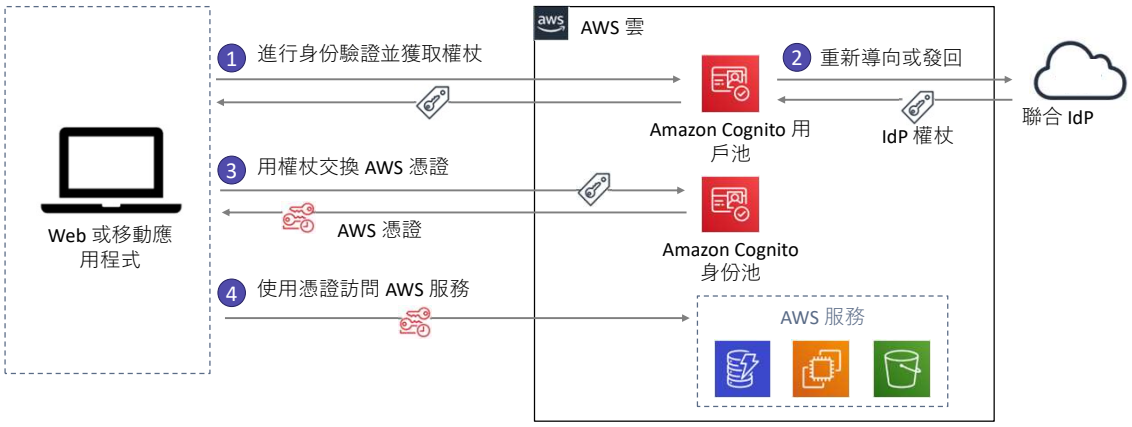
第三個也是最後一個聯合身份驗證選項是使用 Amazon Cognito。*Amazon Cognito* 是一項為 Web 和移動應用程式提供身份驗證、授權和用戶管理的完全託管式服務。使用者可以使用用戶名和密碼直接登錄，或通過 Facebook、Amazon 或 Google 等協力廠商登錄。

Amazon Cognito 的兩個主要組件是*用戶池*和*身份池*。

*用戶池*是 Amazon Cognito 中的使用者目錄。利用用戶池，用戶可以通過 Amazon Cognito 登錄 Web 或移動應用程式。他們還可以通過協力廠商 IdP 進行聯合。用戶池的所有成員都有一個可通過 SDK 訪問的目錄設定檔。

借助*身份池*，您可以為用戶創建唯一身份並分配許可權。借助身份池，用戶可以獲取臨時 AWS 憑證來訪問 AWS 服務或資源。身份池可以通過 Facebook、Google、Login with Amazon 以及 OpenID Connect (OIDC) 提供商與 Amazon Cognito 用戶池的社交登錄進行通信。

Amazon Cognito 示例



在這種情況下，目標是使用 Amazon Cognito 對使用者進行身份驗證，然後向該使用者授予其他 AWS 服務的存取權限。

- 在第 1 步中，應用程式使用者通過 Amazon Cognito 使用者池登錄，在成功通過身份驗證後，收到用戶池權杖。
- 接下來，該應用程式通過 Amazon Cognito 身份池使用用戶池權杖換取 AWS 憑證。
- 最後，應用程式使用者使用這些 AWS 憑證訪問其他 AWS 服務。

第 4 節要點

- IAM 角色提供可由人員、應用程式或服務代入的臨時安全憑證
- 您可以通過 AWS Security Token Service (AWS STS) 請求臨時 AWS 憑證
- 在聯合身份中，用戶身份驗證是在 AWS 帳戶外部進行的
  - 通過使用 AWS STS、SAML 或 Amazon Cognito 完成

本模組中這節內容的要點包括：

- IAM 角色提供可由人員、應用程式或服務代入的臨時安全憑證。
- 您可以通過 AWS Security Token Service (STS) 請求臨時 AWS 憑證。
- 在聯合身份驗證中，用戶身份驗證是在 AWS 帳戶外部發生的。
  - 使用 STS、SAML 或 Amazon Cognito 完成。


第 5 節：多個帳戶
模組 8：確保使用者和應用程式訪問安全
aws
© 2023，Amazon Web Services, Inc. 或其聯屬公司。保留所有權利。

介紹第 5 節：多個帳戶。

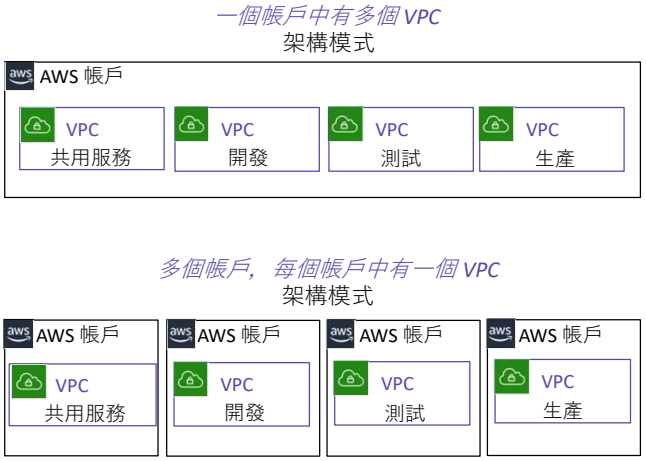
一個帳戶還是多個帳戶？

兩種架構模式

- 大多數企業選擇創建多個帳戶

多個帳戶的優勢

- 隔離業務單元或部門
- 隔離開發、測試和生產環境
- 隔離審計資料、恢復資料
- 針對受監管的工作負載使用單獨的帳戶
- 更容易針對每個業務單元的消耗觸發成本警報



當您使用 AWS 支援企業中的不同團隊和部門時，您可以在兩種常規架構模式之間進行選擇，以隔離和分隔每個團隊使用的資源。

第一種模式是在單個 AWS 帳戶中定義多個 Virtual Private Cloud (VPC)。如果您傾向於開支最小的集中式資訊安全管理方式，則可以選擇使用單個 AWS 帳戶。

第二種模式是創建多個 AWS 帳戶並在每個帳戶中定義 VPC。實際上，大型和小型企業都傾向於為其企業創建多個帳戶。例如，他們可能會為各個業務單元創建各自的帳戶。他們還可以為其開發、測試和生產資源創建單獨的帳戶。

當客戶為開發和生產資源使用不同的 AWS 帳戶時（通常採用整合帳單方式），可以明確區分不同類型的資源。同時也可提供一些安全優勢。

或者，如果貴公司針對生產、開發和測試分別創建了不同的環境，那麼您可以配置三個 AWS 帳戶，使每個環境都擁有獨立帳戶。此外，如果您有多個自主部門，也可以為企業的每個自主部門創建不同的 AWS 帳戶。

當您使用多個帳戶時，更有效的策略是為通用專案資源創建一個 AWS 帳戶。通用資源可能包括網域名稱系統 (DNS) 服務、Microsoft Active Directory 和內容管理系統 (CMS)。您還可以為自主專案或部門創建單獨的帳戶。這種策略使您可以在每個部門或專案帳戶下分配許可權和策略，並跨帳戶授予資源存取權限。

管理多個帳戶遇到的挑戰

- 跨帳戶的安全管理
  - IAM 策略複製
- 創建新帳戶
  - 涉及許多手動流程
- 帳單整合
- 需要集中監管以確保一致性

儘管大多數組織選擇使用多個 AWS 帳戶，但這種選擇會帶來一些挑戰。

首先，您必須確定如何有效管理所有帳戶的安全性。如果您複製所有帳戶中定義的 IAM 策略來確保一致性，則可能涉及自訂自動化、手動操作，或這兩者都要涉及。

此外，您可能會不斷地被要求創建更多帳戶。手動創建這些帳戶需要一些時間。也可能難以跟蹤所有帳戶和每個帳戶的目的。

確定企業中的哪個成本中心應該為哪個帳戶中的哪些資源付費也是一項挑戰。最後，您可能還希望實現確保一致性所需的集中管理。

使用 AWS Organizations 管理多個帳戶

跨多個 AWS 帳戶集中管理和強制實施策略

- 基於組的賬戶管理
- 對 AWS 服務的基於策略的訪問
- 賬戶自動創建和管理
- 整合帳單
- 基於 API

AWS 提供了一項服務，旨在解決這些管理難題。

AWS Organizations 是一項用於帳戶管理的託管服務。組織是您創建的用於整合、集中查看和管理所有 AWS 帳戶的實體。您可以通過啟用的功能集來確定組織的功能。

Organizations 有助於您管理多個 AWS 帳戶的策略。您可以使用該服務創建帳戶組。然後，您可以將策略附加到組，以便在整個帳戶中應用正確的策略。

您可以創建 AWS 帳戶組，然後對每個組應用不同的策略。

Organizations API 能夠以程式設計方式創建新帳戶，並將其添加到組中。附加到該組的策略將自動應用到新帳戶。

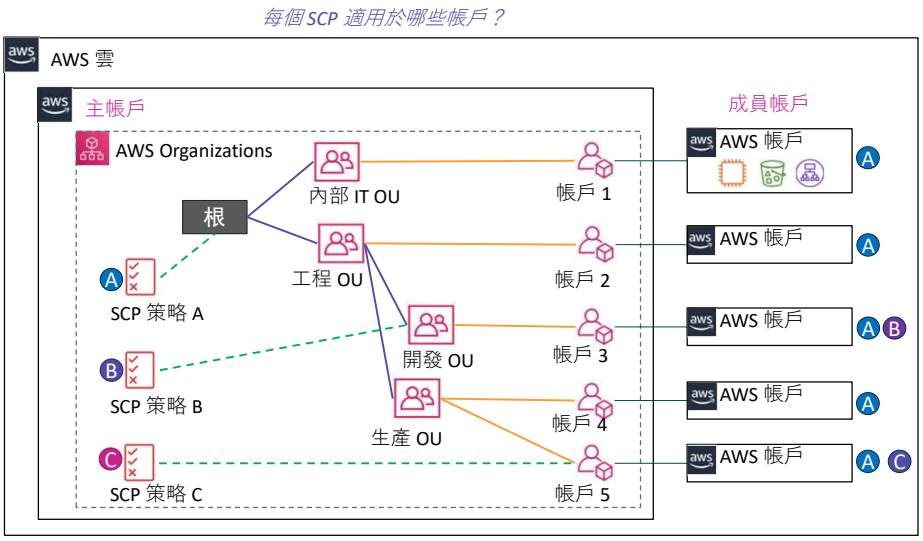
您也可以通過整合帳單，為組織中的所有 AWS 帳戶設置單一付款方式。通過整合帳單，您可以在一個綜合視圖中查看所有帳戶產生的費用。

最後，您可以在 API 級別管理 AWS 服務的使用。例如，您可以將策略應用到一組帳戶，僅允許這些帳戶中的 IAM 使用者從 S3 存儲桶中讀取資料。

AWS Organizations：圖解

在 AWS Organizations 主帳戶中：

1. 創建組織單位 (OU) 層次結構
2. 將帳戶作為成員帳戶分配給 OU
3. 定義將許可權限制應用于特定成員帳戶的服務控制策略 (SCP)
4. 將 SCP 附加到根用戶、OU 或帳戶



這是一個 AWS 組織的示例。它在常規 AWS 帳戶中定義，該帳戶就是幻燈片中的*主帳戶*，因為 AWS 組織是在該帳戶中定義的。

當您在主帳戶中創建*組織*時，組織會自動創建名為*根*的父容器。然後，您可以在組織中的每個根下定義*組織單位*，也稱為 *OU*。每個 OU 都是*成員帳戶*的容器。一個 OU 還可以包含其他 OU，而這些 OU 可以包含更多帳戶。此功能使您能夠創建樹狀層次結構。您可以把根和 OU 視為帳戶中伸出和結束的分支，就像枝葉一樣。

為了跨帳戶配置存取控制，接下來您要定義*服務控制策略 (SCP)*。將每個策略附加到 OU 和帳戶層次結構中的適當位置。該策略從根中流出，影響其下的所有 OU 和帳戶。因此，如果您將 SCP 應用於根（如示例中的 *SCP 策略 A*），它將適用於組織中的所有 OU 和帳戶。您可以將 SCP 附加到根、任何 OU 或單個帳戶。

請記住，與 IAM 策略一樣，SCP 只有在被明確允許且未被應用於用戶的任何其他 SCP 或 IAM 策略明確拒絕的情況下才會授予存取權限。例如，假設應用於組織根的 SCP 策略 A 對特定服務或資源集設置的限制比 SCP 策略 C 更多。那麼，帳戶 5 中的使用者需要遵守策略 A 設置的更嚴格的許可權。類似的，如果單個帳戶級別的任何 IAM 策略明確拒絕用戶的任何操作，則這些 IAM 策略將覆蓋 SCP 中授予該帳戶的任何許可權。

SCP 的示例用途

- 服務控制策略 (SCP) 的特性
  - 它們使您能夠控制成員帳戶中的 IAM 使用者可以訪問哪些服務
  - SCP 不能被本地管理員覆蓋
  - 在單個帳戶中定義的 IAM 策略仍然適用

- SCP 的示例用途
  - 創建*阻止*服務訪問或特定操作的策略
    - 示例：拒絕用戶在所有成員帳戶中禁用 AWS CloudTrail
  - 創建*允許*完全訪問特定服務的策略
    - 示例：允許對 Amazon EC2 和 CloudWatch 的完全存取權限
  - 創建*強制實施資源標記*的策略

服務控制策略 (SCP) 使您能夠控制成員帳戶中的 IAM 使用者可以訪問哪些服務。假設您有要跨多個帳戶應用的特定策略。相較於將這些許可權設置複製到每個帳戶的 IAM 策略文檔，在 SCP 中定義這些策略會更容易。

SCP 應與在每個帳戶中定義的 IAM 策略一起使用。您可以認為 SCP 能夠提供有關服務的一般邊界，以及應當允許或拒絕用戶訪問的一般許可權。然後，您可以使用 IAM 策略設置特定於單個帳戶的更精細的存取控制。

您可以創建旨在阻止（或拒絕）訪問某些服務的 SCP。您還可以定義旨在允許訪問某些服務的 SCP。最後，您可能會決定創建一個 SCP 來執行資源標記。這樣，當在您的帳戶中創建新資源時，您的存取控制或成本分配標記策略可以保持有效。

第 5 節要點

- 您可以使用多個 AWS 帳戶隔離業務單元、開發和測試環境、受監管的工作負載以及審計資料
- AWS Organizations 使您能夠配置自動帳戶創建和整合帳單
- 您可以使用服務控制策略 (SCP) 配置跨帳戶的存取控制

本模組中這節內容的要點包括:

- 您可以使用多個 AWS 帳戶隔離業務單元、開發和測試環境、受監管的工作負載以及審計資料
- AWS Organizations 使您能夠配置自動帳戶創建和整合帳單
- 您可以使用服務控制策略 (SCP) 配置跨帳戶的存取控制

模組 8 – 挑戰實驗：使用 IAM 控制 AWS 帳戶存取權限

現在，您將完成模組 8 – 挑戰實驗：使用 IAM 控制 AWS 帳戶訪問。

業務需求：用戶存取控制

咖啡館必須定義用戶在雲資源中應具有的存取權限級別。然後，他們必須在整個 AWS 賬戶中實施這些存取控制。

Mateo 最近造訪咖啡館時向 Sofía 介紹了 IAM 服務的功能。她打算使用 IAM 來實現自己的目標。

在與 Mateo 討論過咖啡館的 AWS 基礎設施後，Sofía 意識到，她必須解決一些關於咖啡館員工使用 AWS 帳戶的基本安全問題。

咖啡館現在已經小有規模，團隊成員各自分工（如開發人員或資料庫管理員），在 AWS 上構建、維護或訪問應用程式。不過，到目前為止，團隊還沒有根據使用者的角色和職責來明確定義應擁有的訪問級別。

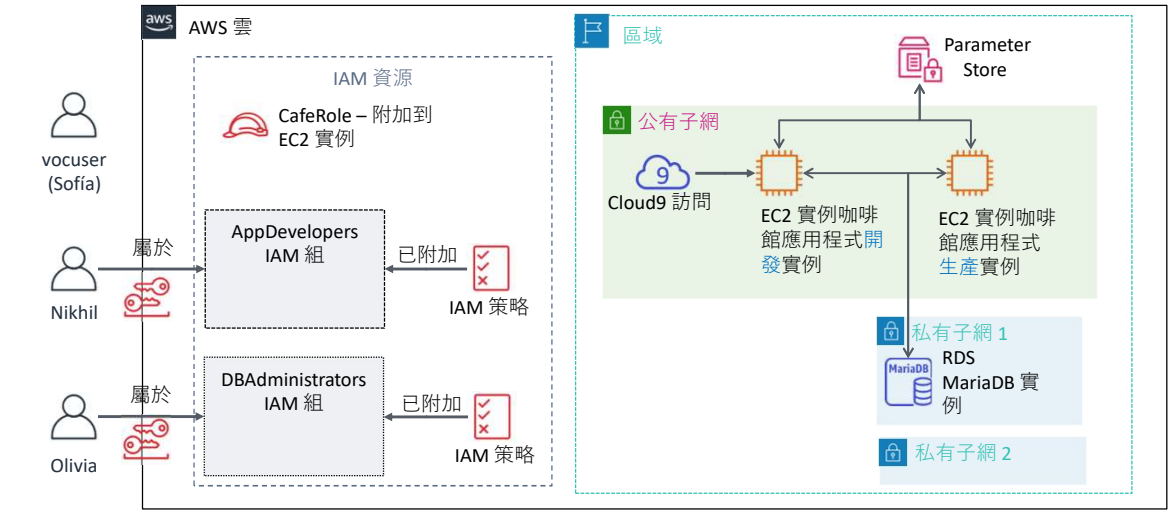
挑戰實驗：任務

1. 使用策略和 IAM 使用者配置 IAM 組
2. 以 Nikhil 的身份登錄並測試訪問
3. 配置有關資料庫管理員使用者存取權限的 IAM
4. 以資料庫管理員的身份登錄並解決資料庫連接問題
5. 使用 IAM 策略模擬器並通過視覺化編輯器創建自訂 IAM 策略

在本挑戰實驗中，您將完成以下任務：

1. 使用策略和 IAM 使用者配置 IAM 組
2. 以 Nikhil 的身份登錄並測試訪問
3. 配置有關資料庫管理員使用者存取權限的 IAM
4. 以資料庫管理員的身份登錄並解決資料庫連接問題
5. 使用 IAM 策略模擬器並通過視覺化編輯器創建自訂 IAM 策略

挑戰實驗：最終產品



該圖總結了您完成實驗後將構建的內容。

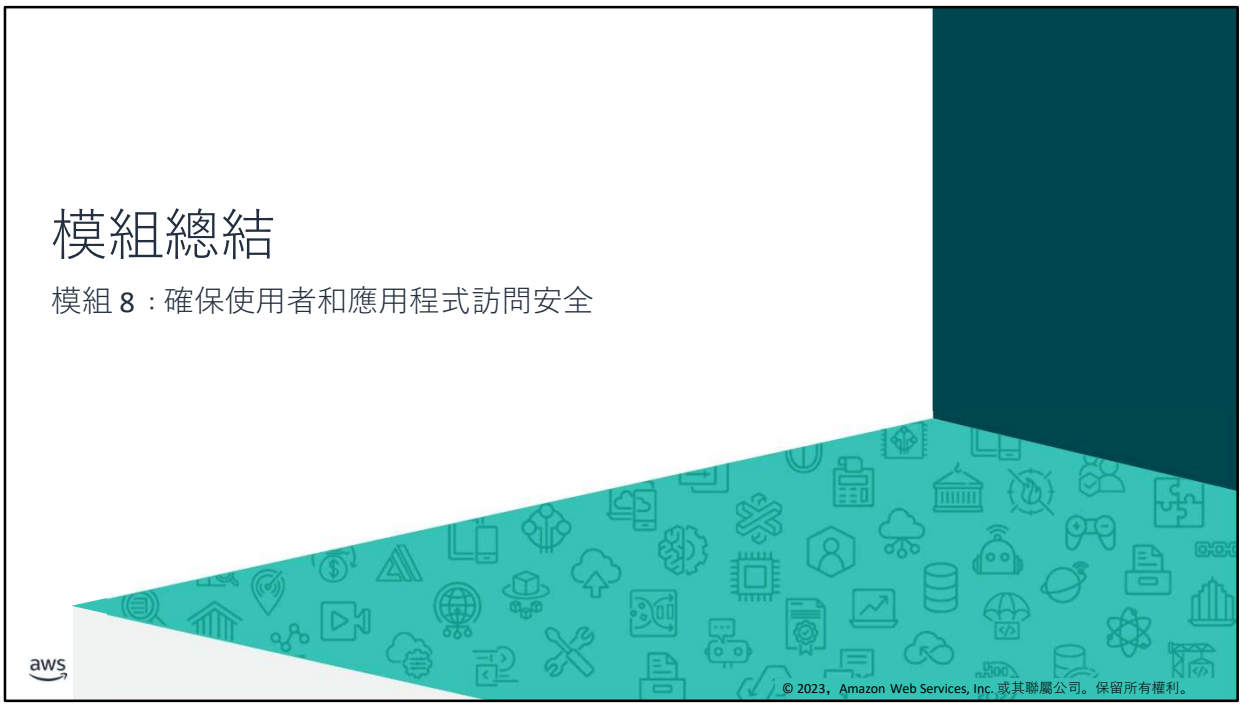
大約 80 分鐘

開始模組 8 – 挑戰實驗：使用 IAM 控制 AWS 帳戶訪問

現在可以開始挑戰實驗了。

挑戰實驗總結：要點

完成這個挑戰實驗之後，您的講師現在可能會帶您討論此挑戰實驗的要點。



現在來回顧下本模組，並對知識考核以及對實踐認證考試問題的討論進行總結。

模組總結

總的來說，在本模組中，您學習了如何：

- 說明 AWS Identity and Access Management (IAM) 用戶、組和角色的用途
- 描述如何允許在架構中使用用戶聯合以提高安全性
- 瞭解 AWS Organizations 服務控制策略 (SCP) 如何提高架構內的安全性
- 描述如何管理多個 AWS 帳戶
- 配置 IAM 用戶

總的來說，在本模組中，您學習了如何：
- 說明 AWS Identity and Access Management (IAM) 用戶、組和角色的用途
- 描述如何允許在架構中使用用戶聯合以提高安全性
- 瞭解 AWS Organizations 服務控制策略 (SCP) 如何提高架構內的安全性
- 描述如何管理多個 AWS 帳戶
- 配置 IAM 用戶

完成知識考核

現在該完成本模組的知識考核了。

考試樣題

公司將訪問金鑰（訪問金鑰 ID 和秘密訪問金鑰）存儲在自訂 AMI 上的文字檔中。該公司使用訪問金鑰訪問從 AMI 創建的實例中的 DynamoDB 表。安全團隊要求採用更安全的解決方案。

哪種解決方案能滿足安全團隊的要求？

選項	答案
A	將訪問金鑰放入 S3 存儲桶中，然後在啟動時從實例中檢索訪問金鑰。
B	將訪問金鑰通過實例使用者資料傳遞到實例。
C	從在私有子網中啟動的金鑰伺服器獲取訪問金鑰。
D	創建具有訪問表許可權的 IAM 角色，然後使用新角色啟用所有實例。

思考答案選項，並根據關鍵字排除錯誤選項。

考試樣題答案

公司將訪問金鑰（訪問金鑰 ID 和秘密訪問金鑰）存儲在自訂 AMI 上的文字檔中。該公司使用訪問金鑰訪問從 AMI 創建的實例中的 DynamoDB 表。安全團隊要求採用更安全的解決方案。

哪種解決方案能滿足安全團隊的要求？

正確答案是 D。

該問題中的關鍵字是“存儲訪問金鑰”、“來自實例的 DynamoDB 表”、“自訂 AMI”和“最安全的解決方案”。

以下是要識別的關鍵字：**“存儲訪問金鑰”、“實例中的 DynamoDB 表”、“自訂 AMI”和“最安全的解決方案”。**

正確答案是 D。EC2 實例的 IAM 角色允許運行在實例上的應用程式訪問 AWS 資源，而無需創建和存儲任何訪問金鑰。任何需要創建訪問金鑰的解決方案都會引入管理該金鑰的複雜性。

其他資源

- AWS Well-Architected Framework – 安全性支柱
- IAM 常見問題
- 創建 IAM 策略視頻
- 不同層的身份視頻
- 身份提供商和聯合身份

如果您想進一步瞭解本模組中涵蓋的主題，以下額外資源可能會對您有所幫助：

- AWS Well-Architected Framework – 安全性支柱
- IAM 常見問題
- 創建 IAM 策略視頻
- 不同層的身份視頻
- 身份提供商和聯合身份

感謝您完成本模組的學習。