



在 上實作基礎設施做為產品 (IaP) AWS

AWS 方案指引



AWS 方案指引: 在上實作基礎設施做為產品 (IaP) AWS

Copyright © 2026 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

簡介	1
為什麼要將基礎設施管理為產品？	1
目標業務成果	1
使用 AWS Service Catalog 管理 IaP	3
支援模組化和程式碼重複使用	4
在 Service Catalog 中定義產品的程式設計選項	5
CloudFormation 指令碼	5
使用 的程式設計方法 AWS CDK	5
與外部佈建程序和工作流程整合	6
產品佈建規格	7
DevSecOps 生命週期支援	7
自訂重複使用和帳戶特定的佈建	7
將 Service Catalog 產品資源定義為應用程式，並進行管理	7
庫存管理	8
使用 AWS Service Catalog 工具	9
Service Catalog Puppet	9
支援佈建工作流程	9
佈建模式	10
快取	11
DevSecOps 生命週期支援	11
成熟度、完整性和支援	12
Service Catalog 工廠	12
摘要和後續步驟	13
資源	14
文件歷史紀錄	15
.....	xvi

在上實作 IaP AWS

Kirsten Kissmeyer , Amazon Web Services (AWS)

2023 年 1 月 ([文件歷史記錄](#))

本指南探索以產品 (IaP) 形式管理 AWS 基礎設施的方法。IaP 提供比基礎設施即程式碼 (IaC) 更高層級的抽象和控制，但使用 IaC 方法來實現其目標。本指南也會探索 AWS 服務 和管理 IaP 的工具，並強調每個工具如何支援您管理基礎設施的目標。本指南中的資訊是根據從非常大型金融產業公司的 AWS Service Catalog 啟用計畫學習而得。

本指南適用於想要開發功能 AWS 雲端 基礎設施服務的使用者，這些服務可視需要為不同的組織使用者、業務單位和第三方輕鬆配置和授權。

為什麼要將基礎設施管理為產品？

將基礎設施資源作為產品管理的優點是，您可以將消費者功能封裝為一組具有標準化定義和組態的資源。產品為組織提供便利的方式來管理和控制 AWS 功能的配置和使用方式。產品可能僅限於指定的[組織單位 \(OUs\)](#) 或需要這些功能的個人。產品 AWS 區域 也可以限制為特定產品。

產品佈建模型也可讓您從中央位置封裝和更新產品的定義。然後，您可以一次性或排程地分發產品更新，因為其實作會隨著時間而變更。

目標業務成果

組織一律尋找更好的方法來管理和佈建其 AWS 基礎設施。您的目標可能包括：

- 實現高度的敏捷性、可靠性、容錯能力和集中控制，其中單一組態點符合不斷發展的內部和外部標準。
- 一種低接觸或按鈕機制，以集中的方式分配基礎設施，同時允許特定團隊或個人在需要時進行自助式存取。
- 能夠將 AWS 基礎設施和服務佈建給內部員工、用戶端帳戶和合作夥伴 OU 帳戶。您可能也想要控制哪些 OUs 或組織可以存取特定區域中的特定基礎設施元件。
- 如果您使用第三方工具（例如 ServiceNow）或自訂工具來管理存取和佈建企業資產和基礎設施的請求，請在您的 AWS 基礎設施和這些工具之間輕鬆整合。
- 能夠同時將 AWS 基礎設施佈建至數十甚至數百個目標帳戶。

- 支援佈建多個 AWS 資源以提供單一功能。
- 能夠在緊密的排程內建立具有所需基礎設施的新帳戶。
- 存取您已佈建之基礎設施的庫存，以及更新或移除基礎設施元件的能力。
- 讓佈建和維護程序更輕鬆、更快速且更安全可靠的方法和技術。

使用 AWS Service Catalog 管理 IaP

AWS 提供稱為的服務[AWS Service Catalog](#)，可支援管理和佈建 AWS 基礎設施做為產品。您可以使用 Service Catalog 快速定義您需要佈建為一組產品的基礎設施、將這些產品的許可授予所需對象，以及實作個別產品所需的佈建和更新模式。

Service Catalog 由支援[AWS CloudFormation](#)。Service Catalog 產品組合、產品及其佈建範本以 CloudFormation 堆疊管理。您可以透過四種方式定義這些堆疊：

- 使用標準 CloudFormation 範本。
- 透過使用 [AWS Cloud Development Kit \(AWS CDK\)](#)和 [Service Catalog Construct Library](#) 搭配您偏好的支援程式設計語言。
- 透過使用第三方工具提供的架構，從描述堆疊的宣告中繼資料產生 CloudFormation 堆疊定義。
- 使用 [Service Catalog API](#)。此 API 提供除了建置產品以外的所有方法。您可以將產品新增至產品組合、從產品組合中移除產品、標記產品和產品組合、定義管理和操作產品服務動作，以及瀏覽和搜尋產品組合和產品定義。

Service Catalog 產品的核心是一組一個或多個 AWS 資源，這些資源設定為提供可自訂的集體（透過參數化）功能。例如，您可以定義 Service Catalog 產品，以在目標帳戶中佈建私有 Amazon Simple Storage Service (Amazon S3) 儲存貯體。S3 儲存貯體是可能具有輸入參數的產品，例如儲存貯體名稱、允許存取的網際網路地址範圍、可存取儲存貯體的一組使用者、生命週期分層政策或儲存貯體版本控制規格。您也可以定義 AWS Identity and Access Management (IAM) 角色，以提供對儲存貯體的存取，做為產品的一部分。

您可以將 Service Catalog 產品新增至一或多個產品組合。Service Catalog 產品組合是集合在一起的產品，通常因為它們具有類似的用途（例如，分析、開發、用戶端存取服務、合作夥伴存取服務等）。

您可以為使用者、群組或角色提供許可，以存取在產品組合層級佈建產品。對於佈建，產品會與啟動 IAM 角色（以自助方式向可擔任該角色的任何人啟動產品）或定義產品可佈建之一或多個帳戶的[堆疊集](#)相關聯。若要使用堆疊集，您必須在 Service Catalog 中樞帳戶中定義 Service Catalog 管理員角色，並在堆疊集的每個目標帳戶中定義 Service Catalog 產品佈建執行角色。

下列各節會更詳細討論 Service Catalog IaP 功能。

主題

- [支援模組化和程式碼重複使用](#)
- [在 Service Catalog 中定義產品的程式設計選項](#)

- [與外部佈建程序和 workflows 整合](#)
- [產品佈建規格](#)
- [DevSecOps 生命週期支援](#)
- [自訂重複使用和帳戶特定的佈建](#)
- [將 Service Catalog 產品資源定義為應用程式，並進行管理](#)
- [庫存管理](#)

支援模組化和程式碼重複使用

您可以從許多不同的 AWS 資源，甚至從其他產品組合產品。理想情況下，您會以模組化方式定義資源，以便在多個產品中重複使用這些資源。資源層級重複使用可讓您在某個位置進行任何未來的變更，而不是使用該資源類型的每個產品。

Service Catalog 提供稱為鏈結的功能，以支援產品層級的可重複使用性。您可以將產品連結到一或多個其他產品。例如，您可能想要將 S3 記錄儲存貯體產品鏈結至更高層級的監控產品。雖然鏈結支援模組化，但它會產生一些操作複雜性，因為您必須管理相依性。Service Catalog 不會在鏈結產品之間自動維護版本控制，因此無法確保對某個產品的變更不會破壞依賴它的其他產品。請謹慎使用鏈結，並開發自己的機制，以確保版本控制和維護相依性。

Service Catalog 原生使用 CloudFormation 將產品佈建範本部署為 CloudFormation 堆疊。不過，Service Catalog 會對產品堆疊的 CloudFormation 部署施加一些限制。特別是，Service Catalog 佈建不支援 CloudFormation `include` 巨集來插入可重複使用的指令碼區段，或參考巢狀 CloudFormation 指令碼（或堆疊）到多個層級。這些 Service Catalog 限制會限制從可重複使用的 CloudFormation 範本或元件定義產品的能力，這是當您在 CloudFormation 中原生定義堆疊時的標準最佳實務。

Note

Service Catalog 可讓您成功定義具有使用這些 CloudFormation 建構的佈建範本的產品。不過，如果您在 Service Catalog CloudFormation 範本中使用 `include` 巨集或巢狀多個層級的指令碼，將會遇到佈建時間錯誤。

這些限制可能會使得在 Service Catalog 中實作模組化和可重複使用的產品變得困難。如果模組化是必要項目，您可以探索[使用 AWS CDK 實作您的產品及其佈建範本](#)，或在[AWS Labs Service Catalog Tools 專案](#)中使用佈建工作流程和引擎。本指南稍後將說明這兩種替代方案。

在 Service Catalog 中定義產品的程式設計選項

使用 Service Catalog 佈建 AWS 基礎設施的兩個程式設計選項是 CloudFormation 範本或 AWS CDK。目前沒有用於定義 Service Catalog 產品的宣告式或無程式碼機制。

CloudFormation 指令碼

CloudFormation 是用於佈建 AWS 基礎設施的已嘗試且真實的 IaC 原生指令碼語言。您可以在中或使用 Visual Studio Code (或簡單文字編輯器) 和 () 等開發工具 AWS 管理主控台 來開發 CloudFormation AWS Command Line Interface 指令碼AWS CLI。

如需詳細資訊，請參閱 [CloudFormation 文件](#)。如需使用 CloudFormation 範本指定 Service Catalog 產品的詳細資訊，請參閱 [AWS::ServiceCatalog::CloudFormationProduct 資源](#)。CloudFormation

使用 的程式設計方法 AWS CDK

AWS CDK 提供優雅且功能強大的物件導向程式設計架構，可讓您使用多種程式設計語言來定義和維護 AWS 基礎設施。您可以使用 AWS CDK 來開發 AWS 類別架構的物件導向、精細自訂和延伸。AWS CDK 適用於想要 AWS 服務 自訂更複雜基礎設施需求的使用者，以及具有必要程式設計技能和經驗的使用者。

若要使用 實作 Service Catalog 解決方案 AWS CDK，您可以使用內建的 Service Catalog 類別來定義產品和產品組合。這些類別由 AWS CDK [aws-cdk-lib.aws_servicecatalog 模組提供](#)。

您可以使用 以多種方式 AWS CDK 實作 產品。為了避免在 CloudFormation 中為產品撰寫佈建範本並維持可重複使用性，我們建議您使用 AWS CDK [ProductStack 類別](#)來代表佈建範本。ProductStack 執行個體是您以程式設計方式新增資源的 AWS CDK 堆疊。例如，您可以新增 S3 儲存貯體、IAM 角色或 Amazon CloudWatch 日誌。當您呼叫 將ProductStack執行個體新增至定義的servicecatalog.CloudFormationProduct執行個體做為其佈建範本時servicecatalog.CloudFormationTemplate.fromProductStack (<ProductStack instance>)，AWS CDK 會自動產生 CloudFormation 範本。

以下是 Amazon S3 產品的 Java ProductStack實作範例。

```
import * as s3 from 'aws-cdk-lib/aws-s3';
import * as cdk from 'aws-cdk-lib';

class S3BucketProduct extends servicecatalog.ProductStack {
  constructor(scope: Construct, id: string) {
```

```
super(scope, id);

new s3.Bucket(this, 'BucketProduct');
}
}

const product = new servicecatalog.CloudFormationProduct(this, 'Product', {
  productName: "My Product",
  owner: "Product Owner",
  productVersions: [
    {
      productVersionName: "v1",
      cloudFormationTemplate:
        servicecatalog.CloudFormationTemplate.fromProductStack(new S3BucketProduct(this,
          'S3BucketProduct')),
    },
  ],
});
```

AWS CDK 提供內建的持續整合和持續部署 (CI/CD) 管道。您可以自訂這些內建管道和軟體開發生命週期 (SDLC) 程序，以符合您自己的程序標準和目標。

自訂 AWS CDK 類別可以從其他類別繼承，以提供專門的函數，而且類別可以由其他類別的執行個體組成。如果您使用共用 AWS CDK 類別架構來實作多個 Service Catalog 產品，請考慮任何版本控制或相容性影響，尤其是跨多個開發團隊。您必須確保變更是回溯相容的，或者您有遵循的版本控制機制，以便您對某個產品的類別變更不會破壞另一個產品。

如需詳細資訊，請參閱 [AWS CDK 文件](#)。

與外部佈建程序和 workflows 整合

您可以使用 AWS SDK APIs 或與 Service Catalog 元件互動 AWS CLI。您可以使用 [AWS SDK Service Catalog API](#)，從任何可以整合 Service Catalog API 呼叫的工具來管理 Service Catalog 產品。API 涵蓋 Service Catalog 建立和管理的所有層面。例如，Terraform 透過在其啟動精靈中呼叫 AWS SDK Service Catalog API 來支援啟動（佈建）Service Catalog 產品。如需詳細資訊，請參閱 AWS 文件中的 [使用 Terraform 啟動 AWS Service Catalog 產品](#)。

您也可以呼叫 AWS CLI Service Catalog 命令，在 Service Catalog 上執行動作。如需支援命令的詳細資訊，請參閱《AWS CLI 命令參考》中的 [servicecatalog](#)。

產品佈建規格

Service Catalog 會以 CloudFormation 堆疊集部署的形式啟動佈建程序，部署 CloudFormation 佈建範本中指定的資源。(範本可直接在 AWS CDK ProductStack 中建立 CloudFormation 或由建構產生。) Service Catalog 產品佈建是已關閉的程序，您無法自訂它來新增初步或後製程序步驟，或對其進行調校。不過，您可以修改佈建範本，以新增 CloudFormation 資源規格形式的步驟。這些可以是 AWS Lambda 或 Lambda 支援的自訂資源 AWS Step Functions，這些資源會執行初步步驟（例如自訂引導以設定佈建期間使用的堡壘主機）和後步（例如移除堡壘主機）。此實作佈建前和佈建後步驟的方法，會受到 include 與佈建範本相同的巢狀堆疊限制。

您可以將目標帳戶指定為個別帳戶，而不是組織單位 (OUs)。您可以撰寫自訂資源或函數來解決此限制。大多數組織會佈建產品組合給 OUs，而不是個別帳戶，因為他們會自動產生帳戶，而且不想手動維護帳戶清單。

DevSecOps 生命週期支援

目前，使用 Service Catalog CloudFormation 指令碼佈建的產品沒有 CI/CD 程序的內建支援。我們建議您在 AWS CodePipeline 或其他 DevOps 工具中建立 CI/CD 程序，透過開發、測試、階段和生產等生命週期環境來開發、測試和發行產品。

AWS CDK 確實為產品提供內建 CI/CD 支援，如本指南先前所述。

自訂重複使用和帳戶特定的佈建

產品應盡可能用於許多不同的自訂用途。Service Catalog 透過產品參數支援可重複使用性。您可以在佈建時提供這些參數做為產品的輸入。

您也可以將 CloudFormation 範本層級將這些參數指定為 AWS Systems Manager 參數存放區值，以套用帳戶特定值和 OU 特定值。這是 CloudFormation 佈建範本設計的最佳實務。佈建產品時，會套用目標帳戶中具名參數的值。例如，您可以將子網路參數指定為參數存放區值，並在特定 OU 帳戶的產品佈建時間套用該子網路。如需將參數存放區值作為 CloudFormation 範本參數的詳細資訊，請參閱 CloudFormation 文件中的[使用動態參考指定範本值](#)。

將 Service Catalog 產品資源定義為應用程式，並進行管理

AWS Service Catalog AppRegistry 提供集中式應用程式搜尋、報告和管理功能。AppRegistry 應用程式可以包含一或多個佈建的產品堆疊，以及與 Service Catalog 無關的 CloudFormation 堆疊。您可以

在定義為部署目標 AWS 帳戶 的 中分組和檢視所有應用程式資源集合。這些帳戶可能是您的開發、測試和生產生命週期帳戶。

您也可以使用 AppRegistry 將中繼資料屬性與應用程式建立關聯。您可以指派包含屬性集的可重複使用屬性群組。然後，您可以使用 AppRegistry 或整合服務搜尋具有指定屬性的應用程式資源並對其採取行動。這些整合服務包括：

- [Application Manager](#) 是的一項功能 AWS Systems Manager，可在您的應用程式和叢集環境中調查和修復 AWS 資源的問題
- [AWS Resource Access Manager](#)，與 AWS 組織主體共用應用程式和屬性群組
- [AWS 使用的服務 AWS Resource Groups](#)
- [AWS Resilience Hub](#) 用於產品結構探索和彈性評估
- [AWS Service Management Connector](#) 宣告和設定與 ServiceNow、JIRA 和其他熱門工具的連線

如需 AppRegistry 的詳細資訊，請參閱下列內容：

- [AppRegistry 管理員指南](#)。
- [使用部落格文章提高應用程式可見度和管理 AWS Service Catalog AppRegistry](#)。本文概述如何在基礎設施控管中使用 AppRegistry，並提供在 AppRegistry 中將基礎設施註冊為應用程式的命令列範例。
- [使用 AppRegistry 和 Application Manager 部落格文章集中管理應用程式](#)。本文概述如何套用 AppRegistry 在上註冊 LAMP Web 應用程式 AWS 管理主控台，並使用 Application Manager 管理它。

庫存管理

Service Catalog 擁有自己的內部庫存管理功能，可在透過產品共用和自助服務佈建產品時註冊產品。不過，我們建議您使用 [AWS Config](#) 或 [AppRegistry](#) 和相關服務來管理產品佈建的資源。這些工具提供更全面且整合的方法，以其他 AWS 基礎設施管理您的佈建 Service Catalog 產品。AWS Config 可讓您在主控台或使用 AWS SDK API 對佈建產品進行清查和執行動作。AppRegistry 與 Application Manager 整合，也為 Service Catalog 佈建產品提供庫存管理。

使用 AWS Service Catalog 工具

如果您想要以更明確的方式使用自訂的佈建工作流程佈建 IaC 產品，您可能想要增強 Service Catalog 功能的一部分。AWS 提供多種工具來支援這些需求。AWS 實驗室專案提供兩種熱門的工具：Service Catalog Puppet 和 Service Catalog Factory。

主題

- [Service Catalog Puppet](#)
- [Service Catalog 工廠](#)

Service Catalog Puppet

Service Catalog Puppet 是使用 AWS Boto3 API 在 Python 中實作。此工具提供數種強大的功能來設定和佈建 Service Catalog 產品。開發人員可以使用做為資訊清單的 YAML 範本來設定 Service Catalog 產品和產品組合佈建資訊。Service Catalog Puppet 佈建工作流程支援比 Service Catalog 需要更複雜部署程序的產品。它們也支援效能最佳化，以在積極的時段內大規模佈建產品。

Service Catalog Puppet 會存取 Service Catalog CloudFormation 範本，以在部署時佈建產品。它會直接呼叫 CloudFormation 來部署產品的佈建範本堆疊，並略過 Service Catalog 自己的堆疊集佈建程序所施加的限制。如果佈建範本使用巨集來包含其他 CloudFormation 指令碼或使用巢狀 CloudFormation 指令碼，您必須在佈建工作流程的引導部分中提供目標帳戶中這些指令碼的存取權。

如需詳細資訊：

- 請參閱 Service Catalog Puppet [文件](#)和 [GitHub 儲存庫](#)。
- 如果您想要使用 Service Catalog Puppet SDK 以程式設計方式與工具互動，以啟動產品和產品組合佈建，請參閱 [SDK 文件](#)。
- [GitOps](#) 是管理 Service Catalog Puppet 環境的預設機制。

Service Catalog Puppet 相當容易讓開發人員學習。它需要熟悉 CloudFormation，才能實作產品佈建範本和 YAML 範本，才能實作資訊清單。有很好的研討會可以讓新開發人員加快速度，例如[自行安排進度的研討會](#)。

支援佈建工作流程

Service Catalog Puppet 使用 Python Luigi 任務協同運作引擎來實作引導和佈建工作流程。這些工作流程中的所有步驟都會實作為 Luigi 工作流程任務。如需 Luigi 的概觀及其與其他熱門工作流程工具

的比較，請參閱 Data Revenue 部落格中的 [Airflow 相對於 Luigi 相對於 Argo 相對於 MLFlow 相對於 KubeFlow](#)。

Luigi 允許 Service Catalog Puppet 控制與工作流程任務相關聯的工作者數量，以及控制工作流程的其他層面，以獲得更佳的擴展和效能。Service Catalog Puppet 也提供 [depends_on 機制](#) 來管理產品和步驟相依性，以及協調產品佈建。此功能可協助您實作和操作管理精細的產品定義和複雜的相依性。

佈建模式

Service Catalog Puppet 支援三種執行模式：[中樞、輻條和非同步](#)。這三種模式都會在已在 Service Catalog 中定義的產品組合中佈建產品。他們倚賴與目標帳戶的 Service Catalog 產品共用，並使用 Service Catalog 管理員和啟動角色來實現這些目標中的佈建。Service Catalog Puppet 會根據 YAML 組態檔案中提供的角色組態，在相同的組織中執行引導步驟。此工具也支援從單一中樞帳戶佈建至多個組織。在此案例中，必須在外部組織中手動執行引導，以允許 Service Catalog Puppet 在外部組織的帳戶中執行必要的佈建動作。

在所有佈建模式中，Service Catalog Puppet 會直接實作產品佈建，而無需呼叫 Service Catalog 的佈建程序。您可以設定佈建資訊清單，以在現有的 Service Catalog 堆疊集限制條件中使用角色和目標帳戶規格。Service Catalog Puppet 會使用此資訊，透過 Luigi 工作流程進行自己的佈建。

除了直接指定 OUs 或帳戶之外，您還可以根據帳戶標記方法定義產品組合佈建的目標。在以帳戶標籤為基礎的佈建中，產品組合產品會佈建到具有指定資訊清單佈建標籤集中所有標籤的所有帳戶。例如，如果您想要將產品組合產品發行到美國東部區域的所有機構生產帳戶，您可以指定標籤 `type:prod`、`partition:us-east` 和 `scope:institutional-client`。您也可以宣告帳戶和 OU 排除，以禁止佈建到具有您指定標籤 OUs 或帳戶，或佈建到屬於 OU 指定目標成員的帳戶。如需帳戶標記的詳細資訊，請參閱 [Service Catalog Tools 文件](#)。

Hub 模式

在中樞佈建模式中，發言帳戶的所有 Luigi 工作流程都會從指定的中樞帳戶管理。中樞帳戶會擔任 IAM 角色，允許它在發言帳戶中執行動作，但任務的管理是從中樞帳戶內部進行。中樞帳戶會同步等待，直到所有發言帳戶佈建任務成功完成或失敗為止。然後報告最終狀態。中樞帳戶模式是最舊且最成熟的佈建模式。不過，許多使用者已移至輻條佈建模式，以實現更大的佈建並行和速度。

呼叫模式

在輻條模式中，Service Catalog 中樞帳戶會啟動 Luigi 工作流程，以在指定的引導式輻條帳戶中執行。語音工作流程完成時，會通知中樞帳戶。發言帳戶中的故障會上升到中樞帳戶。中樞帳戶會輪詢輪輻帳戶，以查看是否已完成並判斷狀態。

輪換模式最不可能要求增加配額 AWS 服務，因為幾乎所有項目都在單獨的輪換帳戶中執行。呼叫模式也提供遠高於中樞模式的並行，同時保留中央控制。相較於中樞模式，它可以提高 800% 的佈建速度。呼叫模式支援透過產品 DependsOn 之間的關係進行產品鏈結，這可確保已佈建相依的產品。包含鏈結產品的產品也可以佈建元件鏈結產品。您也可以使用特殊化 AWS Lambda 函數呼叫來執行必要的步驟。一個輻條中的故障與其他輻條隔離。

在最多 7 個區域中擁有超過 980 個帳戶的企業會使用 呼叫模式。這些企業通常可以在一小時內將產品佈建至其基礎設施中的所有區域和帳戶。

Note

這些結果可能會因聯網基礎設施、工作負載和 AWS 組織中樞和發言帳戶的配額等因素而有所不同。它們也取決於佈建的產品資源、其固有建立時間，以及其對其他資源的相依性。

Aysnc 模式

非同步模式會在語音帳戶中啟動佈建工作流程，但不會等待或接收來自語音的完成回應。

快取

Service Catalog Puppet 用來最佳化工作流程速度的另一個機制是快取代表工作流程中步驟的常見任務。當快取任務完成時，它會將其輸出寫入 Amazon Simple Storage Service (Amazon S3)。下次在具有相同參數的相同工作階段中調用任務時，Service Catalog Puppet 會使用快取的值，而不是重新執行任務。如需詳細資訊，請參閱 Service Catalog Puppet 文件中的[快取](#)。

DevSecOps 生命週期支援

Service Catalog Puppet 包含管理 DevSecOps 管道的支援。您可以使用 Service Catalog Tools 動作（如 [Service Catalog Puppet 概觀](#) 所示）來自動化測試，並在整個 AWS 生命週期帳戶中提升產品，包括建議的 Canary 帳戶。如需詳細資訊，請參閱 Service Catalog Puppet 文件中的[管理您的環境](#)。

為了確保在廣泛生產使用之前偵測到與產品變更相關的任何問題，Service Catalog Puppet 至少需要一個 Canary 帳戶才能進行初始部署。測試並取得新版本的可信度後，您可以將其提升為非 Canary 生產帳戶。如果您發現任何問題，您可以轉返發行版本，並在問題解決時重新導入。當您使用此方法時，如果您發行對生產帳戶有問題的 Canary 版本，則可能會發生生產問題。或者，您可以在發佈變更至生產環境之前，針對每個產品變更執行完整迴歸測試。這會在 CI/CD 程序中帶來額外的額外負荷，但有助於避免生產問題。DevSecOps 管理員有責任為其開發團隊判斷最佳的功能發行案例和方法。

Service Catalog Puppet 允許多個團隊同時開發和測試 Service Catalog 產品解決方案的佈建。根據最佳實務，多個開發人員不應同時變更產品。反之，您可以將產品分解為更精細的元件，以進行個別的同時修改。

Service Catalog Puppet 也透過提供靜態和單元測試功能的聲明陳述式來協助自動化測試。您可以使用政策模擬器來測試服務控制政策 (SCPs) 和 IAM 政策。這些技術上是 end-to-end 測試，但可用於系統整合測試 (SIT) 環境。如需詳細資訊，請參閱 Service Catalog Puppet 文件中的 [使用政策模擬](#) 和 [套用服務控制政策](#)。

成熟度、完整性和支援

雖然 Service Catalog Puppet 並未正式支援 AWS 服務，但已廣泛採用。過去幾年，大型組織已使用此工具，在所需的佈建時段內成功且集中地將產品佈建至數百個 OU 帳戶。它已證實可大規模提供容錯產品佈建。遇到 Service Catalog Puppet 問題的使用者可以將他們記錄在 [GitHub 儲存庫](#) 中，供此 AWS 實驗室解決方案的貢獻者解決。

Service Catalog 工廠

Service Catalog Factory 是由 AWS Labs 提供的另一個工具。它類似於 AWS Control Tower- 它會產生帳戶並呼叫 Service Catalog (可能透過 Puppet) 來在這些帳戶中佈建 IaP。它使用許多與 Service Catalog Puppet 相同的機制來實作其功能。Service Catalog Factory 可以呼叫 Service Catalog 或 Service Catalog Puppet，為帳戶中的產品佈建基礎設施。此工具也支援在多個 AWS 區域和組織中產生帳戶。如需詳細資訊，請參閱 Service Catalog Factory [文件](#) 和 [GitHub 儲存庫](#)。

摘要和後續步驟

Service Catalog 可協助您快速且可靠地將基礎設施佈建為產品。您可以從定義的產品目錄中自助服務基礎設施，或將產品推送到hub-and-spoke模型中指定的目標帳戶。您可以使用 CloudFormation 指令碼或使用 來定義 Service Catalog 產品及其佈建範本 AWS CDK。在這兩種方法中，Service Catalog 透過呼叫 CloudFormation 來佈建產品，以部署代表產品佈建範本的堆疊。堆疊會部署到 CloudFormation 堆疊集中所有指定的目標帳戶。

Service Catalog 開發 AWS CDK 方法支援比 CloudFormation 更大的模組化和重複使用，因為您可以透過使用預先定義的 Service Catalog 產品和產品組合類別以及預先定義的資源類型來定義 產品和其資源。AWS CDK 實作需要更進階的程式設計技能。如果您的組織想要使用標準化資源組態和行為建立自己的可重複使用產品架構，作為 AWS 基礎設施開發的基礎，這可能會是合理的。

您可以使用 Service Catalog Puppet 和 Service Catalog Factory 來增強 Service Catalog 功能，主要用於佈建。Service Catalog Puppet 具有宣告式和標籤型產品佈建規格；內建、可自訂、高效能和專門建置的佈建工作流程；以及內建、可自訂、動作型 CI/CD 和 SDLC 管道。透過使用工作流程相依性和內建測試自動化功能，您可以以較低的操作風險鏈結 Service Catalog 產品。Service Catalog Puppet 可協助您在積極的時段內可靠地將產品佈建至數百個帳戶。Service Catalog Factory 類似 AWS Control Tower。它會產生帳戶並呼叫 Service Catalog，在這些帳戶中佈建 IaP。

Service Catalog 和服務目錄工具提供廣泛的功能，協助您管理 IaP AWS。Service Catalog 和這些工具正在進行持續改進。如需最新功能，請參閱 [AWS Service Catalog 功能](#)和[AWS Service Catalog 產品儲存庫](#)。

資源

參考

- [Service Catalog 文件](#)
- [Service Catalog API](#)
- [AppRegistry](#)
- [CloudFormation 文件](#)
- [CloudFormation 堆疊集](#)
- [AWS::ServiceCatalog::CloudFormationProduct 資源](#)
- [使用 Terraform 啟動 AWS Service Catalog 產品](#)
- [AWS Cloud Development Kit \(AWS CDK\)](#)
- [Service Catalog 建構程式庫](#)
- [AWS CDK ProductStack 類別](#)
- [AWS Organizations 文件](#)

工具

- [Service Catalog Puppet 文件](#)
- [Service Catalog Puppet GitHub 儲存庫](#)
- [Service Catalog Factory 文件](#)
- [Service Catalog Factory GitHub 儲存庫](#)

AWS 規範性指引模式

- [在多個 AWS 帳戶 和 中管理 AWS Service Catalog 產品 AWS 區域](#)
- [跨不同的 AWS 帳戶 和 複製 AWS Service Catalog 產品 AWS 區域](#)
- [使用 自動化 AWS Service Catalog 產品組合和產品部署 AWS CDK](#)

文件歷史紀錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 [RSS 摘要](#)。

變更	描述	日期
初次出版	—	2023 年 1 月 30 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。