

AWS 白皮書

標記 AWS 資源的最佳實務



標記 AWS 資源的最佳實務: AWS 白皮書

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

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

Table of Contents

摘要和介紹	i
您是 Well-Architected 嗎？	1
簡介	1
什麼是標籤？	3
建置您的標記策略	7
定義需求和使用案例	8
定義和發佈標記結構描述	9
AWS Organizations – 標籤政策	12
ExampleInc-CostAllocation.json	12
ExampleInc-DisasterRecovery.json	13
實作和強制執行標記	14
手動管理的資源	14
基礎設施即程式碼 (IaC) 受管資源	15
CI/CD 管道受管資源	16
強制執行	17
測量標記有效性和推動改進	20
標記使用案例	22
成本分配和財務管理的標籤	22
成本分配標籤	22
建置成本分配策略	23
操作和支援的標籤	26
自動化基礎設施活動	27
工作負載生命週期	27
事件管理	29
修補	30
操作可觀測性	31
資料安全、風險管理和存取控制的標籤	32
資料安全和風險管理	32
身分管理和存取控制的標籤	33
結論	35
貢獻者	36
深入閱讀	37
文件修訂	39
注意	41

AWS 詞彙表 42

..... xliii

標記 AWS 資源的最佳實務

發佈日期：2023 年 3 月 30 日 ([文件修訂](#))

Amazon Web Services (AWS) 可讓您以標籤形式將中繼資料指派給許多 AWS 資源。每個標籤都是簡單的標籤，由索引鍵和選用值組成，用於儲存資源或保留在該資源上資料的相關資訊。本白皮書著重於標記使用案例、策略、技術和工具，這些工具可協助您依據目的、團隊、環境或其他與您業務相關的條件來分類資源。實作一致的標記策略可以更輕鬆地篩選和搜尋資源、監控成本和用量，以及管理您的 AWS 環境。

本白皮書以[使用多個帳戶組織 AWS 您的環境](#)白皮書中提供的做法和指導為基礎。建議您先閱讀該白皮書，然後再閱讀。AWS 建議您以整體方式建立雲端基礎。如需詳細資訊，請參閱[建立您的 Cloud Foundation。AWS](#)

您是 Well-Architected 嗎？

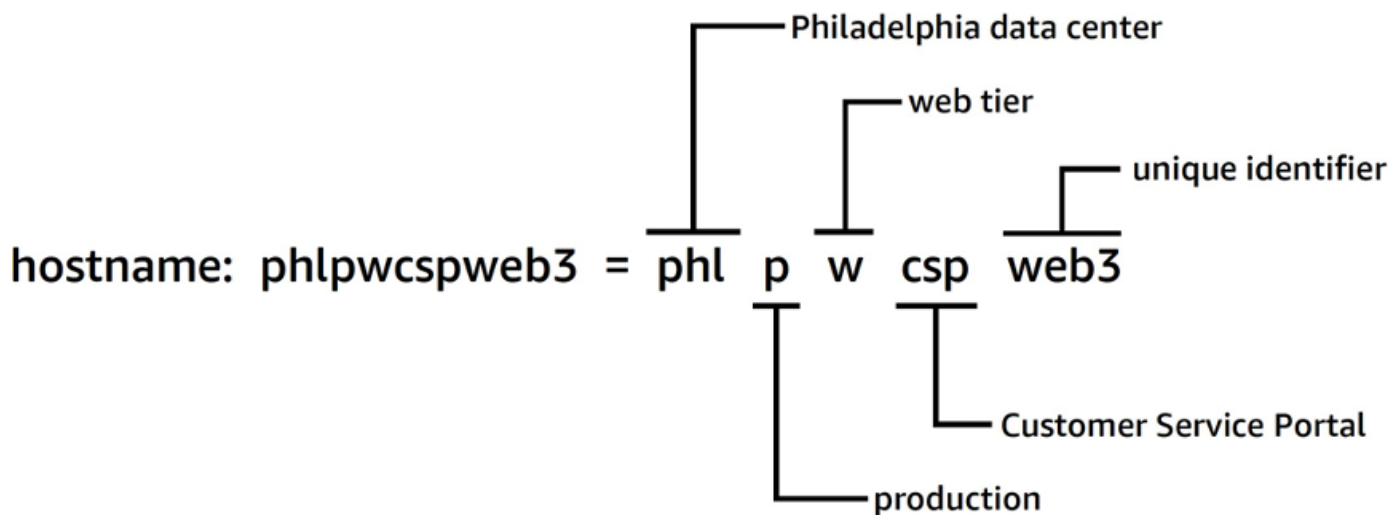
[AWS Well-Architected Framework](#) 可協助您了解在雲端建置系統時所做決策的優缺點。架構的六大支柱可讓您學習架構最佳實務，以設計和操作可靠、安全、有效率、經濟實惠且永續的系統。使用 [AWS Well-Architected Tool](#) 可免費在 中取得 [AWS Management Console](#)，您可以透過回答每個支柱的一組問題，根據這些最佳實務來檢閱工作負載。

如需雲端架構的更多專家指引和最佳實務，請參閱[AWS 架構中心](#)，參考架構部署、圖表和白皮書。

簡介

AWS 透過建立資源，例如 [Amazon EC2 執行個體](#)、[Amazon EBS 磁碟區](#)、[安全群組](#) 和 [AWS Lambda 函數](#)，在 中輕鬆部署工作負載。您也可以擴展和擴展 AWS 託管應用程式的資源機群、存放資料，以及隨著時間擴展 AWS 基礎設施。隨著您的 AWS 用量成長到跨越多個應用程式的許多資源類型，您將需要一個機制來追蹤哪些資源被指派給哪個應用程式。使用此機制來支援您的營運活動，例如成本監控、事件管理、修補、備份和存取控制。

在內部部署環境中，通常在知識管理系統、文件管理系統和內部 Wiki 頁面上擷取此知識。使用組態管理資料庫 (CMDB)，您可以使用標準變更控制程序來存放和管理相關的詳細中繼資料。這種方法提供控管，但需要額外的努力來開發和維護。您可以採取結構化方法來命名資源，但資源名稱只能保留有限數量的資訊。



資源命名的結構化方法

例如，EC2 執行個體具有名為 `Name` 的預先定義標籤，可提供類似的功能，並可讓您在工作負載移至其中時命名工作負載 AWS。

在 2010 年，AWS 推出[資源標籤](#)，以提供彈性且可擴展的機制，以將中繼資料連接至您的資源。本白皮書會引導您完成在整個 AWS 環境中開發和實作強大標記策略的程序。本指南將協助您確保標記一致性和支援決策和營運活動的涵蓋範圍

什麼是標籤？

標籤是套用至資源的[鍵/值對](#)，用於保留該資源的中繼資料。每個標籤都是由索引鍵和選用值組成的標籤。目前並非所有服務和資源類型都支援標籤（請參閱[支援資源群組標記 API 的服務](#)）。其他服務可能透過自己的 APIs 支援標籤。請注意，標籤未加密，不應用於存放敏感資料，例如個人身分識別資訊 (PII)。

使用者使用 AWS CLI、API 或 建立並套用至 AWS 資源的標籤 AWS Management Console 稱為使用者定義的標籤。數個 AWS 服務，例如 AWS CloudFormation、Elastic Beanstalk 和 Auto Scaling，會自動將標籤指派給他們建立和管理的資源。這些金鑰稱為 AWS 產生的標籤，通常字首為 `aws`。此字首不能用於使用者定義的標籤索引鍵。

可新增至 AWS 資源的使用者定義標籤數量有使用需求和限制。如需詳細資訊，請參閱《AWS 一般參考指南》中的[標籤命名限制和要求](#)。AWS 產生的標籤不會計入這些使用者定義的標籤限制。

表 1 – 使用者定義標籤索引鍵和值的範例

執行個體 ID	標籤索引鍵	標籤值
i-01234567abcdef89a	CostCenter	98765
	Stack	Test
i-12345678abcdef90b	CostCenter	98765
	Stack	Production

表 2 – AWS 產生的標籤範例

AWS 產生的標籤金鑰	理由
<code>aws:ec2spot:fleet-request-id</code>	識別啟動執行個體的 Amazon EC2 Spot 執行個體請求
<code>aws:cloudformation:stack-name</code>	識別建立資源的 AWS CloudFormation 堆疊
<code>lambda-console:blueprint</code>	識別用作 AWS Lambda 函數範本的藍圖

AWS 產生的標籤金鑰	理由
elasticbeanstalk:environment-name	識別建立資源的應用程式
aws:servicecatalog:provisionedProductArn	佈建產品 Amazon Resource Name (ARN)
aws:servicecatalog:productArn	從中啟動佈建產品的產品 ARN

AWS 產生的標籤會形成命名空間。例如，在 AWS CloudFormation 範本中，您可以定義一組要一起部署在中的資源stack，其中 stack-name 是您指派來識別資源的描述性名稱。如果您檢查等金鑰aws:cloudformation:stack-name，您可以看到用於範圍參數的命名空間使用三個元素：aws 組織、cloudformation 服務，以及 stack-name 參數。

使用者定義的標籤也可以使用命名空間，建議使用組織識別符做為字首。這可協助您快速識別標籤是來自受管結構描述的內容，還是您在環境中使用的服務或工具所定義的內容。

在在白皮書[上建立您的雲端基礎 AWS](#)中，我們建議實作一組標籤。不同的企業很可能具有不同的允許模式，以及特定標籤的不同清單。查看表 3 中的範例：

表 3 – 相同的標籤索引鍵、不同的值驗證規則

組織	標籤索引鍵	標籤值驗證	標籤值範例
公司 A	CostCenter	5432, 5422, 5499	5432
公司 B	CostCenter	ABC*	ABC123

如果這兩個結構描述位於不同的組織中，則標籤衝突沒有問題。不過，如果這兩個環境合併，命名空間可能會衝突，且驗證會變得更加複雜。此案例似乎不太可能，但會取得或合併企業，而且還有其他案例，例如與受管服務供應商合作的用戶端、遊戲發佈者或風險投資企業，其中來自不同組織的帳戶是共用 AWS 組織的一部分。透過使用商業名稱做為定義唯一命名空間的字首，可以避免這些挑戰，如表 4 所示：

表 4 – 在標籤索引鍵中使用命名空間

組織	標籤索引鍵	標籤值驗證	標籤值範例
公司 A	company-a :CostCenter	5432, 5422, 5499	5432
公司 B	company-b :CostCenter	ABC*	ABC123

在定期取得和剝離業務的大型複雜組織中，這種情況會更頻繁發生。隨著新收購的流程和實務在更廣泛的群組中協調一致，情況也會得到解決。具有不同的命名空間很有幫助，因為舊標籤的使用可以在上報告，並且聯絡相關團隊採用新的結構描述。命名空間也可以用來指示範圍，或代表與組織擁有者一致的使用案例或責任區域。

表 5 – 標籤索引鍵內範圍或使用案例範圍的範例

使用案例	標籤索引鍵	理由	允許值
資料分類	example-inc:info-sec:data-classification	資訊安全定義的一組資料分類	sensitive , company-confidential , customer-identifiable
作業	example-inc:dev-ops:environment	實作測試和開發環境的排程	development , staging, quality-assurance , production
災難復原	example-inc:disaster-recovery:rpo	定義資源的復原點目標 (RPO)	6h, 24h
成本分配	example-inc:cost-allocation:business-unit	財務團隊需要每個團隊用量和支出的成本報告	corporate , recruitment , support, engineering

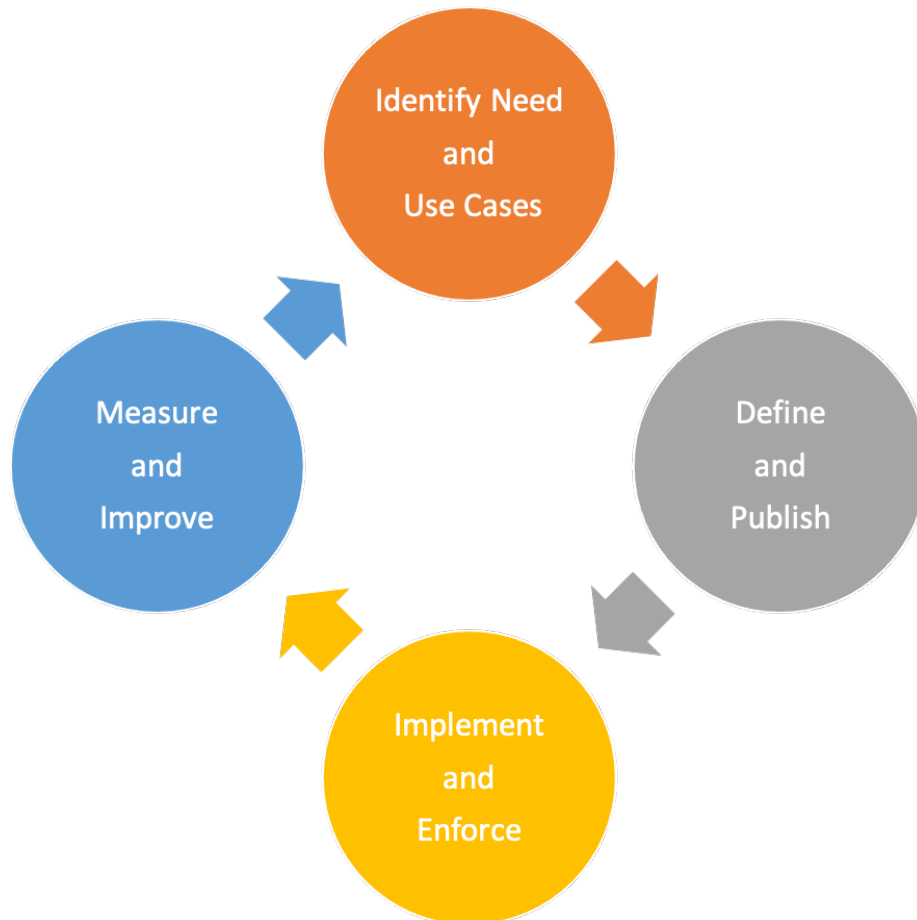
標籤簡單且靈活。索引鍵和標籤的值都是可變長度字串，並且可以支援寬字元集。如需長度和字元集的詳細資訊，請參閱《AWS 一般參考》中的[標記 AWS 資源](#)。標籤區分大小寫，這表示 `costCenter` 和 `costcenter` 是不同的標籤索引鍵。在不同的國家/地區，單字的拼法可能不同，這可能會影響您的金鑰。例如，在美國，可能將金鑰定義為 `costcenter`，但在英國 `costcentre` 可能是首選的。這些是與資源標記觀點不同的索引鍵。在標記策略中定義拼寫、大小寫和標點符號。使用這些定義作為建立或管理資源的任何人的參考。下一節會更詳細地討論本主題：[建置您的標記策略](#)。

 Note

雖然標籤索引鍵區分大小寫，但 IAM 對 IAM 資源有額外的驗證，以防止套用只在大小寫中不同的標籤索引鍵。我們建議不要使用只有不同大小寫的金鑰。如需詳細資訊，請參閱 [IAM 資源的標籤](#)。

建置您的標記策略

如同許多操作實務，實作標記策略是反覆運算和改善的過程。從您立即的優先順序開始，並根據需要擴展標記結構描述。



標記策略反覆運算和改進週期

在整個過程中，擁有權是責任和進展的關鍵。由於標籤可用於各種用途，因此整體標記策略可以分割為組織內的責任區域。標記可讓以程式設計方式處理取決於資源特性的活動。可以從標記中受益的利益相關者範圍取決於組織的大小和操作實務。較大的組織可以從明確定義參與建立和實作標記策略之團隊的責任中受益。有些利益相關者可以負責識別標記的需求（定義使用案例）；有些利益相關者可以負責維護、實作和改善標記策略。

透過指派所有權，您便處於實作策略個別層面的良好位置。在適當的情況下，此所有權可以正式化為政策，並記錄在責任矩陣中（例如 RACI：Responsible、accounted、Conchoned 和 Informed），或在共同責任模型中。在較小的組織中，團隊可能會在標記策略中扮演多個角色，從需求定義到實作和強制執行。

定義需求和使用案例

與基本需要取用中繼資料的利益相關者互動，開始建置您的策略。這些團隊會定義資源需要加上標籤以支援其活動的中繼資料，例如報告、自動化和資料分類。它們概述了如何組織資源，以及它們需要映射到哪些政策。這些利益相關者在組織中可以擁有的角色和職能範例包括：

- 財務和業務單位需要透過將其映射到成本來了解投資價值，以優先處理效率不佳時需要採取的動作。了解產生的成本與價值，有助於識別業務或產品供應項目的不成功線。這會導致有關持續支援、採用替代方案（例如，使用 SaaS 服務或受管服務）或淘汰無獲利商業產品的明智決策。
- 治理和合規需要了解資料的分類（例如公有、敏感或機密）、特定工作負載是否在或超出特定標準或法規的稽核範圍，以及服務的重要性（無論服務或應用程式是否為業務關鍵），以套用適當的控制和監督，例如許可、政策和監控。
- 營運和開發需要了解工作負載生命週期、其支援產品的實作階段，以及發行階段（例如開發、測試、生產分割）的管理，及其相關的支援優先順序和利益相關者管理需求。備份、修補、可觀測性和棄用等職責也需要定義和理解。
- 資訊安全 (InfoSec) 和安全操作 (SecOps) 概述了必須套用哪些控制項，以及建議使用哪些控制項。InfoSec 通常會定義控制項的實作，SecOps 通常負責管理這些控制項。

根據您的使用案例、優先順序、組織大小和營運實務，您可能需要組織內各種團隊的代表，例如財務（包括採購）、資訊安全、雲端啟用和雲端營運。您也需要應用程式和程序擁有者的代表，才能執行修補、備份和還原、監控、任務排程和災難復原等功能。這些代表可協助推動標籤策略的定義、實作和衡量其有效性。他們應該從利益相關者及其使用案例[向後工作](#)，並進行跨職能研討會。在研討會中，他們有機會分享他們的觀點和需求，並協助推動整體策略。本白皮書稍後將說明參與者及其參與各種使用案例的範例。

利益相關者也會定義和驗證強制標籤的金鑰，並且可以建議選用標籤的範圍。例如，財務團隊可能需要將資源與內部成本中心、業務單位或兩者相關聯。因此，它們可能需要強制某些標籤索引鍵 BusinessUnit，例如 CostCenter 和。個別開發團隊可能會決定將其他標籤用於自動化目的，例如 EnvironmentName、OptIn 或 OptOut。

主要利益相關者需要同意標記策略方法，並記錄合規和控管相關問題的答案，例如：

- 需要解決哪些使用案例？
- 誰負責標記資源（實作）？
- 如何強制執行標籤，以及將使用哪些方法和自動化（主動或被動）？
- 如何衡量標記有效性和目標？

- 應該多久檢閱一次標記策略？
- 誰推動改進？如何完成此操作？

然後，Cloud Enablement、Cloud Business Office 和 Cloud Platform Engineering 等業務職能可以扮演促進者的角色，以建構標記策略、協助推動採用，並透過測量進度、移除障礙和減少重複的工作來確保應用程式的一致性。

定義和發佈標記結構描述

使用一致的方法來標記您的 AWS 資源，包括強制性和選用的標籤。全面的標記結構描述可協助您實現此一致性。下列範例可協助您開始使用：

- 同意強制性標籤索引鍵
- 定義可接受的值和標籤命名慣例（大寫或小寫、破折號或底線、階層等）
- 確認值不會構成個人身分識別資訊 (PII)
- 決定誰可以定義和建立新的標籤索引鍵
- 同意如何新增強制性標籤值，以及如何管理選用標籤

檢閱下列[標記類別](#)表，其可做為標記結構描述中可能包含內容的基準。您仍然需要判斷您將用於標籤金鑰的慣例，以及每個慣例允許哪些值。標記結構描述是您為環境定義此項目的文件。

表 6 – 最終標記結構描述的範例（第 1 部分）

使用案例	標籤索引鍵	理由	允許的值（列出 或值字首/尾）	用於成本 分配	資源類型	範圍	必要
成本分配	example-incident-cost-allocation : ApplicationId	追蹤每行業務所產生的成本與價值	DataLakeX , RetailSiteX	Y	全部	全部	強制性
	example-incident-cost-allocation : BusinessUnitId	依業務單位監控成本	Architecture , DevOps, Finance	Y	全部	全部	強制性
成本分配	example-incident-cost-allocation : CostCenter	依成本中心監控成本	123-*	Y	全部	全部	強制性
成本分配	example-incident-cost-allocation : Owner	哪些預算持有者負責此工作負載	Marketing , RetailSupport	Y	全部	全部	強制性
存取控制	example-incident-access-control : LayerId	識別SubComponent/層，以根據角色授予對資源的存取權	DB_Layer, Web_Layer , App_Layer	N	全部	全部	選用
自動化	example-incident-automa	實作測試和開發環境的排程，也	Prod, Dev, Test, Sandbox	N	EC2、RDS BS	全部	強制性

表 6 – 最終標記結構描述的範例（第 2 部分）

使用案例	標籤索引鍵	理由	允許的值（列出或值字首/尾）	用於成本分配	資源類型	範圍	必要
DevOps	example-incident:operations:Owner	哪個團隊/小組負責資源的建立和維護	Squad01	N	全部	全部	強制性
災難復原	example-incident:disaster-recovery:rpo	定義資源的復原點目標 (RPO)	6h, 24h	N	S3、EBS	生產	強制性
資料分類	example-incident:data:classification	分類資料以進行合規和管理	Public, Private, Confidential, Restricted	N	S3、EBS	全部	強制性
合規	example-incident:compliance:framework	識別工作負載受制於的合規架構	PCI-DSS, HIPAA	N	全部	生產	強制性

定義標記結構描述後，請在版本控制的儲存庫中管理結構描述，讓所有相關利益相關者都能存取，以便於參考和追蹤更新。這種方法可提高效率，並實現敏捷性。

AWS Organizations – 標籤政策

中的政策 AWS Organizations 可讓您將其他類型的控管套用至組織中的 AWS 帳戶。[標籤政策](#)是以 JSON 格式表達標記結構描述的方式，讓平台可以在 AWS 您的環境中報告並選擇性地強制執行結構描述。標籤政策會定義特定資源類型上標籤索引鍵可接受的值。此政策可以是值清單的形式，或是字首後面加上萬用字元 (*)。簡單字首方法不像分散的值清單那麼嚴格，但需要的維護較少。

下列範例示範如何定義標記政策，以驗證指定金鑰可接受的值。使用結構描述的人類友好表格式定義，您可以將此資訊轉錄為一或多個標籤政策。個別政策可用於支援委派所有權，或某些政策可能僅適用於特定案例。

ExampleInc-CostAllocation.json

以下是報告成本分配標籤的標籤政策範例：

```
{
  "tags": {
    "example-inc:cost-allocation:ApplicationId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:ApplicationId"
      },
      "tag_value": {
        "@@assign": [
          "DataLakeX",
          "RetailSiteX"
        ]
      }
    },
    "example-inc:cost-allocation:BusinessUnitId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:BusinessUnitId"
      },
      "tag_value": {
        "@@assign": [
          "Architecture",
          "DevOps",
          "FinanceDataLakeX"
        ]
      }
    }
  }
}
```

```

    }
  },
  "example-inc:cost-allocation:CostCenter": {
    "tag_key": {
      "@@assign": "example-inc:cost-allocation:CostCenter"
    },
    "tag_value": {
      "@@assign": [
        "123-*"
      ]
    }
  }
}
}
}

```

ExampleInc-DisasterRecovery.json

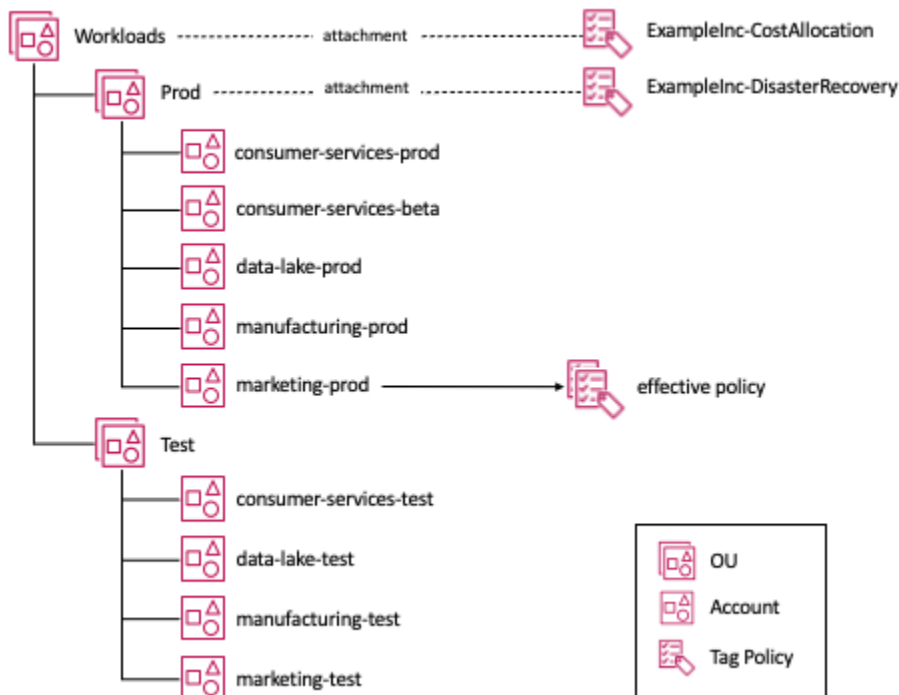
以下是報告災難復原標籤的標籤政策範例：

```

{
  "tags": {
    "example-inc:disaster-recovery:rpo": {
      "tag_key": {
        "@@assign": "example-inc:disaster-recovery:rpo"
      },
      "tag_value": {
        "@@assign": [
          "6h",
          "24h"
        ]
      }
    }
  }
}

```

在此範例中，ExampleInc-CostAllocation標籤政策會連接至 Workloads OU，因此適用於 Prod和Test子 OUs 中的所有帳戶。同樣地，ExampleInc-DisasterRecovery標籤政策會連接至 Prod OU，因此僅適用於低於此 OU 的帳戶。[使用多個帳戶組織您的環境](#) 白皮書會更詳細地探索建議的 OU 結構。



將標籤政策連接至 OU 結構

查看圖表中的marketing-prod帳戶，這兩個標籤政策都適用於此帳戶，因此我們有有效政策的概念，這是適用於帳戶之指定類型政策的卷積。如果您主要手動管理您的資源，則可以造訪主控制台中的[資源群組和標籤編輯器：標籤政策來檢閱有效政策](#)。如果您使用基礎設施做為程式碼 (IaC) 或指令碼來管理您的資源，您可以使用 [AWS::Organizations::DescribeEffectivePolicy](#) API 呼叫。

實作和強制執行標記

在本節中，我們將介紹下列資源管理策略可用的工具：手動、基礎設施即程式碼 (IaC)，以及持續整合/持續交付 (CI/CD)。這些方法的關鍵維度是部署頻率越來越頻繁。

手動管理的資源

這些通常是屬於[採用基礎或遷移階段](#)的工作負載。通常，這些是使用傳統寫入程序建置的簡單主要靜態工作負載，或是使用 AWS Application Migration Service 內部部署環境等工具遷移的工作負載。遷移工具，例如 Migration Hub 和 Application Migration Service，可以套用標籤作為遷移程序的一部分。不過，如果在原始遷移期間未套用標籤，或標記結構描述自那時起已變更，[標籤編輯器](#) (的功能 AWS Management Console) 可讓您使用各種搜尋條件搜尋資源，並大量新增、修改或刪除標籤。搜尋條件

可以包含具有或不存在特定標籤或值的資源。AWS 資源標記 API 可讓您以程式設計方式執行這些函數。

由於這些工作負載已現代化，因此會引入 Auto Scaling 群組等資源類型。這些資源類型可提高彈性並改善彈性。自動擴展群組會代表您管理 Amazon EC2 執行個體，不過，您可能仍希望 EC2 執行個體與手動建立的資源一致地加上標籤。[Amazon EC2 啟動範本](#) 提供指定 Auto Scaling 應套用至其建立之執行個體的標籤的方法。

手動管理工作負載的資源時，自動化資源標記會很有幫助。有各種可用的解決方案。其中一種方法是使用 AWS Config 規則，它可以檢查 `required_tags`，然後啟動 Lambda 函數來套用它們。本白皮書稍後 AWS Config 規則 會更詳細地說明。

基礎設施即程式碼 (IaC) 受管資源

AWS CloudFormation 提供常見的語言，用於佈建 AWS 環境中的所有基礎設施資源。CloudFormation 範本是以自動化方式建立 AWS 資源的 JSON 或 YAML 檔案。使用 CloudFormation 範本建立 AWS 資源時，您可以使用 CloudFormation Resource Tags 屬性，在建立時將標籤套用至支援的資源類型。使用 IaC 管理標籤和資源有助於確保一致性。

建立資源時 AWS CloudFormation，服務會自動將一組 AWS 定義的標籤套用至 AWS CloudFormation 範本建立的資源。這些時間為：

```
aws:cloudformation:stack-name
aws:cloudformation:stack-id
aws:cloudformation:logical-id
```

您可以根據 AWS CloudFormation 堆疊輕鬆定義資源群組。這些 AWS 定義的標籤由堆疊建立的資源繼承。不過，對於 Auto Scaling 群組中的 Amazon EC2 執行個體，需要在 AWS CloudFormation 範本中 Auto Scaling 群組的定義中設定 [AWS::AutoScaling::AutoScalingGroup TagProperty](#)。或者，如果您使用 [EC2 啟動範本](#) 搭配 Auto Scaling 群組，您可以在其定義中定義標籤。建議搭配 Auto Scaling 群組或 AWS 容器服務使用 [EC2 啟動範本](#)。這些服務可協助確保 Amazon EC2 執行個體的一致標記，也支援[跨多個執行個體類型和購買選項的 Auto Scaling](#)，這可以改善彈性並最佳化運算成本。

[AWS CloudFormation 勾點](#) 為開發人員提供了一種方法，使其應用程式的關鍵層面與其組織的標準保持一致。勾點可設定為提供警告或防止部署。此功能最適合用於檢查範本中的金鑰組態元素，例如 Auto Scaling 群組是否設定為將客戶定義的標籤套用至其將啟動的所有 Amazon EC2 執行個體，或確保所有 Amazon S3 儲存貯體都使用必要的加密設定建立。在這兩種情況下，此合規的評估都會推送至部署程序中稍早的，並在部署之前使用 AWS CloudFormation 掛鉤。

AWS CloudFormation 提供偵測從範本佈建的資源（請參閱[支援偏離偵測的資源](#)）何時已修改，且資源不再符合其預期的範本組態的功能。這稱為偏離。如果您使用自動化將標籤套用至透過 IaC 管理的資源，則您正在修改它們，並引入偏離。使用 IaC 時，目前建議在 IaC 範本中管理任何標記需求、實作 AWS CloudFormation 勾點，以及發佈可供自動化使用的 AWS CloudFormation Guard 規則集。

CI/CD 管道受管資源

隨著工作負載的成熟度增加，可能會採用持續整合和持續部署 (CI/CD) 等技術。這些技術透過提高測試自動化，讓部署小型變更更頻繁，協助降低部署風險。偵測部署引入意外行為的可觀測性策略，可以在使用者影響最小的情況下自動復原部署。隨著您進入每天部署數十次的階段，追溯套用標籤不再實際。所有項目都必須以程式碼或組態、版本控制，並盡可能在部署到生產環境之前進行測試和評估。在合併[開發和操作 \(DevOps\) 模型](#)中，許多實務將操作考量事項視為程式碼，並在部署生命週期的早期進行驗證。

理想情況下，您希望儘早推送這些檢查（如 AWS CloudFormation 勾點所示），以便在 AWS CloudFormation 範本離開開發人員的機器之前，確信範本符合您的政策。

[AWS CloudFormation Guard 2.0](#) 提供撰寫預防性合規規則的方法，讓您使用 CloudFormation 定義任何內容。範本會根據開發環境中的規則進行驗證。顯然，此功能有一系列的應用程式，但在本白皮書中，我們將查看一些範例，以確保始終使用 [AWS::AutoScaling::AutoScalingGroup TagProperty](#)。

以下是 CloudFormation Guard 規則的範例：

```
let all_asgs = Resources.*[ Type == 'AWS::AutoScaling::AutoScalingGroup' ]

rule tags_asg_automation_EnvironmentId when %all_asgs != empty {
  let required_tags = %all_asgs.Properties.Tags.*[
    Key == 'example-inc:automation:EnvironmentId' ]
  %required_tags[*] {
    PropagateAtLaunch == 'true'
    Value IN ['Prod', 'Dev', 'Test', 'Sandbox']
    <<Tag must have a permitted value
      Tag must have PropagateAtLaunch set to 'true'>>
  }
}

rule tags_asg_costAllocation_CostCenter when %all_asgs != empty {
  let required_tags = %all_asgs.Properties.Tags.*[
    Key == 'example-inc:cost-allocation:CostCenter' ]
  %required_tags[*] {
    PropagateAtLaunch == 'true'
```

```

    Value == /^123-/
    <<Tag must have a permitted value
        Tag must have PropagateAtLaunch set to 'true'>>
  }
}

```

在程式碼範例中，我們為類型為 `AutoScalingGroup` 的所有資源篩選範本 `AutoScalingGroup`，然後有兩個規則：

- **tags_asg_automation_EnvironmentId** - 檢查具有此金鑰的標籤是否存在、在允許的值清單中具有值，且 `PropagateAtLaunch` 設定為 `true`
- **tags_asg_costAllocation_CostCenter** - 檢查標籤是否存在於此索引鍵，其值開頭為定義的字首值，且 `PropagateAtLaunch` 設定為 `true`

強制執行

如前所述，資源群組和標籤編輯器提供識別資源無法滿足套用至組織 OUs 之標籤政策中定義的標記要求的方法。從組織成員帳戶內存取資源群組和標籤編輯器主控台工具，會顯示套用至該帳戶的政策，以及帳戶內不符合標籤政策需求的資源。如果從管理帳戶存取（如果標籤政策已在下的服務中啟用存取 AWS Organizations），則可以檢視[組織中所有連結帳戶的標籤政策合規](#)。

在標籤政策本身中，您可以針對特定資源類型啟用強制執行。在下列政策範例中，我們新增了強制執行，以便類型 `ec2:instance` 和 `ec2:volume` 的所有資源都必須符合政策。有一些已知的限制，例如資源上必須有標籤，才能由標籤政策評估。如需清單，[請參閱支援使用標籤政策強制執行的資源](#)。

ExampleInc-Cost-Allocation.json

以下是報告和/或強制執行成本分配標籤的標籤政策範例：

```

{
  "tags": {
    "example-inc:cost-allocation:ApplicationId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:ApplicationId"
      },
      "tag_value": {
        "@@assign": [
          "DataLakeX",
          "RetailSiteX"
        ]
      },
    },
    "enforced_for": {

```

```
    "@@assign": [
      "ec2:instance",
      "ec2:volume"
    ]
  },
  "example-inc:cost-allocation:BusinessUnitId": {
    "tag_key": {
      "@@assign": "example-inc:cost-allocation:BusinessUnitId"
    },
    "tag_value": {
      "@@assign": [
        "Architecture",
        "DevOps",
        "FinanceDataLakeX"
      ]
    },
    "enforced_for": {
      "@@assign": [
        "ec2:instance",
        "ec2:volume"
      ]
    }
  },
  "example-inc:cost-allocation:CostCenter": {
    "tag_key": {
      "@@assign": "example-inc:cost-allocation:CostCenter"
    },
    "tag_value": {
      "@@assign": [
        "123-*"
      ]
    },
    "enforced_for": {
      "@@assign": [
        "ec2:instance",
        "ec2:volume"
      ]
    }
  }
}
```

AWS Config (**required_tag**)

AWS Config 是一項服務，可讓您評估、稽核和評估 AWS 資源的組態（請參閱 [支援的資源類型 AWS Config](#)）。在標記的情況下，我們可以使用 `required_tags` 規則（請參閱 [required_tags 支援的資源類型](#)）來識別缺少具有特定金鑰之標籤的資源。從先前的範例中，我們可能會在所有 Amazon EC2 執行個體上測試金鑰是否存在。在金鑰不存在的情況下，執行個體會註冊為不合規。此 AWS CloudFormation 範本描述 AWS Config 規則，以測試資料表、Amazon S3 儲存貯體、Amazon EC2 執行個體和 Amazon EBS 磁碟區中描述的強制性金鑰是否存在。

```
Resources:
  MandatoryTags:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: ExampleIncMandatoryTags
      Description: These tags should be in place
      InputParameters:
        tag1Key: example-inc:cost-allocation:ApplicationId
        tag2Key: example-inc:cost-allocation:BusinessUnitId
        tag3Key: example-inc:cost-allocation:CostCenter
        tag4Key: example-inc:automation:EnvironmentId
      Scope:
        ComplianceResourceTypes:
          - "AWS::S3::Bucket"
          - "AWS::EC2::Instance"
          - "AWS::EC2::Volume"
      Source:
        Owner: AWS
      SourceIdentifier: REQUIRED_TAGS
```

對於手動管理資源的環境，可以增強 AWS Config 規則，以使用透過 AWS Lambda 函數的自動修復，自動將缺少的標籤索引鍵新增至資源。雖然這適用於靜態工作負載，但在您開始透過 IaC 和部署管道管理資源時，其效率會逐漸降低。

AWS Organizations – 服務控制政策 (SCPs) 是一種組織政策，可用來管理組織中的許可。SCPs 可讓您集中控制組織或組織單位 (OU) 中所有帳戶的最大可用許可。SCPs 只會影響由屬於組織一部分的帳戶管理的使用者和角色。雖然它們不會直接影響資源，但會限制使用者和角色的許可，其中包含標記動作的許可。對於標記，除了標籤政策可以提供的標籤之外，SCPs 還可以为標籤強制執行提供額外的精細程度。

在下列範例中，政策會拒絕不存在 `example-inc:cost-allocation:CostCenter` 標籤的 `ec2:RunInstances` 請求。

以下是拒絕 SCP：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRunInstanceWithNoCostCenterTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/example-inc:cost-allocation:CostCenter": "true"
        }
      }
    }
  ]
}
```

無法依設計擷取套用至連結帳戶的有效服務控制政策。當您使用 SCPs 強制執行標記時，開發人員需要提供文件，以確保其資源符合已套用至其帳戶的政策。在其帳戶中提供 CloudTrail 事件的唯讀存取權，可以支援開發人員在其資源不合規時進行偵錯任務。

測量標記有效性和推動改進

實作標記策略之後，請務必針對目標使用案例衡量其有效性。有效性的衡量方式會因使用案例而有所不同。例如：

- 成本歸因 - 您可以根據使用 [AWS Cost Explorer](#) 或 [AWS 成本和用量報告](#) 等工具的支出來衡量資源的標記涵蓋範圍。例如，您可以追蹤產生費用的標記或未標記資源百分比，特別是監控特定標籤索引鍵。
- 自動化 - 如果已達成所需的結果，您可能想要稽核。例如，是否在營業時間外暫停非生產 Amazon EC2 執行個體，稽核執行個體開始和停止時間。

管理帳戶中的 [資源群組和標籤編輯器](#) 提供額外的功能，以分析組織中所有連結帳戶的標籤政策合規性。

根據標記有效性的測量結果，識別是否需要在任何步驟中進行任何改善或修改，例如使用案例定義、標記結構描述實作或強制執行。進行必要的變更並重複週期，直到達到所需的有效性為止。在成本歸因範例中，您可以查看百分比改善。

由於開發人員和運算子需要執行資源的實際標記，因此讓他們取得所有權至關重要。這並非團隊在 AWS 採用過程中通常承擔的唯一額外責任。提高開發和操作其應用程式的安全性和成本也很重要。組織通常會使用目標和目標作為鼓勵採用新實務的方式，因此也可以在此處套用。

標記使用案例

主題

- [成本分配和財務管理的標籤](#)
- [操作和支援的標籤](#)
- [資料安全、風險管理和存取控制的標籤](#)

成本分配和財務管理的標籤

組織最常處理的標記使用案例之一是成本和用量的可見性和管理。這通常有一些原因：

- 通常，這是眾所周知的案例，而需求也眾所周知。例如，財務團隊想要查看跨多個服務、功能、帳戶或團隊之工作負載和基礎設施的總成本。實現此成本可見性的一種方法是透過一致的資源標記。
- 標籤及其值已明確定義。通常，成本分配機制已存在於組織的財務系統中，例如，透過成本中心、業務單位、團隊或組織職能進行追蹤。
- 快速、可證明的投資回報。當資源標籤一致時，可以追蹤一段時間內的成本最佳化趨勢，例如，針對已適當化、自動擴展或排程的資源。

了解您在 中產生成本的方式，AWS 可讓您做出明智的財務決策。與實現的業務成果相比，了解您在資源、工作負載、團隊或組織層級產生成本的位置，可增強您對在適用層級交付價值的了解。

工程團隊可能沒有資源的財務管理經驗。連接具有 AWS 財務管理專業技能的人員，該人員可以培訓工程和開發團隊了解 AWS 財務管理的基礎知識，並在財務和工程之間建立關係，以培養 FinOps 的文化，將有助於實現業務的可衡量成果，並鼓勵團隊以成本為重心建置。Well-Architected Framework [的成本最佳化支柱](#)會深入探討建立良好的財務實務，但我們將探討本白皮書中的幾個基本原則。

成本分配標籤

成本分配是指在定義程序後，將產生的成本指派或分配給這些成本的使用者或受益者。在本白皮書中，我們將成本分配分為兩種類型：顯示和退款。

顯示工具和機制有助於提高成本意識。退款有助於成本復原並推動成本感知。顯示是關於特定實體產生的費用的呈現、計算和報告，例如業務單位、應用程式、使用者或成本中心。例如：「上個月基礎設施工程團隊負責 \$X 的 AWS 支出」。退款是有關透過組織的內部會計程序向這些實體實際收取產生的成本，例如金融系統或日誌憑證。例如：「從基礎設施工程團隊的 AWS 預算中扣除 \$X。」在這兩種情況下，適當地標記資源有助於將成本分配給實體，唯一的區別是是否預期有人付款。

您組織的財務控管可能需要透明地計算應用程式、業務單位、成本中心和團隊層級所產生的成本。執行 [Cost Allocation Tags](#) 支援的成本歸因，可提供準確歸因實體從適當標記的資源所產生的成本所需的資料。

- 問責 — 確保將成本分配給負責資源使用的人員。單一服務或群組可能需要負責支出審查和報告。
- 財務透明度 — 透過為領導層建立有效的儀表板和有意義的成本分析，顯示對 IT 的現金配置的清晰觀點。
- 知情 IT 投資 — 根據專案、應用程式或業務線追蹤投資報酬率，並授權團隊做出更好的業務決策，例如，將更多資金分配給產生營收的應用程式。

總而言之，成本分配標籤有助於告訴您：

- 誰擁有支出並負責將其最佳化？
- 產生支出的工作負載、應用程式或產品為何？哪個環境或階段？
- 哪些支出區域成長速度最快？
- 根據過去的趨勢，可以從 AWS 預算中扣除多少支出？
- 在特定工作負載、應用程式或產品中，成本最佳化工作有何影響？

啟用成本分配的資源標籤有助於定義組織內的衡量實務，可用於提供 AWS 用量的可見性，以提高支出責任的透明度。它還專注於在成本和用量可見性方面建立適當的精細程度，並透過成本分配報告和 KPI 追蹤影響雲端消耗行為。

建置成本分配策略

定義和實作成本分配模型

為部署的資源建立帳戶和成本結構 AWS。建立 AWS 支出成本、這些成本的產生方式，以及產生這些成本的人員或來源之間的關係。常見的成本結構是以組織內 AWS Organizations AWS 帳戶的環境和實體為基礎，例如業務或工作負載。成本結構可以基於多個屬性，以允許以不同方式或以不同精細程度來檢查成本，例如將個別工作負載的成本累積到其服務的業務範圍。

選擇符合所需結果的成本結構時，請評估成本分配機制的實作簡易性與所需準確性。這可能包括責任、工具可用性和文化變更方面的考量。AWS 客戶通常從中開始的三種熱門成本分配模型為：

- 以帳戶為基礎 — 此模型需要最少的工作量，並為顯示和退款提供高準確性，並且適用於具有定義帳戶結構的組織（並且與[使用多個帳戶組織您的 AWS 環境](#)白皮書的建議一致）。這可提供每個帳戶

清晰的成本可見性。如需成本可見性和分配，您可以使用 [AWS Cost Explorer](#)、[成本和用量報告](#)，以及 [AWS 預算](#) 進行成本監控和追蹤。這些工具依 提供篩選和分組選項 AWS 帳戶。從成本分配的角度來看，此模型不必依賴個別資源的準確標記。

- **業務單位或團隊型** — 成本可分配給企業內的團隊、業務單位或組織。此模型需要中等程度的工作量，為表演和費用提供高準確度，並且適用於具有已定義帳戶結構（通常使用 AWS Organizations）的組織，在各種團隊、應用程式和工作負載類型之間進行區隔。這可在團隊和應用程式之間提供清晰的成本可見性，而且由於額外的優勢可降低在單一內達到 [AWS 服務配額](#) 的風險 AWS 帳戶。例如，每個團隊可能有五個帳戶 (prod、staging、testdev、 、sandbox)，而且沒有任何兩個團隊和應用程式共用相同的帳戶。透過這類結構，[AWS Cost Categories](#) 會提供將帳戶或其他標籤（「中繼標記」）分組為類別的功能，這些功能可在上一個範例中提及的工具中追蹤。請務必注意，AWS Organizations 允許標記帳戶和組織單位 (OUs)，但這些標籤不適用於成本分配和帳單報告（即您無法 AWS Cost Explorer 依 OU 在中分組或篩選成本）。AWS Cost Categories 應該用於此目的。
- **標籤型** — 此模型比前兩個模型需要更多的精力，並且會根據需求和最終目標提供高準確性的顯示和收費。雖然我們強烈建議您採用 [使用多個帳戶組織 AWS 您的環境](#) 白皮書中概述的實務，但逼真的客戶通常會發現自己具有混合且複雜的帳戶結構，需要一些時間才能移出。在此案例中，實作嚴格且有效的標記策略是關鍵，接著在 Billing and Cost Management 主控台中 [啟用成本分配的相關標籤](#)（在中 AWS Organizations，只能從 Management Payer 帳戶針對成本分配啟用標籤）。為成本分配啟用標籤後，先前方法中提到的成本可見性和分配工具可用於顯示和退款。請注意，成本分配標籤不是追溯性的，只有在啟用成本分配之後，才會出現在帳單報告和成本追蹤工具中。

總而言之，如果您需要依業務單位追蹤成本，您可以使用 [AWS Cost Categories](#) 對 AWS 組織內的連結帳戶進行相應分組，並在帳單報告中檢視此分組。當您為生產環境和非生產環境建立個別帳戶時，您也可以篩選 等工具中與環境相關的成本 [AWS Cost Explorer](#)，或使用 [AWS Budgets](#) 追蹤這些成本。最後，如果您的使用案例需要更精細的成本追蹤，例如個別的工作負載或應用程式，您可以相應地標記這些帳戶中的資源、[啟用這些標籤索引鍵以進行管理帳戶的成本分配](#)，然後依帳單報告工具中的標籤索引鍵篩選該成本。

建立成本報告和監控程序

從識別對內部利益相關者重要的成本類型開始（例如，每日支出、帳戶成本、X 成本、攤銷成本）。透過這樣做，您可以比等待最終 AWS 發票更快地減少與意外或異常支出相關的預算風險。標籤提供啟用這些報告案例的屬性。從報告獲得的洞見可以通知您的動作，以減輕異常和意外支出對財務預算的影響。當成本意外激增時，請務必評估交付的值是否發生意外激增，以便您判斷是否需要採取和需要採取哪些動作。

開發標籤策略以支援成本分配時，請記住下列元素：

- **AWS Organizations** - 多個帳戶內的成本分配可以依帳戶、帳戶群組或為這些帳戶上的資源建立的標籤群組來執行。為位於中個別帳戶中的資源建立的標籤，AWS Organizations 只能用於管理帳戶的成本分配。
- **AWS 帳戶** - 一個內部的成本分配 AWS 帳戶 可以由服務或區域等其他維度執行。您可以進一步標記帳戶中的資源，並使用這類資源標籤的群組。
- **成本分配標籤** - 使用者建立的標籤和 AWS 產生的標籤都可以在必要時啟用成本分配。在帳單主控台 (管理帳戶的 AWS Organizations) 中啟用成本分配標籤，有助於處理顯示和退款。
- **Cost Categories** - AWS Cost Categories 允許在 AWS 組織內分組帳戶和分組標籤 (「中繼標記」)，這進一步提供了透過 AWS Cost Explorer AWS 預算和成本和用量報告等工具分析這些類別相關 AWS 成本的能力。

為企業內的業務單位、團隊或組織執行顯示和收費

使用成本結構和成本分配標籤支援的成本分配程序來歸納成本。標籤可用於向不直接負責支付成本，但負責產生這些成本的團隊提供表演。此方法可讓您了解其對支出的貢獻，以及這些成本的產生方式。對直接負責成本的團隊執行收費，以回收已消耗的資源費用，並讓他們了解這些成本及其產生方式。

測量和循環效率或價值 KPIs

同意一組單位成本或 KPI 指標，以衡量雲端財務管理投資的影響。本練習會建立跨技術和業務利益相關者的通用語言，並告知以效率為基礎的案例，而不是僅專注於絕對、彙總支出的案例。如需詳細資訊，請參閱此部落格，討論[單位指標如何協助在業務職能之間建立一致性](#)。

分配無法配置的支出

根據組織的會計實務，不同的費用類型可能需要不同的處理方式。識別無法標記的資源或成本類別。根據所使用的服務和計劃使用的服務，同意如何處理和衡量此類不可分配支出的機制。例如，請參閱《[AWS Resource Groups 和標籤使用者指南](#)》中的 [和標籤編輯器](#) 支援的資源清單。AWS Resource Groups

無法標記的成本類別的常見範例是承諾型折扣的一些費用，例如預留執行個體 (RI) 和 Savings Plans (SP)。雖然訂閱費用和未使用的 SP 和 RI 費用無法在出現在帳單報告工具之前加上標籤，但您可以在事後追蹤 RI 和 SP 折扣如何套用至中的 AWS Organizations 帳戶、資源及其標籤。例如，AWS Cost Explorer 您可以查看攤銷成本，依相關標籤索引鍵花費的群組，並套用與您使用案例相關的篩選條件。在 AWS 成本和用量報告 (CUR) 中，您可以篩選出與 RI 和 SP 折扣涵蓋的用量對應的行 (在 [CUR 文件](#) 的使用案例區段中閱讀更多資訊)，然後選取僅與您相關的欄。為成本分配啟用的每個標籤索引鍵都會在 CUR 報告結尾顯示在自己的個別資料欄中，類似於它在其他舊版帳單報告中呈現的方式，例如[每](#)

[月成本分配報告](#)。如需其他參考，請參閱 [AWS Well-Architected 實驗室](#) 以取得從 CUR 資料取得成本和用量洞察的範例。

報告

除了可用於協助進行表演和收費 AWS 的工具之外，還有一系列其他 AWS 建立和第三方解決方案，可協助監控標記資源的成本，並衡量標記策略的有效性。根據組織的需求和最終目標，可以投入時間和資源來建置自訂解決方案，或購買其中一個 [AWS 雲端 管理工具能力合作夥伴](#) 提供的工具。如果您決定使用與業務相關的控制參數建立自己的單一事實來源成本分配工具，AWS 成本和用量報告 (CUR) 會提供最詳細的成本和用量資料，並啟用建立自訂最佳化儀表板，允許依帳戶、服務、成本類別、成本分配標籤和多個其他維度進行篩選和分組。在開發 AWS 的 CUR 型解決方案中，可以用作這些工具之一，請查看 AWS Well-Architected 實驗室網站上的 [雲端智慧儀表板](#)。

操作和支援的標籤

AWS 環境會有多個帳戶、資源和工作負載，且具有不同的操作需求。標籤可用來提供內容和指引，以支援營運團隊，以增強您服務的管理。標籤也可以用來提供受管資源的操作控管透明度。

推動營運標籤一致定義的一些主要因素包括：

- 在自動化基礎設施活動期間篩選資源。例如，部署、更新或刪除資源時。另一個是擴展資源以最佳化成本和減少非工作時間用量。如需工作範例，請參閱 [AWS 執行個體排程器](#) 解決方案。
- 識別隔離或棄用的資源。超過定義生命週期或已被內部機制標記為隔離的資源應適當加上標籤，以協助支援人員進行調查。取代的資源應在隔離、封存和刪除之前加上標籤。
- 一組資源的支援需求。資源通常有不同的支援需求，例如，這些需求可以在團隊之間協商，或設定為應用程式關鍵性的一部分。如需操作模型的進一步指引，請參閱 [卓越營運支柱](#)。
- 增強事件管理程序。透過使用在事件管理程序中提供更高透明度的標籤來標記資源，支援團隊和工程師以及主要事件管理 (MIM) 團隊可以更有效地管理事件。
- 備份。標籤也可以用來識別資源需要備份的頻率，以及備份複本需要前往何處或要還原備份的位置。 [備份和復原方法的方案指引 AWS](#)。
- 修補。在 中執行的修補可變執行個體對於您的整體修補策略和零時差漏洞修補 AWS 都至關重要。更深入的修補策略指引，請參閱 [規範性指引](#)。本 [部落格](#) 討論了零時差漏洞的修補。
- 操作可觀測性。將操作 KPI 策略翻譯為資源標籤，有助於營運團隊更好地追蹤是否達到目標，以增強業務需求。制定 KPI 策略是一個單獨的主題，但往往專注於穩定狀態或衡量變革影響和結果的業務。 [KPI 儀表板](#) (AWS Well-Architected 實驗室) 和操作 KPI 研討會 ([AWS 企業支援主動式服務](#)) 都測量穩定狀態的效能。AWS 企業策略部落格文章 [測量轉型的成功](#)，探索轉型計畫的 KPI 測量，例如 IT 現代化或從內部部署遷移到內部部署 AWS。

自動化基礎設施活動

管理基礎設施時，標籤可用於各種自動化活動。例如，使用 [AWS Systems Manager](#) 可讓您管理您建立之已定義鍵值對所指定資源上的自動化和 Runbook。對於受管節點，您可以定義一組標籤，以透過作業系統和環境來追蹤或鎖定節點。然後，您可以為群組中的所有節點執行更新指令碼，或檢閱這些節點的狀態。[Systems Manager 資源](#) 也可以加上標籤，以進一步精簡和追蹤您的自動化活動。

自動化環境資源的開始和停止生命週期，可為任何組織提供大幅降低成本。[上的執行個體排程器 AWS](#) 是解決方案的範例，可在不需要時啟動和停止 Amazon EC2 和 Amazon RDS 執行個體。例如，使用週末不需要執行的 Amazon EC2 或 Amazon RDS 執行個體的開發人員環境，不會利用這些執行個體關閉時可提供的成本節省潛力。透過分析團隊及其環境的需求，並正確標記這些資源以自動化其管理，您可以有效地利用預算。

執行個體排程器在 Amazon EC2 執行個體上使用的範例排程標籤：

```
{
  "Tags": [
    {
      "Key": "Schedule",
      "ResourceId": "i-1234567890abcdef8",
      "ResourceType": "instance",
      "Value": "mon-9am-fri-5pm"
    }
  ]
}
```

工作負載生命週期

檢閱支援操作資料的準確性。請確定定期檢閱與您的工作負載生命週期相關聯的標籤，且適當的利益相關者會參與這些檢閱。

表 7 – 在工作負載生命週期中檢閱操作標籤

使用案例	標籤索引鍵	理由	範例數值
帳戶擁有者	example-incident:account-owner:owner	帳戶的擁有者及其包含的資源。	ops-center , dev-ops, app-team

使用案例	標籤索引鍵	理由	範例數值
帳戶擁有者檢閱	example-inc:account-owner:review	檢閱帳戶擁有權詳細資訊是最新且正確的。	< 以標記程式庫中定義的正確格式檢閱日期 >
資料擁有者	example-inc:data-owner:owner	資料所在帳戶的資料擁有者。	bi-team, logistics , security
資料擁有者檢閱	example-inc:data-owner:review	檢閱資料擁有權詳細資訊為最新且正確的。	< 以標記程式庫中定義的正確格式檢閱日期 >

將標籤指派給暫停帳戶，然後再遷移到暫停的 OU

如[使用多個帳戶組織您的 AWS 環境](#)白皮書所述，暫停帳戶並移至暫停的 OU 之前，應將標籤新增至帳戶，以協助內部追蹤和稽核帳戶生命週期。例如，組織 ITSM 票證系統上的相對 URL 或票證參考，顯示暫停之應用程式的稽核線索。

表 8 - 當工作負載生命週期進入新階段時新增操作標籤

使用案例	標籤索引鍵	理由	範例數值
帳戶擁有者	example-inc:account-owner:owner	帳戶的擁有者及其包含的資源。	ops-center , dev-ops, app-team
資料擁有者	example-inc:data-owner:owner	資料所在帳戶的資料擁有者。	bi-team, logistics , security
暫停日期	example-inc:suspension:date	帳戶暫停的日期	< 以標記程式庫中定義的正確格式暫停日期 >

使用案例	標籤索引鍵	理由	範例數值
暫停的核准	example-inc:suspension:approval	帳戶停用核准的連結	workload/ deprecation

事件管理

從事件記錄、優先順序、調查、通訊、解決到關閉，標籤可以在事件管理的所有階段扮演重要角色。

標籤可詳細說明記錄事件的位置、應通知事件的團隊或團隊，以及定義的呈報優先順序。請務必記住，標籤未加密，因此請考慮您存放在其中的資訊。此外，在組織、團隊和報告部門中，責任會變更，因此請考慮將連結儲存到安全入口網站，以便更有效地管理此資訊。這些標籤不需要是排他性的。例如，應用程式 ID 可用於在 IT 服務管理入口網站中查詢呈報路徑。請確定您的操作定義清楚此標籤正用於多個用途。

操作需求標籤也可以詳細說明，以協助事件管理員和操作人員進一步完善其目標，以回應事件或事件。

[執行手冊](#)和[程序手冊](#)的相對連結（知識系統基礎 URL）可以包含為標籤，以協助回應團隊識別對應的程序、程序和文件。

表 9 - 使用操作標籤通知事件管理

使用案例	標籤索引鍵	理由	範例數值
事件管理	example-inc:incident-management:escalationlog	支援團隊用來記錄事件的系統	jira, servicenow , zendesk
事件管理	example-inc:incident-management:escalationpath	呈報路徑	ops-center , dev-ops, app-team
成本分配和事件管理	example-inc:cost-a	依成本中心監控成本。這是雙重使用標籤的範例，其中使用成	123 - *

使用案例	標籤索引鍵	理由	範例數值
	llocation: CostCenter	本中心做為事件記錄 的應用程式碼	
排程備份	example-inc:backup:schedule	資源的備份排程	Daily
手冊/事件管理	example-inc:incident-management:playbook	記錄的手冊	webapp/incident/playbook

修補

組織可以使用 AWS Systems Manager Patch Manager 和 自動化其針對可變運算環境的修補策略，並使可變執行個體與該應用程式環境定義的修補基準保持一致 AWS Lambda。這些環境中可變執行個體的標記策略可以透過將上述執行個體指派給修補程式群組和維護 Windows 來管理。請參閱下列開發 → 測試 → 生產分割的範例。AWS 規範指引可用於[可變執行個體的修補程式管理](#)。

表 10 - 操作標籤可以是環境特定的

開發	安裝	生產
<pre>{ "Tags": [{ "Key": "Maintenance Window", "ResourceId": "i-012345678ab9ab1 11", "ResourceType": "instance", "Value": "cron(30 23 ? * TUE#1 *)" }], {</pre>	<pre>{ "Tags": [{ "Key": "Maintenance Window", "ResourceId": "i-012345678ab9ab4 44", "ResourceType": "instance", "Value": "cron(30 23 ? * TUE#2 *)" }], {</pre>	<pre>{ "Tags": [{ "Key": "Maintenance Window", "ResourceId": "i-012345678ab9ab7 77", "ResourceType": "instance", "Value": "cron(30 23 ? * TUE#3 *)" }], {</pre>

開發	安裝	生產
<pre> "Key": "Name", "ResourceId": "i-012345678ab9ab2 22", "ResourceType": "instance", "Value": "WEBAPP" }, { "Key": "Patch Group", "ResourceId": "i-012345678ab9ab3 33", "ResourceType": "instance", "Value": "WEBAPP-DEV- AL2" }] } </pre>	<pre> "Key": "Name", "ResourceId": "i-012345678ab9ab5 55", "ResourceType": "instance", "Value": "WEBAPP" }, { "Key": "Patch Group", "ResourceId": "i-012345678ab9ab6 66", "ResourceType": "instance", "Value": "WEBAPP-TEST- AL2" }] } </pre>	<pre> "Key": "Name", "ResourceId": "i-012345678ab9ab8 88", "ResourceType": "instance", "Value": "WEBAPP" }, { "Key": "Patch Group", "ResourceId": "i-012345678ab9ab9 99", "ResourceType": "instance", "Value": "WEBAPP-PROD- AL2" }] } </pre>

零時差漏洞也可以透過定義標籤來補充修補策略來管理。如需詳細指引，[請參閱使用 AWS Systems Manager 避免具有同日安全修補的零時差漏洞。](#)

操作可觀測性

需要可觀測性，才能獲得對環境效能的可行洞見，並協助您偵測和調查問題。它也有次要用途，可讓您定義和測量關鍵績效指標 (KPIs) 和服務水準目標 (SLOs)，例如執行時間。對於大多數組織而言，重要的操作 KPIs 是偵測的平均時間 (MTTD) 和從事件復原的平均時間 (MTTR)。

在整個可觀測性中，內容很重要，因為收集資料，然後收集關聯的標籤。無論您關注的服務、應用程式或應用程式層為何，都可以篩選和分析該特定資料集。標籤可用來自動加入 CloudWatch 警示，以便在超過特定指標閾值時發出提醒。例如，標籤索引鍵 `example-inc:ops:alarm-tag` 及其值可能表示建立 CloudWatch 警示。[使用標籤來建立和維護 Amazon EC2 執行個體的 Amazon CloudWatch 警示](#) [Amazon EC2](#) 中會說明示範此作法的解決方案。

設定太多警示可以輕鬆建立警示風暴 - 當大量警示或通知快速讓操作員負擔過重，並在操作員手動分類和排定個別警示的優先順序時降低其整體效率。警示的其他內容可以以標籤的形式提供，這表示規則可以在 Amazon EventBridge 中定義，以協助確保將焦點集中在上游問題，而不是下游相依性。

DevOps 的操作角色 DevOps 通常被忽略，但對於許多組織而言，中央營運團隊仍然在正常營業時間之外提供重要的第一個回應。（如需此模型的詳細資訊，請參閱[卓越營運白皮書](#)。）與擁有工作負載的 DevOps 團隊不同，他們通常沒有相同的知識深度，因此標籤在儀表板和警示中提供的內容，可以引導他們找到問題的正確 Runbook，或啟動自動化 Runbook（請參閱部落格文章[使用 自動化 Amazon CloudWatch 警示 AWS Systems Manager](#)）。

資料安全、風險管理和存取控制的標籤

組織在適當處理資料儲存和處理方面，需要滿足不同的需求和義務。資料分類是數個使用案例的重要前綴，例如存取控制、資料保留、資料分析和合規。

資料安全和風險管理

在 AWS 環境中，您可能擁有具有不同合規和安全性需求的帳戶。例如，您可能有一個開發人員沙盒，以及一個帳戶託管高管制工作負載的生產環境，例如處理付款。透過將它們隔離到不同的帳戶，您可以[套用不同的安全控制](#)、[限制對敏感資料的存取](#)，並減少受管制工作負載的稽核範圍。

對所有工作負載採用單一標準可能會導致挑戰。雖然許多控制項同樣適用於環境，但對於不需要符合特定法規架構的帳戶，以及永遠不會有個人識別資料的帳戶（例如，開發人員沙盒或工作負載開發帳戶），有些控制項是過多或不相關的。這通常會導致誤判安全問題清單，必須分類並關閉，無需採取任何動作，這需要擺脫應調查的問題清單。

表 11 – 資料安全和風險管理標籤範例

使用案例	標籤索引鍵	理由	範例數值
事件管理	example-inc:incident-management:escalationlog	支援團隊用來記錄事件的系統	jira, servicenow , zendesk
事件管理	example-inc:incident-management:escalationpath	呈報路徑	ops-center , dev-ops, app-team

使用案例	標籤索引鍵	理由	範例數值
資料分類	example-inc:data:classification	分類資料以遵循法規和控管	Public, Private, Confidential , Restricted
合規	example-inc:compliance:framework	識別工作負載受制於的合規架構	PCI-DSS, HIPAA

手動管理整個 AWS 環境的不同控制項既耗時又容易出錯。下一個步驟是自動化部署適當的安全控制，並根據該帳戶的分類設定資源檢查。透過將標籤套用至帳戶和其中的資源，可以針對工作負載適當自動化和設定控制項的部署。

範例：

工作負載包含具有 標籤的 Amazon S3 儲存貯體，其值example-inc:data:classification為 Private。安全工具自動化會部署 AWS Config 規則 s3-bucket-public-read-prohibited，檢查 Amazon S3 儲存貯體的封鎖公開存取設定、儲存貯體政策和儲存貯體存取控制清單 (ACL)，確認儲存貯體的組態適合其資料分類。為了確保儲存貯體的內容與 分類一致，[Amazon Macie 可以設定為檢查個人識別資訊 \(PII\)](#)。部落格[使用 Amazon Macie 驗證 S3 儲存貯體資料分類](#)會更深入探索此模式。

某些法規環境，例如保險和醫療保健，可能受到強制性資料保留政策的約束。使用標籤與 Amazon S3 生命週期政策結合的資料保留，是將物件轉換範圍限定到不同儲存層的有效簡單方式。Amazon S3 生命週期規則也可以在強制保留期間到期後，用來使資料刪除的物件過期。請參閱[搭配 Amazon S3 生命週期使用物件標籤來簡化資料生命週期](#)，以取得此程序的深入指南。

此外，在分類或處理安全調查結果時，標籤可以為調查人員提供有助於降低風險的重要內容，並協助適當的團隊調查或緩解調查結果。

身分管理和存取控制的標籤

AWS 使用 管理環境中的存取控制時 AWS IAM Identity Center，標籤可以啟用多種模式以進行擴展。可以套用數種委派模式，有些是以標記為基礎。我們會個別處理它們，並提供連結以進一步閱讀。

個別資源的 ABAC

IAM Identity Center 使用者和 IAM 角色支援屬性型存取控制 (ABAC)，可讓您根據標籤定義對操作和資源的存取。ABAC 有助於減少更新許可政策的需求，並協助您從公司目錄基礎存取員工屬性。如果您已

經使用多帳戶策略，除了角色型存取控制 (RBAC) 之外，還可以使用 ABAC，為在同一帳戶中操作的多個團隊提供對不同資源的精細存取權。例如，IAM Identity Center 使用者或 IAM 角色可以包含限制存取特定 Amazon EC2 執行個體的條件，否則必須明確列出在每個政策中才能存取這些執行個體。

由於 ABAC 授權模型取決於存取操作和資源的標籤，因此請務必提供護欄以防止意外存取。SCPs 可用來保護整個組織的標籤，只允許在特定條件下修改標籤。部落格[使用中的服務控制政策和 IAM 實體的許可界限保護用於授權的資源標籤 AWS Organizations](#)，提供如何實作此項目的資訊。https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html

如果長期的 Amazon EC2 執行個體用於支援更傳統的操作實務，則可以使用此方法，部落格[為 Amazon EC2 執行個體設定 IAM Identity Center ABAC，Systems Manager Session Manager](#) 會更詳細地討論這種形式的屬性型存取控制。如前所述，並非所有資源類型都支援標記，而且並非所有資源類型都支援使用標籤政策強制執行，因此在上開始實作此策略之前，最好先評估這一點 AWS 帳戶。

若要了解支援 ABAC 的服務，請參閱[AWS 使用 IAM 的服務](#)。

結論

AWS 資源可以針對各種用途進行標記，從實作成本分配策略到支援自動化或授權存取 AWS 資源。由於涉及的利益相關者群組數量以及資料來源和標籤控管等考量，對某些組織而言，實作標記策略可能具有挑戰性。

在本白皮書中，我們概述了有關根據營運實務、定義的使用案例、涉及程序的利益相關者，以及提供的工具和服務在組織中設計和實作標記策略的建議 AWS。涉及標記策略時，這是一個反覆運算和改進的過程，從立即的優先順序開始，識別整個組織中相關的使用案例，然後根據需要實作和擴展標記結構描述，同時持續衡量和提高有效性。我們已指出，組織內一組定義明確的標籤，可讓您將 AWS 用量和用量與負責資源和業務目的的團隊相關聯，以符合組織策略和價值。

貢獻者

本文件的貢獻者包括：

- Chris Pates , Amazon Web Services 資深專家技術客戶經理
- Vijay Shekhar Rao , Amazon Web Services 企業支援主管
- Nataliya Godunok , Amazon Web Services 資深專家技術客戶經理
- Yogish Kutkunje Pai , Amazon Internet Services Private Limited 資深解決方案架構師
- Jamie Ibbs , Amazon Web Services 資深專家技術客戶經理

深入閱讀

如需詳細資訊，請參閱

- [AWS re : Invent 2020：向後工作：Amazon 的創新方法](#)
- [AWS 方案指引：使用 AWS Systems Manager 自動修補混合雲端中的可變執行個體](#)
- [AWS 架構中心](#)

AWS Well-Architected

- [AWS Well-Architected 架構](#)
- [卓越營運支柱 - AWS Well-Architected Framework](#)
- [災難復原計劃 \(DR\) - AWS Well-Architected 可靠性支柱](#)
- [成本最佳化支柱 - AWS Well-Architected Framework](#)
- [AWS Well-Architected 實驗室：啟用 AWS 產生的成本分配標籤](#)
- [AWS Well-Architected 實驗室：標籤政策](#)
- [AWS Well-Architected 實驗室：AWS CUR 查詢程式庫](#)

AWS 部落格

- [AWS Health 感知 – 自訂組織和個人 AWS 帳戶的 AWS Health 提醒](#)
- [如何自動標記 Amazon EC2 資源以回應 API 事件](#)
- [AWS 產生與使用者定義的成本分配標籤](#)
- [使用的成本標記和報告 AWS Organizations](#)
- [使用修補程式管理員 AWS Systems Manager 修補您的 Windows EC2 執行個體](#)
- [使用 避免具有同日安全修補的零時差漏洞 AWS Systems Manager](#)

AWS 文件

- [使用成本分配標籤 - AWS 帳單與成本管理 以及成本管理和成本管理](#)
- [什麼是 AWS 成本和用量報告](#)
- [AWS Resource Groups API 參考](#)
- [如何使用 IAM 政策標籤來限制如何建立 EC2 執行個體或 EBS 磁碟區？](#)

- [可變與不可變的更新模型](#)

Other (其他)

- Bryar , C. 和 Carr , B. (2021 年) 。 [向後工作：Amazon 內部的洞見、案例和秘密](#)。倫敦馬克米蘭。
- [AWS CloudFormation Guard](#) (GitHub)

文件修訂

若要收到此白皮書更新的通知，請訂閱 RSS 摘要。

變更	描述	日期
次要更新	釐清指引，建議不要使用僅因大小寫而不同的金鑰。	2025 年 6 月 10 日
次要更新	已變更 CloudEndure 參考 AWS Application Migration Service。	2025 年 5 月 8 日
次要更新	身分管理的更新	2023 年 3 月 30 日
次要修訂版	更新了 ABAC 中個別資源的參考。	2023 年 2 月 24 日
次要修訂版	更新了指南以符合 IAM 最佳實務。如需更多詳細資訊，請參閱 IAM 中的安全最佳實務 。	2023 年 2 月 6 日
主要修訂版	針對 AWS Config 規則 支援的資源類型新增更具體的參考 <code>required_tags</code> 。	2023 年 1 月 18 日
主要修訂版	更新以包含最新的實務和服務功能，尤其是在身分領域。	2022 年 9 月 29 日
次要更新	修正 PDF 版本的資料表格式。	2022 年 4 月 25 日
主要修訂版	更新文件結構並擴展了標記策略和使用案例章節。根據最新的工具、技術和可用資源，新增了更多規範性指引。	2022 年 4 月 22 日
初次出版	白皮書首次發佈。	2018 年 12 月 1 日

 Note

若要訂閱 RSS 更新，必須為所使用的瀏覽器啟用 RSS 外掛程式。

注意

客戶有責任對本文件中的資訊進行自己的獨立評定。本文件：(a) 僅供參考，(b) 代表目前的 AWS 產品和實務，這些產品和實務如有變更，恕不另行通知，且 (c) 不會從 AWS 及其附屬公司、供應商或授權方提供「原樣」的任何承諾 AWS 或保證，而無任何明示或暗示的保證、聲明或條件。AWS 對其客戶的責任和責任由 AWS 協議控制，本文件不屬於，也不會修改 AWS 與其客戶之間的任何協議。

© 2022 Amazon Web Services, Inc. 或其附屬公司。保留所有權利。

AWS 詞彙表

如需最新的 AWS 術語，請參閱 AWS 詞彙表 參考中的[AWS 詞彙表](#)。

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。