管理员指南

AWS Service Catalog

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Service Catalog: 管理员指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 Service Catalog？ 1
视频：AWS Service Catalog 简介 1
概述 2
用户 2
产品 2
HashiCorp 支持 Terraform 开源和 Terraform Cloud 2
预配置产品 3
产品组合 3
版本控制 3
权限 3
约束 3
管理员初始工作流程 4
最终用户初始工作流程 4
配额 5
AWS Organizations 5
约束配额 5
产品组合配额 5
产品配额 6
预配置产品配额 6
区域配额 6
服务操作配额 6
TagOptions 配额 6
设置 7
..... 7
注册获取 AWS 账户 7
创建具有管理访问权限的用户 7
向管理员授予权限 9
向最终用户授予权限 11
安装和配置 Terraform 预置引擎 12
队列确定 12
将代理混淆添加到您的 Terraform 预置引擎中 13
入门 17
入门库 17
先决条件 18

了解更多 ... 18
通过 AWS CloudFormation 产品开始入门 ... 18
步骤 1：下载模板 ... 19
步骤 2：创建密钥对 ... 23
步骤 3：创建产品组合 ... 24
步骤 4：在产品组合中创建新产品 ... 24
步骤 5：添加模板约束 ... 25
步骤 6：添加启动约束 ... 26
步骤 7：向最终用户授予产品组合的访问权限 ... 28
步骤 8：测试最终用户体验 ... 29
开始使用 Terraform 产品 ... 30
更新为外部产品类型 ... 31
先决条件：配置您的 Terraform 预置引擎 ... 32
步骤 1：下载 Terraform 配置文件 ... 33
步骤 2：创建 Terraform 产品 ... 34
步骤 3：创建产品组合 ... 35
步骤 4：将产品添加至产品组合 ... 36
步骤 5：创建启动角色 ... 36
步骤 6：添加启动约束 ... 40
步骤 7：授予最终用户访问权限 ... 41
第 8 步：与最终用户共享产品组合 ... 42
步骤 9：测试最终用户体验 ... 42
步骤 10：监控 Terraform 预配置操作 ... 43
安全性 ... 45
数据保护 ... 45
利用加密来保护数据 ... 46
Identity and Access Management ... 47
受众 ... 47
基于身份的策略示例 AWS Service Catalog ... 47
AWS 托管策略 ... 52
使用服务相关角色 ... 60
故障排除 AWS Service Catalog 身份和访问权限 ... 65
控制访问权限 ... 67
日志记录和监控 ... 67
合规性验证 ... 67
弹性 ... 68

基础架构安全性 68
安全最佳实践 69
管理目录 70
管理产品组合 70
创建、查看和删除产品组合 71
查看产品组合详细信息 71
创建和删除产品组合 71
添加产品 72
添加约束 74
向用户授予访问权限 75
共享产品组合 76
共享和导入产品组合 83
管理产品 86
查看产品页面 86
创建产品 86
将产品添加到产品组合 89
更新产品 89
将产品与外部存储库中的模板文件进行同步 91
删除产品 98
管理版本 105
使用约束 106
启动约束 107
通知约束 112
标签更新约束 113
堆栈集约束 113
模板约束 114
使用服务操作 118
先决条件 118
步骤 1：配置最终用户权限 118
步骤 2：创建服务操作 120
步骤 3：将服务操作与产品版本关联 120
步骤 4：测试最终用户体验 121
步骤 5：使用 AWS CloudFormation 管理服务操作 121
步骤 6：问题排查 121
将 AWS Marketplace 产品添加到您的产品组合 123
使用 AWS Marketplace 管理 AWS Service Catalog 产品 124

手动管理和添加 AWS Marketplace 产品 124
使用 AWS CloudFormation StackSets 128
堆栈集和堆栈实例 129
堆栈集约束 129
管理预算 129
先决条件 130
创建预算 131
关联预算 132
查看预算 133
取消关联预算 133
管理预配置产品 134
以管理员的身份管理预配置产品 134
更改预配置产品的所有者 135
另请参阅 135
更新预配置产品的模板 135
教程：确定用户资源分配 136
管理 Terraform 开源产品状态错误 140
状态错误示例 140
管理 Terraform 开源产品状态文件 141
管理标签 143
AutoTags 143
TagOption 图书馆 144
使用发布产品 TagOptions 145
管理 TagOptions 149
TagOptions 与AWS Organizations标签策略一起使用 150
外部引擎 154
注意事项 155
参数解析 155
预置 157
Updating 160
正在终止 163
标记 164
监控 166
监控工具 166
自动化工具 166
CloudWatch 指标 167

启用 CloudWatch 指标 167
可用指标和维度 167
查看 AWS Service Catalog 指标 168
CloudTrail 日志 169
AWS Service Catalog信息在 CloudTrail 169
了解 AWS Service Catalog 日志文件条目 170
控制台品牌 172
AWS 区域 支持控制台品牌 172
文档历史记录 175
早期更新 176
..... clxxx

什么是 Service Catalog？

借助 Service Catalog，组织可以创建和管理获准在 AWS 上使用的 IT 服务的目录。这些 IT 服务可谓包罗万象，从虚拟机映像、服务器、软件和数据库，再到完整的多层应用程序架构等。

Service Catalog 允许组织集中管理通常部署的 IT 服务，并帮助组织实现一致的监管和满足合规性要求。最终用户可在遵循组织设定约束的情况下快速部署他们所需的已获得批准的 IT 服务。

Service Catalog 具有以下优势：

- 标准化

  管理员可以通过限制可启动产品的位置、可使用的实例类型以及多种其他配置选项，来管理已获批准的资产。这样可以为整个组织的产品预配置创建标准化的环境。
- 自助服务发现和启动

  用户浏览其有权访问的产品（服务或应用程序）的列表，找到要使用的产品并将其作为预配置产品自行启动。
- 访问权限的精细控制

  管理员从目录中将产品汇总为产品组合、添加预配置时要使用的约束和资源标签，然后通过 AWS Identity and Access Management (IAM) 用户和组授予对产品组合的访问权限。
- 扩展性和版本控制

  管理员可将产品添加到任意数量的产品组合并施加限制，无需创建另一个副本。将产品更新为新版本后，更新会传播到涵盖该产品的每个产品组合中的所有产品。

有关更多信息，请参阅 Service Catalog 产品详细信息页面。

作为使用 AWS Management Console 的替代方案，Service Catalog API 提供对所有最终用户操作的编程控制。有关更多信息，请参阅 Service Catalog 开发人员指南。

视频：AWS Service Catalog 简介

此视频 (7:27) 介绍了如何创建、组织和管理 AWS 的精选产品目录，以及如何共享具有权限级别的产品。因此，最终用户无需直接访问底层 AWS 服务即可快速预配置经批准的 IT 资源。

AWS Service Catalog 简介

Service Catalog 概述

开始使用 Service Catalog 之前，了解其组件及管理员和最终用户的初始工作流程会很有用。

用户

Service Catalog 支持以下类型的用户：

- 目录管理员（管理员）- 管理产品目录（应用程序和服务）、将产品组织到产品组合中并向最终用户授予访问权限。目录管理员准备 AWS CloudFormation 模板、配置约束并管理分配给产品的 IAM 角色，以提供高级资源管理。
- 最终用户 - 接收来自最终用户的 IT 部门或经理的 AWS 凭证，并使用 AWS Management Console 启动最终用户拥有其权限的产品。最终用户有时简称为用户，可被授予不同的权限，具体取决于您的操作要求。例如，用户可能拥有最高级别权限 (启动和管理其使用的产品所需的所有资源)，或仅拥有使用特定服务功能的权限。

产品

产品是指您希望可用于在 AWS 上进行部署的 IT 服务。产品包含一个或多个 AWS 资源，如 EC2 实例、存储卷、数据库、监控配置和网络组件，也可以是打包的 AWS Marketplace 产品。产品可以是运行 AWS Linux 的单个计算机实例，也可以是运行在自己环境中的完全配置的多层 Web 应用程序，或其中的任何内容。

您可以通过导入 AWS CloudFormation 模板创建产品。AWS CloudFormation 模板定义了产品所需的 AWS 资源、资源之间的关系，以及最终用户在启动产品来配置安全组、创建密钥对和执行其他自定义操作时可插入的参数。

HashiCorp 支持 Terraform 开源和 Terraform Cloud

AWS Service Catalog支持快速、自助式配置，并在其中管理您的 HashiCorp Terraform 开源和 Terraform Cloud 配置。AWS您可以将 Service Catalog 用作单一工具，在 AWS 中大规模组织、管理和分发 Terraform 配置。您可以使用 Service Catalog 的主要功能，包括对标准化和预先批准的 Terraform 模板进行编目、访问控制、最低权限配置、版本控制、标记以及与成千上万个 AWS 账户共享。您的最终用户会看到他们有权访问的产品和版本的简单列表，随后只需一个操作即可部署这些产品。

要了解更多信息并完成 Terraform 产品教程，请查看 开始使用 Terraform 产品。

预配置产品

通过允许您将产品实例作为一个单元进行预配置、标记、更新和终止，AWS CloudFormation 堆栈简化了对产品生命周期的管理。AWS CloudFormation 堆栈包含采用 JSON 或 YAML 格式的 AWS CloudFormation 模板及其关联的资源集合。预配置产品 是一个堆栈。当最终用户启动产品时，由 Service Catalog 预配置的产品实例是运行该产品所需的资源堆栈。有关更多信息，请参阅《AWS CloudFormation 用户指南》。

产品组合

产品组合是包含配置信息在内的产品集合。产品组合可帮助管理可使用特定产品的人员及其使用方式。利用 Service Catalog，您可以为组织内的每类用户创建一个自定义产品组合，并选择性授予对适当产品组合的访问权限。当您向产品组合添加新版本的产品时，该版本会自动供所有当前用户使用。

此外，您还可以与其他 AWS 账户共享产品组合，并允许这些账户的管理员对您的产品组合应用额外的约束（例如限制用户可以创建的 EC2 实例）。通过使用产品组合、权限、共享和约束，您可以确保用户所启动的产品经过正确配置，能够满足组织的需求并符合其标准。

版本控制

Service Catalog 允许您在目录中管理多个版本的产品。这种方法可以让您根据软件更新或配置变更来添加新版本的模板及关联的资源。

创建产品的新版本时，更新会自动分发到具有该产品访问权限的所有用户，允许用户选择要使用的产品版本。用户可以快速轻松地将产品的运行实例更新为新版本。

权限

向用户授予产品组合的访问权限，让用户能够浏览该产品组合并启动其中的产品。您可以应用 AWS Identity and Access Management (IAM) 权限来控制谁可以查看和修改您的目录。IAM 权限可以分配给 IAM 用户、组和角色。

当用户启动已分配有 IAM 角色的产品时，Service Catalog 将使用该角色通过 AWS CloudFormation 启动此产品的云资源。通过向每个产品分配一个 IAM 角色，您可以避免向用户授予执行未获批操作的权限，并使他们可以使用目录预配置资源。

约束

约束用于控制为某个产品部署特定 AWS 资源的方式。您可以使用约束对产品进行限制，以便进行管理或控制成本。存在不同的类型的 AWS Service Catalog 约束：启动约束、通知约束和模板约束。

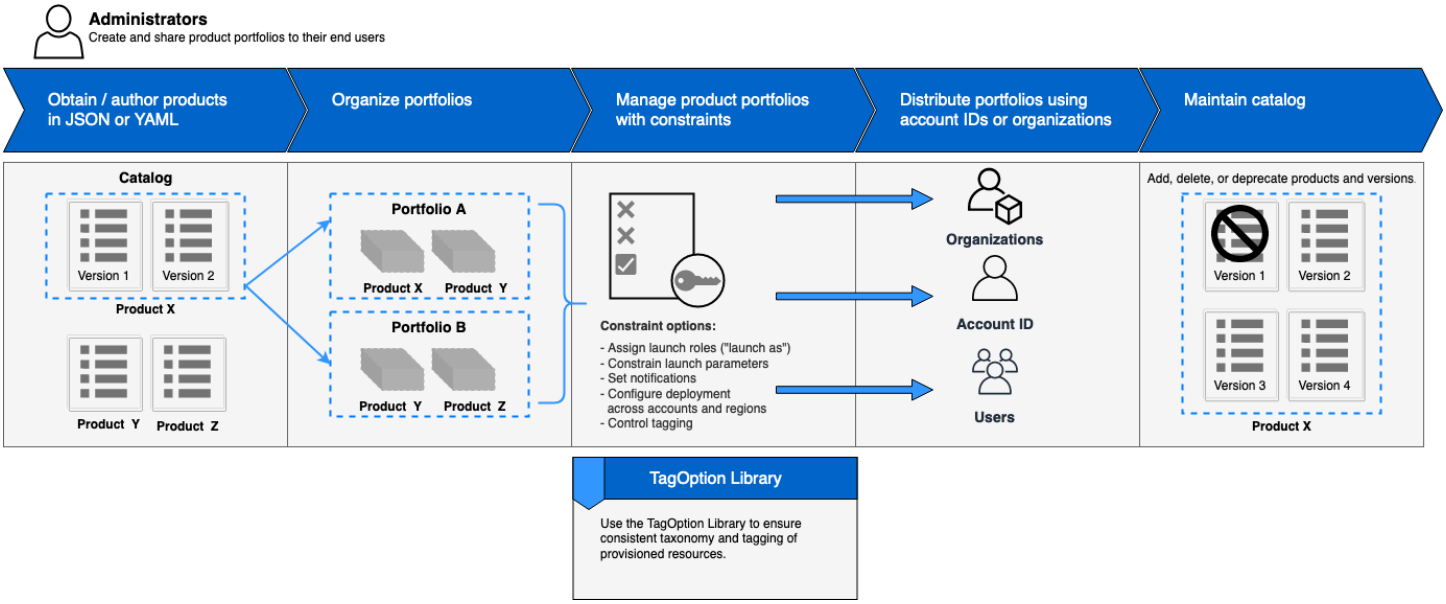
通过启动约束，您可为产品组合中的产品指定一个角色。使用此角色在启动时预配置资源，以便您可以限制用户权限，又不影响用户从目录预配置产品的能力。

通知约束使您能够使用 Amazon SNS 主题获取有关堆栈事件的通知。

模板约束用于限制用户在启动产品时可以使用的配置参数 (例如 EC2 实例类型或 IP 地址范围)。借助模板约束，您可以重复使用产品的常规 AWS CloudFormation 模板，并根据每个产品或每个产品组合对模板进行限制。

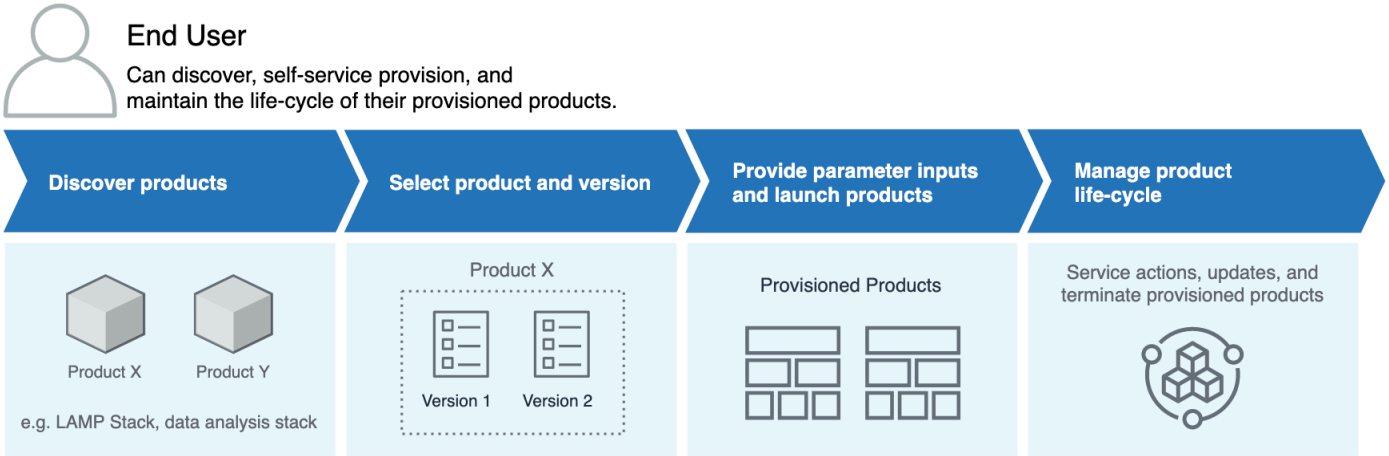
管理员初始工作流程

此图展示了管理员创建目录的初始工作流程。



最终用户初始工作流程

此图展示了最终用户的初始工作流程。



AWS Service Catalog 默认服务限额

您的AWS账户具有以下默认配额：约束AWS Organizations、产品组合、产品、预配置产品、区域、服务操作和 TagOptions。

您可以使用 Service Quotas 管理您的限额或请求增加配额。有关 Service Quotas 的更多信息，请参阅《Service Quotas 用户指南》中的什么是服务限额？。要了解如何请求提高配额，请参阅请求提高配额。

AWS Organizations

- 每个组织的 AWS Service Catalog 委托管理员数量：50

约束配额

- 每个产品组合的每个产品的约束数：100

产品组合配额

- 每个产品组合的用户、组和角色数：100
- 每个产品组合的产品数：150
- 每个产品组合的标签数：20
- 每个产品组合的共享账户数：5000
- 每个标签键的标签值：25

产品配额

- 每个产品的用户、组和角色数：200
- 每个产品的产品版本数：100
- 每个产品的标签数：20
- 每个标签键的标签值：25

预配置产品配额

- 每个预配置产品的标签数：50

区域配额

- 产品组合数：100
- 产品数：350

服务操作配额

- 每个区域的服务操作：200
- 每个产品版本的服务操作关联：25

TagOptions 配额

- TagOptions 每个资源：25
- 每个值为 TagOption：25

设置 AWS Service Catalog

在你开始之前 AWS Service Catalog，完成以下任务。

主题

- 注册获取 AWS 账户
- 创建具有管理访问权限的用户

注册获取 AWS 账户

如果你没有 AWS 账户，请完成以下步骤来创建一个。

报名参加 AWS 账户

1. 打开https://portal.aws.amazon.com/billing/注册。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

   当你注册时 AWS 账户，一个 AWS 账户根用户已创建。root 用户可以访问所有内容 AWS 服务 以及账户中的资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。您可以随时前往 https://aws.amazon.com/并选择“我的账户”，查看您当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

在你注册之后 AWS 账户，保护你的 AWS 账户根用户，启用 AWS IAM Identity Center，然后创建一个管理用户，这样你就不会使用 root 用户来执行日常任务。

保护你的 AWS 账户根用户

1. 登录 。AWS Management Console以账户所有者的身份选择 Root 用户并输入你的 AWS 账户 电子邮件地址。在下一页上，输入您的密码。

   有关使用 root 用户登录的帮助，请参阅中以 root 用户身份登录 AWS 登录 用户指南。

2. 为您的 root 用户开启多重身份验证 (MFA)。

   有关说明，请参阅为您的MFA设备启用虚拟设备 AWS 账户 用户指南中的 root IAM 用户（控制台）。

创建具有管理访问权限的用户

1. 启用 “IAM身份中心”。

   有关说明，请参阅启用 AWS IAM Identity Center中的 AWS IAM Identity Center 用户指南。

2. 在 IAM Identity Center 中，向用户授予管理访问权限。

   有关使用教程 IAM Identity Center 目录 作为您的身份来源，请参阅使用默认设置配置用户访问权限 IAM Identity Center 目录中的 AWS IAM Identity Center 用户指南。

以具有管理访问权限的用户身份登录

- 要使用您的 Ident IAM ity Center 用户登录URL，请使用您在创建 Ident IAM ity Center 用户时发送到您的电子邮件地址的登录信息。

  有关使用 Ident IAM ity Center 用户登录的帮助，请参阅登录 AWS 访问中的门户 AWS 登录 用户指南。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个遵循应用最低权限权限的最佳实践的权限集。

   有关说明，请参阅中的创建权限集 AWS IAM Identity Center 用户指南。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

   有关说明，请参阅中的添加群组 AWS IAM Identity Center 用户指南。

要提供访问权限，请为您的用户、组或角色添加权限：

- 中的用户和群组 AWS IAM Identity Center:

  创建权限集合。按照中创建权限集中的说明进行操作 AWS IAM Identity Center 用户指南。

- IAM通过身份提供商管理的用户：

创建适用于身份联合验证的角色。按照《IAM用户指南》中为第三方身份提供商创建角色（联合）中的说明进行操作。

- IAM用户：
  - 创建您的用户可以担任的角色。按照《用户指南》中为IAM用户创建角色中的IAM说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《用户指南》中向用户（控制台）添加权限中的IAM说明进行操作。

向 AWS Service Catalog 管理员授予权限

作为目录管理员，您需要 AWS Service Catalog 管理员控制台视图的访问权限以及允许您执行诸如以下任务的 IAM 权限：

- 创建和管理产品组合
- 创建和管理产品
- 添加模板约束以控制最终用户在启动产品时可用的选项
- 添加启动约束以定义最终用户在启动产品时 AWS Service Catalog 将担任的 IAM 角色
- 授予最终用户对产品的访问权限

您或者管理您 IAM 权限的管理员必须将完成本教程所需的策略附加到您的 IAM 用户、组或角色。

向目录管理员授予权限

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在导航窗格中，展开访问权限管理，然后选择角色。如果您已创建希望用作目录管理员的 IAM 用户，请选择该用户名，然后选择添加权限。否则，请创建一个用户，如下所示：
   a. 选择添加用户。
   b. 对于 User name，键入 **`ServiceCatalogAdmin`**。
   c. 选择 Programmatic access 和 AWS Management Console access。
   d. 选择下一步：权限。
3. 选择直接附上现有策略。
4. 选择创建策略，然后执行以下操作：
   a. 选择 JSON 选项卡。

b. 复制下面的示例策略，然后将其粘贴到策略文档 中：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateKeyPair",
                "iam:AddRoleToInstanceProfile",
                "iam:AddUserToGroup",
                "iam:AttachGroupPolicy",
                "iam:CreateAccessKey",
                "iam:CreateGroup",
                "iam:CreateInstanceProfile",
                "iam:CreateLoginProfile",
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:Get*",
                "iam:List*",
                "iam:PutRolePolicy",
                "iam:UpdateAssumeRolePolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

c. 选择下一步：标签。

d. （可选）选择添加标签以便将键值对与资源关联。您最多可添加 50 个标签。

> **Note**
>
> 标签是可以添加到资源的键值对。其可以用于识别、组织和搜索资源。有关更多信息，请参阅《AWS 一般参考 参考指南》中的标记 AWS 资源。

e. 选择下一步：审核。

f. 对于 Policy Name，键入 **`ServiceCatalogAdmin-AdditionalPermissions`**。

> **Important**
>
> 您必须向管理员授予 Amazon S3 权限，使其可以访问 AWS Service Catalog 存储在 Amazon S3 中的模板。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的用户策略示例。

g. 请选择创建策略。

5. 返回包含权限页面的浏览器窗口，然后选择 Refresh。
6. 在搜索字段中，键入 **ServiceCatalog** 以筛选策略列表。
7. 选择 **AWSServiceCatalogAdminFullAccess**、**ServiceCatalogAdmin-AdditionalPermissions** 策略的复选框，然后选择下一步：审核。
8. 如果您要更新用户，请选择 Add permissions。

   如果您要创建用户，请选择 Create user。您可以下载或复制凭证，然后选择 Close。
9. 要以目录管理员身份登录，请使用账户特定的 URL。要查找此 URL，请在导航窗格中选择 Dashboard，然后选择 Copy Link。将链接粘贴到您的浏览器中，然后使用您在此过程中创建或更新的 IAM 用户的名称和密码。

向 AWS Service Catalog 最终用户授予权限

您必须先向最终用户授予对 AWS Service Catalog 最终用户控制台视图的访问权限，他们才能使用 AWS Service Catalog。要授予访问权限，您需要将策略附加到最终用户所使用的 IAM 用户、用户组或角色。在以下过程中，我们将 **AWSServiceCatalogEndUserFullAccess** 策略附加到 IAM 用户组。

向最终用户组授予权限

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在导航窗格中，选择用户组。
3. 选择创建组，然后执行以下操作：

   a. 对于组名称，键入 **Endusers**。

   b. 在搜索字段中，键入 **AWSServiceCatalog** 以筛选策略列表。

   c. 选中该 **AWSServiceCatalogEndUserFullAccess** 策略的复选框。您也可以改而选择 **AWSServiceCatalogEndUserReadOnlyAccess**。

   d. 选择创建组。
4. 在导航窗格中，选择用户。
5. 选择添加用户，然后执行以下操作：
   a. 对于用户名，为用户键入一个名称。
   b. 选择密码 - AWS 管理控制台访问权限。
   c. 选择下一步：权限。
   d. 选择 Add user to group。
   e. 选中 Endusers 组的复选框，选择下一步：标签，然后选择下一步：审核。
   f. 在审核页面上，选择创建用户。下载或复制凭证，然后选择关闭。

安装和配置 Terraform 预置引擎

要成功通过 AWS Service Catalog 使用 Terraform 产品，您必须在管理 Terraform 产品的同一个账户中安装和配置 Terraform 预置引擎。首先，您可以使用 AWS 提供的 Terraform 预置引擎，该引擎安装和配置了 Terraform 预置引擎与 AWS Service Catalog 一同运作所需的代码和基础架构。此一次性设置大约需要 30 分钟。 AWS Service Catalog提供了一个 GitHub 存储库，其中包含有关安装和配置 Terraform 配置引擎的说明。

队列确定

当您调用预配置操作时，AWS Service Catalog 会准备一条负载消息以发送到预置引擎中的相关队列。为了为队列构建 ARN，AWS Service Catalog 会进行以下假设：

- 预置引擎位于产品所有者的账户中
- 预置引擎与对 AWS Service Catalog 的调用位于同一区域
- 预置引擎队列遵循记录的命名方案，详见下文

例如，如果使用账户 0000000000000 创建的产品`us-east-1`从账户 1111111111 调入，则假定 SQS ARN ProvisionProduct 是正确的。AWS Service Catalog `arn:aws:sqs:us-east-1:0000000000000:ServiceCatalogTerraformOSProvisionOperationQueue`

同样的逻辑也适用于由 `DescribeProvisioningParameters` 调用的 Lambda 函数。

将代理混淆添加到您的 Terraform 预置引擎中

端点上的代理混淆上下文密钥用于限制 **lambda:Invoke** 操作的访问权限

由 AWS Service Catalog 提供的引擎创建的参数解析器 Lambda 函数具有仅向 AWS Service Catalog 服务主体授予跨账户 `lambda:Invoke` 权限的访问策略：

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": "lambda:InvokeFunction",
          "Resource": "arn:aws:lambda:us-
east-1:account_id:function:ServiceCatalogTerraformOSParameterParser"
    }
  ]
}
```

这应该是 AWS Service Catalog 的集成正常运行所需的唯一权限。但是，您可以使用 `aws:SourceAccount` 代理混淆上下文密钥进一步对其限制。AWS Service Catalog 向这些队列发送消息时，AWS Service Catalog 会使用预配置账户的 ID 填充密钥。当您打算通过产品组合共享分发产品并希望确保只有特定账户使用您的引擎时，这一点很实用。

例如，您可以使用如下所示的条件将您的引擎限制为仅允许源自 000000000000 和 111111111111 的请求：

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": "lambda:InvokeFunction",
          "Resource": "arn:aws:lambda:us-
east-1:account_id:function:ServiceCatalogTerraformOSParameterParser",
          "Condition": {
```

```
            "StringLike": {
              "aws:SourceAccount": ["000000000000", "111111111111"]
        }
      }
    }
  ]
}
```

端点上的代理混淆上下文密钥用于限制 `sqs:SendMessage` 操作的访问权限

由 AWS Service Catalog 提供的引擎创建的预配置操作引入 Amazon SQS 队列的访问策略为仅向 AWS Service Catalog 服务主体授予跨账户 `sqs:SendMessage`（和关联的 KMS）权限：

```
{
      "Version": "2008-10-17",
      "Statement": [
        {
          "Sid": "Enable AWS Service Catalog to send messages to the queue",
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": "sqs:SendMessage",
          "Resource": [
            "arn:aws:sqs:us-
east-1:account_id:ServiceCatalogTerraformOSProvisionOperationQueue"
          ]
        },
        {
          "Sid": "Enable AWS Service Catalog encryption/decryption permissions when
 sending message to queue",
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": [
            "kms:DescribeKey",
            "kms:Decrypt",
            "kms:ReEncrypt",
            "kms:GenerateDataKey"
          ],
          "Resource": "arn:aws:kms:us-east-1:account_id:key/key_id"
    }
```

```
  ]
}
```

这应该是 AWS Service Catalog 的集成正常运行所需的唯一权限。但是，您可以使用 `aws:SourceAccount` 代理混淆上下文密钥进一步对其限制。AWS Service Catalog 向这些队列发送消息时，AWS Service Catalog 会使用预配置账户的 ID 填充密钥。当您打算通过产品组合共享分发产品并希望确保只有特定账户使用您的引擎时，这一点很实用。

例如，您可以使用如下所示的条件将您的引擎限制为仅允许源自 000000000000 和 111111111111 的请求：

```
{
      "Version": "2008-10-17",
      "Statement": [
      {
        "Sid": "Enable AWS Service Catalog to send messages to the queue",
        "Effect": "Allow",
        "Principal": {
          "Service": "servicecatalog.amazonaws.com"
        },
        "Action": "sqs:SendMessage",
        "Resource": [
          "arn:aws:sqs:us-
east-1:account_id:ServiceCatalogTerraformOSProvisionOperationQueue"
        ],
        "Condition": {
          "StringLike": {
            "aws:SourceAccount": ["000000000000", "111111111111"]
          }
        }
      },
      {
        "Sid": "Enable AWS Service Catalog encryption/decryption permissions when
 sending message to queue",
        "Effect": "Allow",
        "Principal": {
          "Service": "servicecatalog.amazonaws.com"
        },
        "Action": [
          "kms:DescribeKey",
          "kms:Decrypt",
          "kms:ReEncrypt",
          "kms:GenerateDataKey"
```

```
        ],
        "Resource": "arn:aws:kms:us-east-1:account_id:key/key_id"
    }
  ]
}
```

入门

您可以使用“入门库”中一个架构良好的产品模板开始使用 AWS Service Catalog，也可以按照入门教程中的步骤进行操作。

在本教程中，您将以目录管理员和最终用户的身份执行作业。作为目录管理员，您可以创建产品组合，然后创建产品。作为最终用户，您需验证自己是否可访问最终用户控制台并启动产品。产品可以是以下任一项：

- 一个在 Amazon Linux 上运行的云开发环境，基于 AWS CloudFormation 模板，该模板定义了产品可以使用的 AWS 资源。
- 一种在 Terraform 预置引擎上运行的开源环境，基于 tar.gz 配置文件，该文件定义了产品可以使用的 AWS 资源。

> **Note**
>
> 开始之前，请确保您已完成 设置 AWS Service Catalog 中的步骤。

主题

- 入门库
- 通过 AWS CloudFormation 产品开始入门
- 开始使用 Terraform 产品

入门库

AWS Service Catalog 提供了一个由架构完善的产品模板组成的入门库，因此您可以快速入门。您可以将我们的入门库产品组合中的任何产品复制到您自己的账户，然后根据您的需求进行自定义。

主题

- 先决条件
- 了解更多

先决条件

在使用入门库中的模板之前，请确保您具有以下内容：

- 使用 AWS CloudFormation 模板所需的权限。有关更多信息，请参阅使用 AWS Identity and Access Management 控制访问。
- 管理 AWS Service Catalog 所需的管理员权限。有关更多信息，请参阅 the section called “Identity and Access Management”。

了解更多

有关架构完善的框架的更多信息，请参阅 AWS 架构完善。

通过 AWS CloudFormation 产品开始入门

您可以使用“入门库”中一个架构良好的产品模板开始使用 AWS Service Catalog，也可以按照入门教程中的步骤进行操作。

在本教程中，您将以目录管理员和最终用户的身份执行作业。作为目录管理员，您需要创建产品组合，然后创建产品。作为最终用户，您需验证自己是否可访问最终用户控制台并启动产品。该产品是一个在 Amazon Linux 上运行的云开发环境，基于 AWS CloudFormation 模板，该模板定义了产品可以使用的 AWS 资源。

> **Note**
>
> 开始之前，请确保您已完成 设置 AWS Service Catalog 中的步骤。

主题

- 步骤 1：下载 AWS CloudFormation 模板
- 步骤 2：创建密钥对
- 步骤 3：创建 产品组合
- 步骤 4：在产品组合中创建新产品
- 步骤 5：添加模板约束以限制实例大小
- 步骤 6：添加启动约束以分配 IAM 角色
- 步骤 7：向最终用户授予产品组合的访问权限

- 步骤 8：测试最终用户体验

步骤 1：下载 AWS CloudFormation 模板

您可以使用 AWS CloudFormation 模板来配置和预配置产品组合和产品。模板是 JSON 或 YAML 格式的文本文件，描述了您希望预配置的资源。有关更多信息，请参阅 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-formats.html 用户指南 中的AWS CloudFormation模板格式。您可以使用 AWS CloudFormation 编辑器或自己选择的文本编辑器创建和保存模板。在本教程中，我们提供了一个简单模板来帮助您入门。此模板会启动为 SSH 访问而配置的单个 Linux 实例。

> **Note**
>
> 使用 AWS CloudFormation 模板需要特殊权限。在您开始之前，确保您拥有正确的权限。有关更多信息，请参阅 入门库 中的先决条件。

模板下载

为本教程提供的示例模板，`development-environment.template`，可在以下网址获得：https://awsdocs.s3.amazonaws.com/servicecatalog/development-environment.template。

模板概述

示例模板的文本如下所示：

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

  "Description" : "AWS Service Catalog sample template. Creates an Amazon EC2 instance
                    running the Amazon Linux AMI. The AMI is chosen based on the
 region
                    in which the stack is run. This example creates an EC2 security
                    group for the instance to give you SSH access. **WARNING** This
                    template creates an Amazon EC2 instance. You will be billed for
 the
                    AWS resources used if you create a stack from this template.",

  "Parameters" : {
    "KeyName": {
      "Description" : "Name of an existing EC2 key pair for SSH access to the EC2
 instance.",
```

```
      "Type": "AWS::EC2::KeyPair::KeyName"
    },

    "InstanceType" : {
      "Description" : "EC2 instance type.",
      "Type" : "String",
      "Default" : "t2.micro",
      "AllowedValues" : [ "t2.micro", "t2.small", "t2.medium", "m3.medium",
"m3.large",
        "m3.xlarge", "m3.2xlarge" ]
    },

    "SSHLocation" : {
      "Description" : "The IP address range that can SSH to the EC2 instance.",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "Must be a valid IP CIDR range of the form x.x.x.x/x."
   }
  },

  "Metadata" : {
    "AWS::CloudFormation::Interface" : {
      "ParameterGroups" : [{
        "Label" : {"default": "Instance configuration"},
        "Parameters" : ["InstanceType"]
      },{
        "Label" : {"default": "Security configuration"},
        "Parameters" : ["KeyName", "SSHLocation"]
      }],
      "ParameterLabels" : {
        "InstanceType": {"default": "Server size:"},
        "KeyName": {"default": "Key pair:"},
        "SSHLocation": {"default": "CIDR range:"}
      }
    }
  },

  "Mappings" : {
    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "HVM64" : "ami-08842d60" },
      "us-west-2"      : { "HVM64" : "ami-8786c6b7" },
```

```
      "us-west-1"      : { "HVM64" : "ami-cfa8a18a" },
      "eu-west-1"      : { "HVM64" : "ami-748e2903" },
      "ap-southeast-1" : { "HVM64" : "ami-d6e1c584" },
      "ap-northeast-1" : { "HVM64" : "ami-35072834" },
      "ap-southeast-2" : { "HVM64" : "ami-fd4724c7" },
      "sa-east-1"      : { "HVM64" : "ami-956cc688" },
      "cn-north-1"     : { "HVM64" : "ami-ac57c595" },
      "eu-central-1"   : { "HVM64" : "ami-b43503a9" }
    }

  },

  "Resources" : {
    "EC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "InstanceType" : { "Ref" : "InstanceType" },
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" }, "HVM64" ] }
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : { "Ref" : "SSHLocation"}
        } ]
      }
    }
  },

  "Outputs" : {
    "PublicDNSName" : {
      "Description" : "Public DNS name of the new EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "EC2Instance", "PublicDnsName" ] }
    },
    "PublicIPAddress" : {
```

```
      "Description" : "Public IP address of the new EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "EC2Instance", "PublicIp" ] }
    }
  }
}
```

模板资源

模板声明在启动产品时将创建的资源。它包含以下部分：

- AWSTemplateFormatVersion（可选）-用于创建此AWS模板的模板格式的版本。最新的模板格式版本是 2010-09-09，并且它是目前唯一的有效值。
- 描述（可选）- 模板的描述。
- 参数（可选）- 用户必须指定的用于启动产品的参数。对于每个参数，模板包含一个说明，还有键入的值必须满足的约束。有关约束的更多信息，请参阅 使用 AWS Service Catalog 约束。

  使用 `KeyName` 参数可以指定 Amazon Elastic Compute Cloud (Amazon EC2) 密钥对名称，最终用户使用 AWS Service Catalog 启动您的产品时必须提供该名称。您将在接下来的步骤中创建密钥对。
- 元数据（可选）- 提供有关模板的其他信息的对象。AWS::CloudFormation: Interfac e 键定义了最终用户控制台视图如何显示参数。`ParameterGroups` 属性定义如何对参数分组以及这些组的标题。`ParameterLabels` 属性定义容易记住的参数名称。当用户指定参数来启动基于此模板的产品时，最终用户控制台视图在标题 `Server size:` 下显示标记为 `Instance configuration` 的参数，并在标题 `Key pair:` 下显示标记为 `CIDR range:` 和 `Security configuration` 的参数。
- 映射（可选）- 可用来指定条件参数值的密钥和关键值的映射，与查找表类似。您可以使用 “资源” 和 “输出” 部分中的 Fn:: FindInMap 内部函数将键与相应的值进行匹配。上述模板包含了对应每项的 AWS 区域和亚马逊机器映像（AMI）列表。AWS Service Catalog 使用此映射根据用户在 AWS Management Console 中选择的 AWS 区域确定要使用的 AMI。
- 资源（必需）- 堆栈资源及其属性。您可引用模板的资源和输出部分中的资源。在上述模板中，我们指定了一个运行 Amazon Linux 的 EC2 实例和一个允许 SSH 对该实例访问权限的安全组。EC2 实例资源的属性部分使用用户键入的信息来配置实例类型和 SSH 访问的密钥名称。

  AWS CloudFormation 使用当前 AWS 区域从之前定义的映射中选择 AMI ID 并向其分配安全组。安全组已配置为允许端口 22 上来自用户指定的 CIDR IP 地址范围的入站访问。
- 输出（可选）- 告知用户产品启动完成的时间的文本。提供的模板获得已启动实例的公有 DNS 名称并将其显示给用户。用户需要此 DNS 名称来使用 SSH 连接到实例。

有关模板剖析页面的更多信息，请参阅《AWS CloudFormation 用户指南》中的模板参考。

步骤 2：创建密钥对

要允许您的最终用户启动基于本教程示例模板的产品，您必须创建 Amazon EC2 密钥对。密钥对是用于加密数据的公有密钥与用于解密数据的私有密钥的组合。有关密钥对的更多信息，请确保您已登录 AWS 控制台，然后在 Amazon EC2 用户指南中查看 Amazon EC2 密钥对。

本教程的 AWS CloudFormation 模板包含以下`KeyName`参数：`development-environment.template`

```
. . .
  "Parameters" : {
    "KeyName": {
      "Description" : "Name of an existing EC2 key pair for SSH access to the EC2
 instance.",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
. . .
```

最终用户在使用 AWS Service Catalog 密钥对启动基于模板的产品时必须指定密钥对的名称。

如果您的账户中已有一个您希望使用的密钥对，则可以跳至 步骤 3：创建 产品组合。否则，请完成以下步骤。

创建密钥对

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中的 Network & Security 下，选择 Key Pairs。
3. 在 Key Pairs 页上，选择 Create Key Pair。
4. 对于 Key pair name，键入易于记住的名称，然后选择 Create。
5. 当控制台提醒您保存私有密钥文件时，请将该文件保存到安全位置。

> **Important**
>
> 这是您保存私有密钥文件的唯一机会。

步骤 3：创建 产品组合

要为用户提供产品，首先请为这些产品创建产品组合。

创建产品组合

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧的导航窗格中，选择产品组合，然后选择创建产品组合。
3. 键入以下值：
   - 产品组合名称 – **`Engineering Tools`**
   - 产品组合描述 — **`Sample portfolio that contains a single product.`**
   - 拥有者 – **`IT (it@example.com)`**
4. 选择创建。

步骤 4：在产品组合中创建新产品

创建产品组合后，您就可以在产品组合中创建产品了。在本教程中，您将创建名为 Linux Desktop 的产品，该产品是在 Amazon Linux 上运行的云开发环境，位于工程工具产品组合内。

在产品组合中创建产品

1. 如果您已完成上一步骤，则将显示 Portfolios 页面。否则，打开 https://console.aws.amazon.com/servicecatalog/。
2. 选择并打开您在步骤 2 中创建的工程工具产品组合。
3. 选择上传新产品。
4. 在产品详细信息部分的创建产品页面上，输入以下内容：
   - 产品名称 – **`Linux Desktop`**
   - 产品描述 – **`Cloud development environment configured for engineering staff. Runs AWS Linux.`**
   - 拥有者 – **`IT`**
   - 分销商 –（空白）
5. 在版本详细信息页面上，选择使用 CloudFormation 模板。然后选择指定 Amazon S3 模板 URL 并输入以下内容：

- 选择模板 – **`https://awsdocs.s3.amazonaws.com/servicecatalog/development-environment.template`**
- 版本标题 – **`v1.0`**
- 描述 – **`Base Version`**

6. 在支持详细信息部分，输入以下内容：
   - 联系电子邮件 – **`ITSupport@example.com`**
   - 支持链接 – **`https://wiki.example.com/IT/support`**
   - 支持描述 – **`Contact the IT department for issues deploying or connecting to this product.`**
7. 选择创建产品。

步骤 5：添加模板约束以限制实例大小

约束在产品组合级别添加对产品的另一层控制。约束可以控制产品的启动上下文（启动约束），也可以将规则添加到 AWS CloudFormation 模板（模板约束）。有关更多信息，请参阅 使用 AWS Service Catalog 约束。

将模板约束添加到 Linux Desktop 产品可阻止用户在启动时选择大型实例类型。利用开发环境模板，用户能够从 6 个实例类型中进行选择；此约束会将有效的实例类型限制为两个最小的类型：`t2.micro` 和 `t2.small`。有关更多信息，请参阅 Amazon EC2 用户指南中的 T 2 实例。

将模板约束添加到 Linux Desktop 产品

1. 在产品组合详细信息页面上，选择约束，然后选择创建约束。
2. 在创建约束页面中，对于产品项，选择 Linux Desktop。然后，对于约束类型项，选择模板。
3. 在模板约束部分，选择文本编辑器。
4. 将以下内容粘贴到文本编辑器中：

```
{
  "Rules": {
    "Rule1": {
      "Assertions": [
        {
          "Assert" : {"Fn::Contains": [["t2.micro", "t2.small"], {"Ref":
 "InstanceType"}]},
```

```
            "AssertDescription": "Instance type should be t2.micro or t2.small"
          }
        ]
      }
    }
}
```

5. 在约束描述中，输入 **`Small instance sizes`**。
6. 选择创建。

步骤 6：添加启动约束以分配 IAM 角色

启动约束指定 IAM 角色，最终用户启动产品时 AWS Service Catalog 将担任此角色。

在此步骤中，您添加启动约束到 Linux Desktop 产品，以便 AWS Service Catalog 可使用作为产品 AWS CloudFormation 模板一部分的 IAM 资源。

您向产品分配的作为启动约束的 IAM 角色必须具有以下权限

1. AWS CloudFormation
2. 产品 AWS CloudFormation 模板中的服务
3. 自服务拥有的 Amazon S3 存储桶中读取 AWS CloudFormation 模板的访问权限。

此启动约束让最终用户可以启动产品，并在启动之后将其作为预配置产品进行管理。有关更多信息，请参阅 AWS Service Catalog 启动约束。

没有启动约束时，您需要先将额外的 IAM 权限授予您的最终用户，他们才能使用 Linux Desktop 产品。例如，`ServiceCatalogEndUserAccess` 策略仅授予访问 AWS Service Catalog 最终用户控制台视图所需的最低 IAM 权限。

借助使用启动约束，您可以遵循 IAM 最佳实践，将最终用户 IAM 权限保持在最低限度。有关更多信息，请参阅《IAM 用户指南》中的授予最低权限。

添加启动约束

1. 遵照 IAM 用户指南中在 JSON 选项卡上创建新策略的说明。
2. 在 JSON 策略文档中，粘贴以下内容：
   - `cloudformation`— 允许 AWS Service Catalog 创建、读取、更新、删除、列出和标记 AWS CloudFormation 堆栈的完全权限。

- `ec2`— 允许 AWS Service Catalog 列出、读取、写入、预配置和标记作为 AWS Service Catalog 产品一部分的 Amazon Elastic Compute Cloud (Amazon EC2) 资源的全部权限。根据您要部署的 AWS 资源，此权限可能会发生更改。
- `ec2`— 为您的 AWS 账户创建新的托管策略，并将指定的托管策略附加到指定的 IAM 角色。
- `s3`— 允许对 AWS Service Catalog 拥有的 Amazon S3 存储桶的访问权限。要部署产品，AWS Service Catalog 需要预配置构件的访问权限。
- `servicecatalog`— 允许 AWS Service Catalog 拥有代表最终用户列出、读取、写入、标记和启动资源的权限。
- `sns`— 允许 AWS Service Catalog 拥有列出、读取、写入和标记启动约束 Amazon SNS 主题的权限。

> **Note**
>
> 根据您要部署的基础资源，您可能需要修改示例 JSON 策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStacks",
                "cloudformation:GetTemplateSummary",
                "cloudformation:SetStackPolicy",
                "cloudformation:ValidateTemplate",
                "cloudformation:UpdateStack",
                "ec2:*",
                "servicecatalog:*",
                "sns:*"
            ],
            "Resource": "*"
        },
        {
         "Effect":"Allow",
```

```
            "Action":[
               "s3:GetObject"
            ],
            "Resource":"*",
            "Condition":{
               "StringEquals":{
                  "s3:ExistingObjectTag/servicecatalog:provisioning":"true"
               }
            }
         }
      ]
}
```

3. 请选择下一步，标签。
4. 选择下一步，审核。
5. 在查看策略页面上，输入 **linuxDesktopPolicy** 作为名称。
6. 选择创建策略。
7. 在导航窗格中，选择角色。然后选择创建角色并执行以下操作：
   a. 对于选择可信实体，选择 AWS 服务，然后在其他 AWS 服务的用例下选择 Service Catalog。选择 Service Catalog 用例，然后选择下一步。
   b. 搜索linuxDesktopPolicy策略，然后选中该复选框。
   c. 请选择 Next（下一步）。
   d. 对于角色名称，键入 **linuxDesktopLaunchRole**。
   e. 选择创建角色。
8. 打开AWS Service Catalog控制台，网址为 https://console.aws.amazon.com/servicecatalog。
9. 选择 Engineering Tools 产品组合。
10. 在产品组合详细信息页面上，选择约束选项卡，然后选择创建约束。
11. 对于产品，选择 Linux Desktop，对于约束类型，选择启动。
12. 请选择选择 IAM 角色。接下来选择 “linuxDesktopLaunch角色”，然后选择 “创建”。

步骤 7：向最终用户授予产品组合的访问权限

现在，您已创建产品组合并添加产品，可以向最终用户授予访问权限。

先决条件

如果您还没有为最终用户创建 IAM 群组，请参阅 向 AWS Service Catalog 最终用户授予权限。

提供对产品组合的访问权限

1. 在产品组合详细信息页面上，选择访问权限选项卡。
2. 选择授予访问权限。
3. 在群组选项卡上，选中最终用户的 IAM 群组复选框。
4. 选择添加访问权限。

步骤 8：测试最终用户体验

要验证最终用户是否可以成功访问最终用户控制台视图并启动您的产品，请以最终用户的身份登录 AWS 并执行这些任务。

验证最终用户可以访问最终用户控制台

1. 遵照《IAM 用户指南》中的以 IAM 用户身份登录的说明进行操作。
2. 在菜单栏中，选择您在其中创建了 `Engineering Tools` 产品组合的 AWS 区域。在本教程中，选择 us-east-1 区域。
3. 打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/ 以查看：
   - 产品 - 用户可使用的产品。
   - 预配置产品 - 用户已启动的预配置产品。

要验证最终用户可以启动 Linux Desktop 产品

请注意，在本教程中，选择 us-east-1 区域。

1. 在控制台的产品部分，选择 Linux Desktop。
2. 选择启动产品，启动配置产品的向导。
3. 在启动：Linux Desktop 页面上，输入 **`Linux-Desktop`** 用作预配置产品名称。
4. 在参数页面上，输入以下内容，然后选择下一步：
   - 服务器大小 - 选择 **`t2.micro`**。
   - 密钥对 - 选择您在 步骤 2：创建密钥对 中创建的密钥对。

- CIDR 范围 - 输入 IP 地址的有效 CIDR 范围以连接到实例。您可以使用默认值 (0.0.0.0/0) 以允许从任何 IP 地址进行访问，也可以为后跟 **/32** 的 IP 地址以仅允许您的 IP 地址访问，或者为二者之间的某个值。

5. 选择启动产品以启动堆栈。控制台显示 Linux-Desktop 堆栈的堆栈详细信息页面。产品的初始状态是更改中。AWS Service Catalog 需要花费几分钟时间来启动产品。要查看当前状态，请刷新您的浏览器。产品启动之后，状态将为可用。

开始使用 Terraform 产品

AWS Service Catalog支持快速、自助式配置，并在其中管理您的 HashiCorp Terraform 配置。AWS 您可以将 AWS Service Catalog 用作单一工具，在 AWS 中大规模组织、管理和分发 Terraform 配置。AWS Service Catalog 支持 Terraform 的多个关键功能，包括对标准化和预先批准的 Terraform 模板进行编目、访问控制、版本控制、标记以及与其他 AWS 账户共享。在 AWS Service Catalog 中，您的最终用户会看到他们有权访问的产品和版本的简单列表，随后只需一个操作即可部署这些产品。

> **Note**
>
> 为了继续支持 HashiCorp 技术，由于最近对Terraform的许可变更，将以前对Terraform 开源的任何提法AWS Service Catalog更改为外部提法。外部产品类型包括对 Terraform 社区版（以前称为 Terraform 开源）的支持。有关将现有 Terraform 开源产品和预配置产品迁移到外部产品类型的更多信息和说明，请查看 将现有的 Terraform 开源产品和预配置产品更新为外部产品类型。

以下教程中的步骤将帮助您在 AWS Service Catalog 中开始使用 Terraform 产品。

作为目录管理员，您使用中央管理员账户（中心账户）工作。Terraform 社区版和 Terraform 云产品都需要 Terraform 预置引擎，您可以在 Terraform 社区版的预置引擎（外部产品类型） 和 Terraform 云的预置引擎 中了解更多信息。

在本教程中，请使用管理员账户执行以下任务：

- 使用 Terraform 云或外部产品类型创建 Terraform 产品。Service Catalog 使用“外部”产品类型来支持 Terraform 社区版产品。
- 将产品与产品组合关联
- 创建启动限制以允许您的最终用户预配置产品
- 为产品添加标签

- 与最终用户账户（分支账户）共享产品组合和 Terraform 产品

在本教程中，您将在管理员中心账户中使用组织共享选项共享产品组合，该账户也是组织的管理账户。有关组织共享的更多信息，请参阅 共享产品组合。

您在本教程中创建的 Terraform 产品中包含的 AWS 资源是一个简单的 Amazon S3 桶。

> **Note**
>
> 开始之前，请确保您已完成 设置 AWS Service Catalog 中的步骤。

主题

- 将现有的 Terraform 开源产品和预配置产品更新为外部产品类型
- 先决条件：配置您的 Terraform 预置引擎
- 步骤 1：下载 Terraform 配置文件
- 步骤 2：创建 Terraform 产品
- 步骤 3：创建 AWS Service Catalog 产品组合
- 步骤 4：将产品添加至产品组合
- 步骤 5：创建启动角色
- 步骤 6：为 Terraform 产品添加启动约束
- 步骤 7：授予最终用户访问权限
- 第 8 步：与最终用户共享产品组合
- 步骤 9：测试最终用户体验
- 步骤 10：监控 Terraform 预配置操作

将现有的 Terraform 开源产品和预配置产品更新为外部产品类型

为了继续支持 HashiCorp 技术，由于最近对Terraform的许可变更，将以前对Terraform 开源的任何提法AWS Service Catalog更改为外部提法。“外部”产品类型包括对 Terraform 社区版（以前称为 Terraform 开源）的支持。AWS Service Catalog 不再支持 Terraform 开源作为任何新产品或预置产品的有效产品类型。您只能更新或终止现有 Terraform 开源资源，包括产品版本和预置产品。

如果尚未这样做，则必须按照本节中的说明将所有现有 Terraform 开源产品和预置产品过渡到外部产品。

1. 更新现有的 AWS Service Catalog Terraform 参考引擎，使其包含对外部和 Terraform 开源产品类型的支持。有关更新 Terraform 参考引擎的说明，请查看我们的GitHub 存储库。
2. 使用新的外部产品类型重新创建任何现有的 Terraform 开源产品。
3. 删除所有使用 Terraform 开源产品类型的现有产品。
4. 重新预置剩余资源以使用新的“外部”产品类型。
5. 终止所有使用 Terraform 开源产品类型的现有预配置产品。

过渡现有产品后，对于任何使用 tar.gz 配置文件的新产品，请使用“外部”产品类型。

AWS Service Catalog 将根据需要支持客户完成此更改。如果这些更改需要您的账户付出大量精力或影响关键产品工作负载，请联系您的客户代表请求帮助。

先决条件：配置您的 Terraform 预置引擎

作为在 AWS Service Catalog 中创建 Terraform 产品的先决条件，您必须在您的 Service Catalog 管理员帐户（中心账户）中安装和配置预置引擎。Terraform 社区版产品（使用外部产品类型）和 Terraform 云产品（使用 Terraform 云产品类型）都需要预置引擎。

> **Note**
>
> 引擎配置为一次性设置，大约需要 30 分钟。

Terraform 社区版的预置引擎（外部产品类型）

AWS Service Catalog 使用外部产品类型来支持 Terraform 社区版产品。外部产品类型还支持其他预置工具，包括 Pulumi、Ansible、Chef 等，具体取决于预置引擎的配置。

对于使用 “外部” 产品类型和 “Terraform 社区版” HashiCorp 的AWS Service Catalog产品，您必须在 AWS Service Catalog管理员帐户（中心帐户）中安装和配置 Terraform 配置引擎。 AWS管理此引擎及其资源。

AWS Service Catalog提供了一个 GitHub 存储库，其中包含有关安装和配置AWS提供的 Terraform 配置引擎的说明。存储库包含以下信息：

- 必需的安装工具
- 构建代码
- 部署到 AWS 账户

- 有关预配置工作流程、质量保证和限制的其他信息

Terraform 云的预置引擎

对于使用 Terraform Cloud 产品类型和 Terraform Cloud HashiCorp 的产品，您必须在AWS Service Catalog管理员帐户（AWS Service Catalog中心帐户）中安装和配置 Terraform 配置引擎。 HashiCorp 在远程环境中管理此引擎。

HashiCorp 提供了一个 GitHub 存储库，其中包含有关为其配置 Terraform Cloud 引擎的说明。AWS Service Catalog存储库包含以下信息：

- 必需的安装工具
- 构建代码
- 部署到 AWS 账户
- 有关预配置工作流程、质量保证和限制的其他信息

步骤 1：下载 Terraform 配置文件

你可以使用 Terraform 配置文件来创建和配置 HashiCorp Terraform 产品。这些配置文件为纯文本文件，描述了要预配置的资源。您可以使用自己选择的文本编辑器来创建、更新和保存配置。要创建产品，您必须将 Terraform 配置以 tar.gz 文件形式上传。在本教程中，AWS Service Catalog 提供了一个简单的配置文件，以便您快速上手。配置文件在 Amazon S3 控制台中创建一个桶。

下载配置文件

AWS Service Catalog 提供了一个示例 `simple-s3-bucket.tar.gz` 配置文件供您在本教程中使用。

配置文件概述

示例配置的文本如下：

```
variable "bucket_name" {
  type = string
}
provider "aws" {
}
```

```
resource "aws_s3_bucket" "bucket" {
  bucket = var.bucket_name
}
output regional_domain_name {
  value = aws_s3_bucket.bucket.bucket_regional_domain_name
}
```

配置资源

配置文件声明 AWS Service Catalog 预配置产品时要创建的资源。它包含以下部分：

- 变量（可选）-管理员用户（中心账户管理员）可以分配的，用于自定义配置的值定义。变量为更改给定配置的行为提供了一致接口。变量关键字之后的标签是变量名称，该名称在同一模块的所有变量中必须是唯一的。此名称用于为变量分配外部值和从模块内部引用变量值。
- 提供商（可选）- 用于资源预配置的云服务提供商，即AWS。 AWS Service Catalog 仅支持 AWS 作为提供商。因此，Terraform 预置引擎会覆盖任何其他列出的提供商为 AWS。
- 资源（必需）- 用于配置的 AWS 基础架构资源。在本教程中，Terraform 配置文件指定为 Amazon S3。
- 输出（可选）- 返回的信息或值，类似于编程语言中的返回值。您可使用输出数据，通过自动化工具来配置基础设施工作流程。

步骤 2：创建 Terraform 产品

安装 Terraform 配置引擎后，你就可以在中创建 HashiCorp Terraform 产品了。AWS Service Catalog 在本教程中，您将创建包含简单 Amazon S3 存储桶的 Terraform 产品。

创建 Terraform 产品

1. 通过以下网址打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/，然后以管理用户身份登录。
2. 导航到管理部分，然后选择产品列表。
3. 选择创建产品。
4. 在“产品详细信息”部分的创建产品页面上，选择外部或 Terraform 云产品类型。Service Catalog 使用外部产品类型来支持 Terraform 社区版产品。
5. 输入以下产品详细信息：
   - 产品名称 – **`Simple S3 bucket`**

- 产品描述 - 包含 Amazon S3 桶的 Terraform 产品。
- 拥有者 – **IT**
- 分销商 –（空白）

6. 在版本详细信息页面上，依次选择上传模板文件和选择文件。在 步骤 1：下载 Terraform 配置文件 中选择您下载的文件。
7. 输入以下信息：
   - 版本名称 – **v1.0**
   - 版本描述 – **Base Version**
8. 在支持详细信息部分，输入以下内容，然后选择创建产品。
   - 联系电子邮件 – **ITSupport@example.com**
   - 支持链接 – **https://wiki.example.com/IT/support**
   - 支持描述 – **Contact the IT department for issues deploying or connecting to this product.**
9. 选择创建产品。

成功创建产品后，AWS Service Catalog 将在产品页面上显示确认横幅。

步骤 3：创建 AWS Service Catalog 产品组合

您可以在 AWS Service Catalog 管理员账户（中心账户）中创建产品组合，以便产品组织和分发给最终用户账户（分支账户）。

创建产品组合

1. 通过以下网址打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/，然后以管理员身份登录。
2. 在左侧的导航窗格中，选择产品组合，然后选择创建产品组合。
3. 输入以下值：
   - 产品组合名称 – **S3 bucket**
   - 产品组合描述 — **Sample portfolio for Terraform configurations.**
   - 拥有者 – **IT (it@example.com)**
4. 选择创建。

步骤 4：将产品添加至产品组合

创建投资组合后，您可以添加在步骤 2 中创建的 HashiCorp Terraform 产品。

要将产品添加到产品组合

1. 导航产品列表页面。
2. 选择您在步骤 2 中创建的简单 S3 桶 Terraform 产品，然后选择操作。从下拉菜单中选择添加产品到产品组合。AWS Service Catalog 显示了向投资组合添加简单 S3 存储桶窗格。
3. 选择 S3 存储桶产品组合，然后关闭创建启动约束。本教程的后面部分中您将创建启动约束。
4. 选择将产品添加到产品组合。

成功将产品添加到产品组合后，AWS Service Catalog 将在产品列表页面上显示确认横幅。

步骤 5：创建启动角色

在此步骤中，您将创建一个 IAM 角色（启动角色），指定 Terraform 配置引擎在最终用户启动 Terra HashiCorp form 产品时AWS Service Catalog可以承担的权限。

您稍后分配给简单 Amazon S3 桶 Terraform 产品作为启动约束的 IAM 角色（启动角色）必须具有以下权限：

- 您的 Terraform 产品基础 AWS 资源的访问权限。在本教程中，这包括对 `s3:CreateBucket*`、`s3:DeleteBucket*`、`s3:Get*`、`s3:List*` 和 `s3:PutBucketTagging` Amazon S3 操作的访问权限。
- 自 AWS Service Catalog 拥有的 Amazon S3 存储桶中读取 Amazon S3 模板的访问权限
- 对 `CreateGroup`、`ListGroupResources`、`DeleteGroup` 和 `Tag` 资源组操作的访问权限。借助这些操作，AWS Service Catalog 能够管理资源组和标签

要在 AWS Service Catalog 管理员账户中创建启动角色

1. 登录 AWS Service Catalog 管理员账户后，遵照 IAM 用户指南中在 JSON 选项卡上创建新策略的说明进行操作。
2. 为您的简单 Amazon S3 存储桶 Terraform 产品创建策略。此策略必须在创建启动角色之前创建，并且包含以下权限：

- `s3`— 允许 AWS Service Catalog 列出、读取、写入、预配置和标记 Amazon S3 产品的完全权限。
- `s3`— 允许对 AWS Service Catalog 拥有的 Amazon S3 存储桶的访问权限。要部署产品，AWS Service Catalog 需要预配置构件的访问权限。
- `resourcegroups`— 允许 AWS Service Catalog 创建、列出、删除和标记 AWS Resource Groups。
- `tag`— 允许 AWS Service Catalog 的标记权限。

> **Note**
>
> 根据您要部署的基础资源，您可能需要修改示例 JSON 策略。

在 JSON 策略文档中，粘贴以下内容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
                }
            }
        },
        {
            "Action": [
                "s3:CreateBucket*",
                "s3:DeleteBucket*",
                "s3:Get*",
                "s3:List*",
                "s3:PutBucketTagging"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
```

```
        },
        {
            "Action": [
                "resource-groups:CreateGroup",
                "resource-groups:ListGroupResources",
                "resource-groups:DeleteGroup",
                "resource-groups:Tag"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "tag:GetResources",
                "tag:GetTagKeys",
                "tag:GetTagValues",
                "tag:TagResources",
                "tag:UntagResources"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

3. a. 请选择下一步，标签。
   b. 选择下一步，审核。
   c. 在查看策略页面上，输入 **S3ResourceCreationAndArtifactAccessPolicy** 作为名称。
   d. 选择创建策略。
4. 在导航窗格中，选择角色，然后选择创建角色。
5. 对于选择可信实体，选择自定义信任策略，然后输入以下 JSON 策略：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GivePermissionsToServiceCatalog",
            "Effect": "Allow",
            "Principal": {
                "Service": "servicecatalog.amazonaws.com"
```

```
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_id:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
TerraformExecutionRole*",
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogExternalParameterParserRole*",
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogTerraformOSParameterParserRole*"
                    ]
                }
            }
        }
    ]
}
```

6. 选择下一步。
7. 在策略列表中，选择您刚才创建的 `S3ResourceCreationAndArtifactAccessPolicy`。
8. 选择下一步。
9. 对于 Role name（角色名称），输入 **`SCLaunch-S3product`**。

> **Important**
>
> 启动角色名称必须以 “SCLaunch” 开头，后跟所需的角色名称。

10. 选择创建角色。

> **Important**
>
> 在 AWS Service Catalog 管理员账户中创建启动角色后，您还必须在 AWS Service Catalog 最终用户账户中创建相同的启动角色。最终用户账户中的角色必须与管理员账户中的角色同名并包含相同的策略。

要在 AWS Service Catalog 最终用户账户中创建启动角色

1. 以管理员身份登录最终用户账户，遵照 IAM 用户指南中在 JSON 选项卡上创建新策略的说明进行操作。
2. 重复上述要在 AWS Service Catalog 管理员账户中创建启动角色中的步骤 2-10。

> **Note**
>
> 在 AWS Service Catalog 最终用户账户中创建启动角色时，请确保在自定义信任策略中使用相同的管理员 **`AccountId`**。

现在您已经在管理员账户和最终用户账户中创建了启动角色，可以向产品添加启动约束了。

步骤 6：为 Terraform 产品添加启动约束

> **Important**
>
> 您必须为 HashiCorp Terraform 产品创建启动约束。如果没有启动约束，最终用户就无法预配置产品。

在管理员账户中创建启动角色后，您就可以关联启动角色和外部或 Terraform 云产品的启动约束了。

此启动约束让最终用户可以启动产品，并在启动之后将其作为预配置产品进行管理。有关更多信息，请参阅 AWS Service Catalog 启动约束。

借助使用启动约束，您可以遵循 IAM 最佳实践，将最终用户 IAM 权限保持在最低限度。有关更多信息，请参阅《IAM 用户指南》中的授予最低权限。

为产品分配启动约束

1. 打开AWS Service Catalog控制台，网址为 https://console.aws.amazon.com/servicecatalog。
2. 从左侧导航控制台中，选择产品组合。
3. 选择 S3 存储桶产品组合。
4. 在产品组合详细信息页面上，选择约束选项卡，然后选择创建约束。
5. 对于产品，选择简单 S3 存储桶。AWS Service Catalog 会自动选择启动约束类型。
6. 选择输入角色名称，然后选择 SCLaunch-S3product。
7. 选择创建。

> **Note**
>
> 在创建启动约束的账户中，以及使用此启动约束启动产品的用户账户中，都必须存在给定的角色名称。

步骤 7：授予最终用户访问权限

将启动限制应用于您的 HashiCorp Terraform 产品后，您就可以向分支账户中的最终用户授予访问权限了。

在本教程中，您会使用主体名称共享，将访问权限授予最终用户。主体名称是群组、角色和用户的名称，管理员可以在产品组合中指定这些名称，然后与产品组合共享。当您共享产品组合时，AWS Service Catalog 会验证这些主体名称是否已经存在。如果名称确实存在，则 AWS Service Catalog 会自动将匹配的 IAM 主体与共享产品组合关联以向最终用户授予访问权限。有关更多信息，请查看共享产品组合。

先决条件

如果您还没有为最终用户创建 IAM 群组，请参阅 向 AWS Service Catalog 最终用户授予权限。

提供对产品组合的访问权限

1. 导航到产品组合页面并选择 S3 存储桶产品组合。
2. 选择访问权限选项卡，然后选择授予访问权限。
3. 在访问权限类型窗格中，选择主体名称。
4. 在主体名称窗格中，选择主体名称类型，然后输入分支账户中所需最终用户的主体名称。

5. 选择授予访问权限。

第 8 步：与最终用户共享产品组合

AWS Service Catalog管理员可以使用 account-to-account 共享或AWS Organizations共享方式使用最终用户帐户分发电子档案夹。在本教程中，您将使用管理员账户（中心账户）与组织共享您的产品组合，该账户也是组织的管理账户。

自管理员中心账户共享产品组合

1. 打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicequotas/。
2. 在产品组合页面上，选择 S3 存储桶产品组合。在操作菜单中，选择共享。
3. 选择 AWS Organizations，然后筛选到您的组织结构。
4. 在 AWS 组织窗格中，选择最终用户账户（分支账户）。

   您还可以根据您的组织结构选择根节点与整个组织、父级组织单位 (OU) 或组织内的子组织单位共享产品组合。有关更多信息，请查看 共享产品组合。
5. 在共享设置窗格中，选择主体共享。
6. 选择共享。

成功与最终用户共享产品组合后，下一步要验证最终用户体验并预配置 Terraform 产品。

步骤 9：测试最终用户体验

要验证最终用户是否可以成功访问最终用户控制台视图并启动您的 **`Simple S3 bucket`** 产品，请以最终用户的身份登录 AWS 并执行以下任务。

验证最终用户可以访问最终用户控制台

- 打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/ 以查看：
  - 产品 - 用户可使用的产品。
  - 预配置产品 - 用户已启动的预配置产品。

要验证最终用户是否可以启动 Terraform 产品

1. 在控制台的产品部分，选择简单 S3 存储桶。

2. 选择启动产品，启动配置产品的向导。
3. 在启动简单 S3 存储桶页面上，输入 **`Amazon S3 product`** 用作预配置产品名称。
4. 在参数页面上，输入以下内容，然后选择下一步：
   - bucket_name — 为 Amazon S3 存储桶提供唯一名称。例如，**`terraform-s3-product`**。
5. 选择启动产品。控制台会显示 Amazon S3 产品启动的堆栈详细信息页面。产品的初始状态是更改中。AWS Service Catalog 需要花费几分钟时间来启动产品。要查看当前状态，请刷新您的浏览器。成功启动产品后，状态为可用。

   AWS Service Catalog 会创建名为 **`terraform-s3-product`** 的新 Amazon S3 存储桶。

步骤 10：监控 Terraform 预配置操作

如果您想监控配置操作，可以查看 Amazon CloudWatch 日志和AWS Step Functions任何配置工作流程。

配置工作流程中包含两台状态机：

- `ManageProvisionedProductStateMachine` — AWS Service Catalog 在预置新的 Terraform 产品以及更新现有 Terraform 预置产品时， 会调用此状态机。
- `TerminateProvisionedProductStateMachine` — AWS Service Catalog 将在终止现有 Terraform 预配置产品时调用此状态机。

要执行监控状态机

1. 打开 AWS 管理控制台并在安装了 Terraform 预置引擎的管理员中心账户中以管理员身份登录。
2. 打开 AWS Step Functions。
3. 在左侧导航面板中，选择状态机。
4. 选择ManageProvisionedProductStateMachine。
5. 在执行列表中，输入预配置的产品 ID 以找到您要执行的项。

> **Note**
>
> 在配置产品时，AWS Service Catalog 会创建预配置产品 ID。预配置产品 ID 的格式如下：**`pp-1111pwtn[ID number]`**。

6. 选择执行 ID。

在生成的执行详细信息页面上，您可以查看预配置工作流程中的所有步骤。您也可以查看任何失败的步骤以确定失败原因。

中的安全性 AWS Service Catalog

云安全位于 AWS 是最高优先级。作为 AWS 客户，您将受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方的共同责任 AWS 还有你。责任共担模式将其描述为云的 安全性和云中 的安全性：

- 云安全 — AWS 负责保护运行的基础架构 AWS 中的服务 AWS 云。 AWS 还为您提供可以安全使用的服务。作为安全措施的一部分，第三方审计师会定期测试和验证我们安全的有效性 AWS 合规计划。

  了解适用于以下方面的合规计划 AWS Service Catalog，请参阅 AWS 按合规计划划分的范围内的服务

- 云端安全 — 您的责任由以下因素决定 AWS 您使用的服务。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用分担责任模型时如何应用分担责任模型 AWS Service Catalog。 以下主题向您展示了如何配置 AWS Service Catalog 以实现您的安全和合规目标。你还会被介绍到其他 AWS 帮助您监控和保护您的服务的服务 AWS Service Catalog 资源的费用。

主题

- 中的数据保护 AWS Service Catalog
- Identity and Access Management AWS Service Catalog
- 登录和监控 AWS Service Catalog
- 合规性验证 AWS Service Catalog
- 韧性在 AWS Service Catalog
- 中的基础设施安全 AWS Service Catalog
- 的安全最佳实践 AWS Service Catalog

中的数据保护 AWS Service Catalog

这些区域有： AWS 分担责任模型适用于以下领域的数据保护 AWS Service Catalog。 如本模型所述，AWS 负责保护运行所有内容的全球基础设施 AWS Cloud。 您有责任保持对托管在此基础架构上的内容的控制。您还负责以下各项的安全配置和管理任务 AWS 服务 你用的。有关数据隐私的更多信息，

请参阅数据隐私FAQ。 有关欧洲数据保护的信息，请参阅 AWS 责任共担模型和GDPR博客文章 AWS 安全博客。

出于数据保护的目的，我们建议您进行保护 AWS 账户 凭据并使用设置个人用户 AWS IAM Identity Center 或者 AWS Identity and Access Management (IAM)。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用SSL/TLS与之通信 AWS 资源的费用。我们需要 TLS 1.2，建议使用 TLS 1.3。
- 使用设置API和用户活动记录 AWS CloudTrail。 有关使用 CloudTrail 轨迹捕获的信息 AWS 活动，请参阅使用中的 CloudTrail 轨迹 AWS CloudTrail 用户指南。
- 使用 AWS 加密解决方案，以及其中的所有默认安全控件 AWS 服务.
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在访问时需要 FIPS 140-3 经过验证的加密模块 AWS 通过命令行界面或API，使用FIPS端点。有关可用FIPS端点的更多信息，请参阅联邦信息处理标准 (FIPS) 140-3。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括你何时使用 AWS Service Catalog 或其他 AWS 服务 使用控制台，API，AWS CLI，或 AWS SDKs。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您URL向外部服务器提供，我们强烈建议您不要在中包含凭据信息，URL以验证您对该服务器的请求。

利用加密来保护数据

静态加密

AWS Service Catalog 使用使用亚马逊管理的密钥进行静态加密的 Amazon S3 存储桶和亚马逊 DynamoDB 数据库。要了解更多信息，请参阅 Amazon S3 和 Amazon DynamoDB 提供的有关静态加密的信息。

传输中加密

AWS Service Catalog 对呼叫者和之间传输的信息使用传输层安全 (TLS) 和客户端加密 AWS.

你可以私下访问 AWS Service Catalog APIs通过创建VPC终端节点从您的亚马逊虚拟私有云 (AmazonVPC) 中获取。对于VPC端点，VPC和之间的路由 AWS Service Catalog 由处理 AWS 无需互联网网关、NAT网关或VPN连接的网络。

使用的最新一代VPC端点 AWS Service Catalog 由提供动力 AWS PrivateLink，一个 AWS 技术实现了两者之间的私有连接 AWS 使用弹性网络接口的服务，IPs并在您的中使用私有接口VPCs。

Identity and Access Management AWS Service Catalog

访问权限 AWS Service Catalog 需要凭证。这些凭证必须具有访问权限 AWS 资源，例如 AWS Service Catalog 投资组合或产品。 AWS Service Catalog 与 AWS Identity and Access Management (IAM) 使您能够授予 AWS Service Catalog 管理员创建和管理产品以及授予所需的权限 AWS Service Catalog 最终用户启动产品和管理预配置产品所需的权限。这些策略要么由创建和管理 AWS 或者由管理员和最终用户单独完成。要控制访问权限，您可以将这些策略附加到与您一起使用的用户、群组和角色 AWS Service Catalog.

受众

你拥有的权限 AWS Identity and Access Management (IAM) 可能取决于你扮演的角色 AWS Service Catalog.

你所拥有的权限 AWS Identity and Access Management (IAM) 也可以取决于你扮演的角色 AWS Service Catalog.

管理员-作为 AWS Service Catalog 管理员，您需要具有管理员控制台的完全访问IAM权限以及允许您执行诸如创建和管理产品组合和产品、管理约束以及向最终用户授予访问权限等任务的权限。

最终用户-在您的最终用户可以使用您的产品之前，您需要向他们授予权限，使他们能够访问 AWS Service Catalog 最终用户控制台。他们还可以拥有启动产品和管理预配置产品的权限。

IAMad ministrator-如果您是管理IAM员，则可能需要详细了解如何编写策略来管理访问权限 AWS Service Catalog。 查看示例 AWS Service Catalog 您可以在中使用的基于身份的策略IAM，请参阅。the section called “AWS 托管策略”

基于身份的策略示例 AWS Service Catalog

主题

- 最终用户的控制台访问权限
- 最终用户的产品访问权限
- 管理预配置产品的示例策略

最终用户的控制台访问权限

AWSServiceCatalogEndUserFullAccess和**AWSServiceCatalogEndUserReadOnlyAccess**策略授予访问权限 AWS Service Catalog 最终用户控制台视图。当拥有其中任一策略的用户选择时 AWS Service Catalog 在 AWS Management Console，最终用户控制台视图显示他们有权启动的产品。

在最终用户可以成功启动产品之前 AWS Service Catalog 您向其授予访问权限，则必须向他们提供其他IAM权限，以允许他们使用每个底层产品 AWS 产品中的资源 AWS CloudFormation 模板。例如，如果产品模板包含亚马逊关系数据库服务 (AmazonRDS)，则必须向用户授予亚马逊启动该产品的RDS权限。

了解如何允许最终用户启动产品，同时强制执行最小访问权限 AWS 资源，请参阅the section called “使用约束”。

如果应用 **AWSServiceCatalogEndUserReadOnlyAccess** 策略，则您的用户将具有最终用户控制台的访问权限，但他们没有启动产品和管理预配置产品所需的权限。您可以使用直接向最终用户授予这些权限IAM，但如果您想限制最终用户的访问权限，则可以 AWS 资源，则应将策略附加到启动角色。然后你使用 AWS Service Catalog 将启动角色应用于产品的启动约束。有关应用启动角色、启动角色限制和示例启动角色的更多信息，请参阅AWS Service Catalog 启动约束。

> **Note**
>
> 如果您向用户授予IAM以下权限 AWS Service Catalog 管理员时，将改为显示管理员控制台视图。除非您希望最终用户具有管理员控制台视图的访问权限，否则不要授予他们这些权限。

最终用户的产品访问权限

在最终用户可以使用您授予访问权限的产品之前，您必须向他们提供额外的IAM权限以允许他们使用每个底层产品 AWS 产品中的资源 AWS CloudFormation 模板。例如，如果产品模板包含亚马逊关系数据库服务 (AmazonRDS)，则必须向用户授予亚马逊启动该产品的RDS权限。

如果应用 **AWSServiceCatalogEndUserReadOnlyAccess** 策略，则您的用户将具有最终用户控制台视图的访问权限，但他们没有启动产品和管理预配置产品所需的权限。您可以在中直接向最终用户授予这些权限IAM，但是如果您想限制最终用户的访问权限，则可以 AWS 资源，则应将策略附加到启动角色。然后你使用 AWS Service Catalog 将启动角色应用于产品的启动约束。有关应用启动角色、启动角色限制和示例启动角色的更多信息，请参阅AWS Service Catalog 启动约束。

管理预配置产品的示例策略

您可以创建自定义策略来帮助满足组织的安全要求。以下示例介绍如何自定义每个操作的访问级别，以提供用户、角色和账户级支持。您可以向用户授予查看、更新、终止和管理以下预配置产品的访问权限：1) 由该用户创建的预配置产品；2) 由其他用户使用该用户的角色创建的预配置产品；或 3) 由该用户登录的账户创建的预配置产品。这种访问模式是分层的 - 授予账户级访问权限时会同时授予角色级和用户级访问权限，添加角色级访问权限时会授予用户级访问权限，但不会授予账户级访问权限。您可以在策略中JSON使用`Condition`区块（如`accountLevel``roleLevel`、或）来指定这些内容`userLevel`。

这些示例也适用于的访问级别 AWS Service Catalog API写入操作：`UpdateProvisionedProduct`和`TerminateProvisionedProduct`，读取操作：`DescribeRecordScanProvisionedProducts`、和`ListRecordHistory`。`ScanProvisionedProducts`和`ListRecordHistory`API操作`AccessLevelFilterKey`用作输入，该键的值对应于此处讨论的`Condition`区块级别（等同`accountLevel`于 “账户”、`roleLevel` “角色” 和 `userLevel` “用户” 的`AccessLevelFilterKey`值）。有关更多信息，请参阅《Service Catalog 开发人员指南》。

示例

- 预配置产品的管理员完全访问权限
- 预配置产品的最终用户访问权限
- 预配置产品的管理员部分访问权限

预配置产品的管理员完全访问权限

下面的策略允许自由读写账户级目录中的预配置产品和记录。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicecatalog:*"
         ],
         "Resource":"*",
         "Condition": {
            "StringEquals": {
               "servicecatalog:accountLevel": "self"
```

```
            }
         }
      }
   ]
}
```

该策略的功能等同于下面的策略：

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

未在任何策略中为以下项指定`Condition`封锁 AWS Service Catalog 被视为与指定`"servicecatalog:accountLevel"`访问权限相同。注意，`accountLevel` 访问权限包含 `roleLevel` 和 `userLevel` 访问权限。

预配置产品的最终用户访问权限

下面的策略将用户的访问权限限制为只能读写自己创建的预配置产品和关联记录。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:DescribeProduct",
                "servicecatalog:DescribeProductView",
                "servicecatalog:DescribeProvisioningParameters",
                "servicecatalog:DescribeRecord",
                "servicecatalog:ListLaunchPaths",
                "servicecatalog:ListRecordHistory",
                "servicecatalog:ProvisionProduct",
```

```
                "servicecatalog:ScanProvisionedProducts",
                "servicecatalog:SearchProducts",
                "servicecatalog:TerminateProvisionedProduct",
                "servicecatalog:UpdateProvisionedProduct"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:userLevel": "self"
                }
            }
        }
    ]
 }
```

预配置产品的管理员部分访问权限

如果将下面的两个策略应用到同一个用户，可产生称作“管理员部分访问权限”的访问类型 - 即提供完全的只读访问和有限的写入访问。这意味着用户能够查看账户级目录中的任何预配置产品或关联记录，但无法对不归该用户所有的任何预配置产品或记录执行任何操作。

第一个策略允许用户对自己创建的预配置产品执行写入操作，但不允许对其他用户创建的预配置产品执行写入操作。第二个策略允许用户对所有用户、角色或账户创建的预配置产品执行读取操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:DescribeProduct",
                "servicecatalog:DescribeProductView",
                "servicecatalog:DescribeProvisioningParameters",
                "servicecatalog:ListLaunchPaths",
                "servicecatalog:ProvisionProduct",
                "servicecatalog:SearchProducts",
                "servicecatalog:TerminateProvisionedProduct",
                "servicecatalog:UpdateProvisionedProduct"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:userLevel": "self"
```

```
                }
            }
        }
    ]
}
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:DescribeRecord",
                "servicecatalog:ListRecordHistory",
                "servicecatalog:ScanProvisionedProducts"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:accountLevel": "self"
                }
            }
        }
    ]
}
```

AWS 的托管策略 AWS Service Catalog AppRegistry

AWS 托管策略：**`AWSServiceCatalogAdminFullAccess`**

你可以附加`AWSServiceCatalogAdminFullAccess`到你的IAM实体。 AppRegistry 还将此策略附加 AppRegistry 到允许代表您执行操作的服务角色。

本政策授予 *`administrative`* 权限，允许完全访问管理员控制台视图，并授予创建和管理产品和产品组合的权限。

权限详细信息

该策略包含以下权限。

- `servicecatalog`— 允许委托人拥有管理员控制台视图的全部权限，以及创建和管理产品组合和产品、管理限制、向最终用户授予访问权限以及在其中执行其他管理任务的能力 AWS Service Catalog.
- `cloudformation`— 允许 AWS Service Catalog 列出、读取、写入和标记的全部权限 AWS CloudFormation 堆栈。
- `config`— 允许 AWS Service Catalog 通过以下方式访问产品组合、产品和预配置产品的权限有限 AWS Config.
- `iam`— 允许主体拥有查看和创建创建和管理产品和产品组合所需的服务用户、用户组或角色的全部权限。
- `ssm`— 允许 AWS Service Catalog 要使用 AWS Systems Manager 列出并阅读当前版本中的 Systems Manager 文档 AWS 账户和 AWS 区域。

查看政策:AWSServiceCatalogAdminFullAccess.

AWS 托管策略：`AWSServiceCatalogAdminReadOnlyAccess`

你可以附加`AWSServiceCatalogAdminReadOnlyAccess`到你的IAM实体。 AppRegistry 还将此策略附加 AppRegistry 到允许代表您执行操作的服务角色。

本政策授予 *`read-only`* 允许完全访问管理员控制台视图的权限。此策略不授予创建或管理产品和产品组合的访问权限。

权限详细信息

该策略包含以下权限。

- `servicecatalog` — 允许主体拥有管理员控制台视图的只读权限。
- `cloudformation`— 允许 AWS Service Catalog 列出和阅读的权限有限 AWS CloudFormation 堆栈。
- `config`— 允许 AWS Service Catalog 通过以下方式访问产品组合、产品和预配置产品的权限有限 AWS Config.
- `iam`— 允许主体查看创建和管理产品和产品组合所需的服务用户、组或角色的有限权限。
- `ssm`— 允许 AWS Service Catalog 要使用 AWS Systems Manager 列出并阅读当前版本中的 Systems Manager 文档 AWS 账户和 AWS 区域。

查看政策:AWSServiceCatalogAdminReadOnlyAccess.

AWS 托管策略：`AWSServiceCatalogEndUserFullAccess`

你可以附加`AWSServiceCatalogEndUserFullAccess`到你的IAM实体。 AppRegistry 还将此策略附加 AppRegistry 到允许代表您执行操作的服务角色。

本政策授予 *`contributor`* 权限，允许完全访问最终用户控制台视图，并授予启动产品和管理预配置产品的权限。

权限详细信息

该策略包含以下权限。

- `servicecatalog` – 允许主体对最终用户控制台视图的完全权限以及启动产品和管理预配置产品的权限。
- `cloudformation`— 允许 AWS Service Catalog 列出、读取、写入和标记的全部权限 AWS CloudFormation 堆栈。
- `config`— 允许 AWS Service Catalog 通过以下方式列出和阅读有关产品组合、产品和预配置产品的详细信息的权限有限 AWS Config.
- `ssm`— 允许 AWS Service Catalog 要使用 AWS Systems Manager 阅读当前版本中的 Systems Manager 文档 AWS 账户和 AWS 区域。

查看政策:AWSServiceCatalogEndUserFullAccess.

AWS 托管策略：`AWSServiceCatalogEndUserReadOnlyAccess`

你可以附加`AWSServiceCatalogEndUserReadOnlyAccess`到你的IAM实体。 AppRegistry 还将此策略附加 AppRegistry 到允许代表您执行操作的服务角色。

本政策授予 *`read-only`* 允许对最终用户控制台视图进行只读访问的权限。此策略不授予启动产品或管理预配置产品的权限。

权限详细信息

该策略包含以下权限。

- `servicecatalog`— 允许主体拥有访问最终用户控制台视图的只读权限。
- `cloudformation`— 允许 AWS Service Catalog 列出和阅读的权限有限 AWS CloudFormation 堆栈。
- `config`— 允许 AWS Service Catalog 通过以下方式列出和阅读有关产品组合、产品和预配置产品的详细信息的权限有限 AWS Config.

- `ssm`— 允许 AWS Service Catalog 要使用 AWS Systems Manager 阅读当前版本中的 Systems Manager 文档 AWS 账户和 AWS 区域。

查看政策:AWSServiceCatalogEndUserReadOnlyAccess.

AWS 托管策略：**`AWSServiceCatalogSyncServiceRolePolicy`**

AWS Service Catalog 将此策略附加到`AWSServiceRoleForServiceCatalogSync`服务相关角色 (SLR)，允许 AWS Service Catalog 将外部存储库中的模板同步到 AWS Service Catalog 产品。

此策略授予的权限允许有限的访问 AWS Service Catalog 操作（例如，API呼叫），以及其他 AWS 服务操作 AWS Service Catalog 取决于。

权限详细信息

该策略包含以下权限。

- `servicecatalog`— 允许 AWS Service Catalog 对象同步角色的访问权限有限 AWS Service Catalog 公众APIs。
- `codeconnections`— 允许 AWS Service Catalog 工件同步角色限制了对CodeConnections 公众的访问权限APIs。
- `cloudformation`— 允许 AWS Service Catalog 对象同步角色的访问权限有限 AWS CloudFormation 公众APIs。

查看政策:AWSServiceCatalogSyncServiceRolePolicy.

服务相关角色详情

AWS Service Catalog 将上述权限详细信息用于用户创建或更新`AWSServiceRoleForServiceCatalogSync`服务相关角色时创建的服务相关角色 AWS Service Catalog 使用的产品 CodeConnections。您可以使用修改此策略 AWS CLI, AWS API，或者通过 AWS Service Catalog console。有关如何创建、编辑和删除服务相关角色的更多信息，请参阅使用服务相关角色 () SLRs AWS Service Catalog.

`AWSServiceRoleForServiceCatalogSync`服务相关角色中包含的权限允许 AWS Service Catalog 代表客户执行以下操作。

- `servicecatalog:ListProvisioningArtifacts`— 允许 AWS Service Catalog 工件同步角色用于列出给定对象的置备工件 AWS Service Catalog 已同步到存储库中模板文件的产品。

- `servicecatalog:DescribeProductAsAdmin`— 允许 AWS Service Catalog 工件同步角色`DescribeProductAsAdminAPI`用于获取的详细信息 AWS Service Catalog 产品及其关联的预配置对象，这些对象已同步到存储库中的模板文件。构件同步角色使用此调用的输出来验证产品对预配置构件的服务限额限制。
- `servicecatalog:DeleteProvisioningArtifact`— 允许 AWS Service Catalog 用于删除已置备对象的对象同步角色。
- `servicecatalog:ListServiceActionsForProvisioningArtifact`— 允许 AWS Service Catalog 构件同步角色，用于确定服务操作是否与置备对象相关联，并确保在关联服务操作时不会删除置备对象。
- `servicecatalog:DescribeProvisioningArtifact`— 允许 AWS Service Catalog artifact sync 角色用于从中检索详细信息 `DescribeProvisioningArtifactAPI`，包括`SourceRevisionInfo`输出中提供的提交 ID。
- `servicecatalog:CreateProvisioningArtifact`— 允许 AWS Service Catalog artifact sync 角色用于在外部存储库中检测到源模板文件有更改（例如，已提交 git-push）时创建新的预配置工件。
- `servicecatalog:UpdateProvisioningArtifact`— 允许 AWS Service Catalog 构件同步角色，用于更新已连接或已同步产品的已配置对象。
- `codeconnections:UseConnection`— 允许 AWS Service Catalog 构件同步角色使用现有连接来更新和同步产品。
- `cloudformation:ValidateTemplate`— 允许 AWS Service Catalog 对象同步角色的访问权限有限 AWS CloudFormation 验证外部存储库中使用的模板的模板格式并验证是否 AWS CloudFormation 可以支持该模板。

AWS 托管策略：**`AWSServiceCatalogOrgsDataSyncServiceRolePolicy`**

AWS Service Catalog 将此策略附加到`AWSServiceRoleForServiceCatalogOrgsDataSync`服务相关角色 (SLR)，允许 AWS Service Catalog 与之同步 AWS Organizations.

此策略授予的权限允许有限的访问 AWS Service Catalog 操作（例如，API呼叫），以及其他 AWS 服务操作 AWS Service Catalog 取决于。

权限详细信息

该策略包含以下权限。

- `organizations`— 允许 AWS Service Catalog 数据同步角色的访问权限有限 AWS Organizations 公众APIs。

查看政策:AWSServiceCatalogOrgsDataSyncServiceRolePolicy.

服务相关角色详情

AWS Service Catalog 将上述权限详细信息用于用户启用时创建的`AWSServiceRoleForServiceCatalogOrgsDataSync`服务相关角色 AWS Organizations 共享投资组合访问权限或创建投资组合共享。您可以使用修改此策略 AWS CLI, AWS API，或者通过 AWS Service Catalog console。有关如何创建、编辑和删除服务相关角色的更多信息，请参阅使用服务相关角色 () SLRs AWS Service Catalog.

`AWSServiceRoleForServiceCatalogOrgsDataSync`服务相关角色中包含的权限允许 AWS Service Catalog 代表客户执行以下操作。

- `organizations:DescribeAccount`— 允许 AWS Service Catalog 要检索的 Organitions 数据同步角色 AWS Organizations有关指定账户的相关信息。
- `organizations:DescribeOrganization`— 允许 AWS Service Catalog Organizations Data Sync 角色用于检索有关用户账户所属组织的信息。
- `organizations:ListAccounts`— 允许 AWS Service Catalog Organizations Data Sync 角色用于列出用户组织中的账户。
- `organizations:ListChildren`— 允许 AWS Service Catalog Organizations Data Sync 角色用于列出包含在指定父组织单位或根目录中的所有组织单位 (UOs) 或帐户。
- `organizations:ListParents`— 允许 AWS Service Catalog Organizations Data Sync 角色用于列出根目录或OUs充当指定子组织单位或账户的直系父级。
- `organizations:ListAWSServiceAccessForOrganization`— 允许 AWS Service Catalog Organizations Data Sync 角色用于检索列表 AWS 用户允许与其组织集成的服务。

已弃用的策略

以下托管策略已弃用：

- ServiceCatalogAdminFullAccess— AWSServiceCatalogAdminFullAccess改用。
- ServiceCatalogAdminReadOnlyAccess— AWSServiceCatalogAdminReadOnlyAccess改用。
- ServiceCatalogEndUserFullAccess— AWSServiceCatalogEndUserFullAccess改用。
- ServiceCatalogEndUserAccess— AWSServiceCatalogEndUserReadOnlyAccess改用。

使用以下过程可确保管理员和最终用户获得使用当前策略的权限。

要从已弃用的策略迁移到当前的策略，请参阅中的添加和删除IAM身份权限 AWS Identity and Access Management 用户指南。

AppRegistry 更新到 AWS 托管策略

查看有关更新的详细信息 AWS AppRegistry 自该服务开始跟踪这些更改以来的管理策略。要获得有关此页面变更的自动提醒，请订阅 “ AppRegistry 文档历史记录” 页面上的订阅RSS源。

更改	描述	日期
AWSServiceCatalogSyncServiceRolePolicy— 更新托管策略	AWS Service Catalog 已将`AWSServiceCatalogSyncServiceRolePolicy` 政策更新`codestar-connections` 为`codeconnections` 。	2024 年 5 月 7 日
AWSServiceCatalogAdminFullAccess— 更新托管策略	AWS Service Catalog 更新了`AWSServiceCatalogAdminFullAccess` 策略以包含所需的权限 AWS Service Catalog 管理员在其账户中创建`AWSServiceRoleForServiceCatalogOrgsDataSync` 服务相关角色 (SLR)。	2023 年 4 月 14 日
AWSServiceCatalogOrgsDataSyncServiceRolePolicy：新托管策略	AWS Service Catalog 添加了`AWSServiceCatalogOrgsDataSyncServiceRolePolicy` 附加到`AWSServiceRoleForServiceCatalogOrgsDataSync` 服务相关角色 (SLR) 的，允许 AWS Service Catalog 与之同步 AWS Organizations。 此政策允许有限的访问权限 AWS Service	2023 年 4 月 14 日

更改	描述	日期
	Catalog 操作（例如，API呼叫），以及其他 AWS 服务操作 AWS Service Catalog 取决于。	
AWSServiceCatalogAdminFullAccess— 更新托管策略	AWS Service Catalog 更新了`AWSServiceCatalogAdminFullAccess` 策略以包含所有权限 AWS Service Catalog 管理员并创建与的兼容性 AppRegistry。	2023 年 1 月 12 日
AWSServiceCatalogSyncServiceRolePolicy：新托管策略	AWS Service Catalog 添加了`AWSServiceCatalogSyncServiceRolePolicy` 策略，该策略附加到`AWSServiceRoleForServiceCatalogSync` 服务相关角色 (SLR)。此政策允许 AWS Service Catalog 将外部存储库中的模板同步到 AWS Service Catalog 产品。	2022 年 11 月 18 日
AWSServiceRoleForServiceCatalogSync— 新的服务相关角色	AWS Service Catalog 添加了`AWSServiceRoleForServiceCatalogSync` 服务相关角色 (SLR)。此角色是必需的 AWS Service Catalog 使用 CodeConnections 、创建、更新和描述 AWS Service Catalog 为产品配置工件。	2022 年 11 月 18 日

更改	描述	日期
AWSServiceCatalogAdminFullAccess— 更新了托管策略	AWS Service Catalog 更新了`AWSServiceCatalogAdminFullAccess` 策略以包含所有必需的权限 AWS Service Catalog 管理员。该策略确定了管理员可以对所有人采取的具体操作 AWS Service Catalog 资源，例如创建、描述、删除等。此外，该政策已更改为支持最近推出的一项功能，即基于属性的访问控制 (ABAC) AWS Service Catalog。 ABAC允许您将`AWSServiceCatalogAdminFullAccess` 策略用作模板来允许或拒绝对以下操作的操作 AWS Service Catalog 基于标签的资源。有关的更多信息ABAC，请参见用ABAC途 AWS中的AWS Identity and Access Management.	2022 年 9 月 30 日
AppRegistry 已开始跟踪更改	AppRegistry 开始跟踪它的变化 AWS 托管策略。	2022 年 9 月 15 日

将服务相关角色用于 AWS Service Catalog

AWS Service Catalog uses AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种与之直接关联的独特IAM角色类型 AWS Service Catalog。 服务相关角色由以下人员预定义 AWS Service Catalog 并包括该服务调用其他服务所需的所有权限 AWS 代表您提供服务。

与服务相关的角色可以进行设置 AWS Service Catalog 更简单，因为您不必手动添加必要的权限。AWS Service Catalog 定义其服务相关角色的权限，除非另有定义，否则只有 AWS Service Catalog 可以扮演自己的角色。定义的权限包括信任策略和权限策略，并且该权限策略不能附加到任何其他IAM实体。

只有在首先删除相关资源后，您才能删除服务相关角色。这样可以保护你的 AWS Service Catalog 资源，因为您不能无意中移除对资源的访问权限。

有关支持服务相关角色的其他服务的信息，请参阅 AWS 与之配合使用的服务，IAM然后在 “服务相关角色” 列中查找 “是” 的服务。请选择是与查看该服务的服务相关角色文档的链接。

`AWSServiceRoleForServiceCatalogSync` 的服务相关角色权限

AWS Service Catalog 可以使用名为的服务相关角色 **`AWSServiceRoleForServiceCatalogSync`**— 此服务相关角色是必需的 AWS Service Catalog 使用 CodeConnections 、创建、更新和描述 AWS Service Catalog 为产品配置工件。

`AWSServiceRoleForServiceCatalogSync` 服务相关角色信任以下服务代入该角色：

- `sync.servicecatalog.amazonaws.com`

名为 allows 的角色权限策略AWSServiceCatalogSyncServiceRolePolicy允许 AWS Service Catalog 对指定资源完成以下操作：

- 操作：`CodeConnections` 上的 `Connection`
- 操作：`Create, Update, and Describe`on f `ProvisioningArtifact` or a AWS Service Catalog product

必须配置权限以允许实IAM体（例如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM用户指南》中的服务相关角色权限。

创建 **`AWSServiceRoleForServiceCatalogSync`** 服务相关角色

您无需手动创建`AWSServiceRoleForServiceCatalogSync`服务相关角色。 AWS Service Catalog 当您在中建立 CodeConnections 服务相关角色时，会自动为您创建服务相关角色 AWS Management Console， AWS CLI，或者 AWS API.

> **Important**
>
> 如果您在其他使用此角色支持的功能的服务中完成某个操作，此服务相关角色可以出现在您的账户中。另外，如果你使用的是 AWS Service Catalog 在 2022 年 11 月 18 日之前提供服务，当时它开始支持与服务相关的角色 AWS Service Catalog 在您的账户中创建了该`AWSServiceRoleForServiceCatalogSync`角色。要了解更多信息，请参阅我的IAM账户中出现了一个新角色。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当你建立时 CodeConnections， AWS Service Catalog 再次为您创建服务相关角色。

您也可以使用IAM控制台创建具有同步功能的服务相关角色 AWS Service Catalog 产品用例。在 AWS CLI 或者 AWS API，使用服务名称创建服务相关角色。`sync.servicecatalog.amazonaws.com`有关更多信息，请参阅《IAM用户指南》中的创建服务相关角色。如果您删除了此服务相关角色，可以使用同样的过程再次创建角色。

`AWSServiceRoleForServiceCatalogOrgsDataSync` 的服务相关角色权限

AWS Service Catalog 可以使用名为的服务相关角色 **`AWSServiceRoleForServiceCatalogOrgsDataSync`**— 此服务相关角色是必需的 AWS Service Catalog 要与之保持同步的组织 AWS Organizations.

`AWSServiceRoleForServiceCatalogOrgsDataSync` 服务相关角色信任以下服务代入该角色：

- `orgsdatasync.servicecatalog.amazonaws.com`

除了 `AWSServiceCatalogOrgsDataSyncServiceRolePolicy` 托管策略外，`AWSServiceRoleForServiceCatalogOrgsDataSync` 服务相关角色还要求您使用以下信任策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "orgsdatasync.servicecatalog.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

名为 allows 的角色权限策略AWSServiceCatalogOrgsDataSyncServiceRolePolicy允许 AWS Service Catalog 对指定资源完成以下操作：

- 操作：DescribeAccount、DescribeOrganization、以及在 Organizations accounts 中的 ListAWSServiceAccessForOrganization
- 操作：ListAccounts、ListChildren、以及在 Organizations accounts 中的 ListParent

必须配置权限以允许实IAM体（例如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM用户指南》中的服务相关角色权限。

创建 **AWSServiceRoleForServiceCatalogOrgsDataSync** 服务相关角色

您无需手动创建AWSServiceRoleForServiceCatalogOrgsDataSync服务相关角色。 AWS Service Catalog 将您的启用与 AWS Organizations 共享或共享产品组合视为许可的操作 AWS Service Catalog 代表你SLR在后台创建。

AWS Service Catalog 在您请求时自动为您创建服务相关角色，EnableAWSOrganizationsAccess或者CreatePortfolioShare在 AWS Management Console， AWS CLI，或者 AWS API.

> **Important**
>
> 如果您在其他使用此角色支持的功能的服务中完成某个操作，此服务相关角色可以出现在您的账户中。要了解更多信息，请参阅我的IAM账户中出现了一个新角色。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当你请求EnableAWSOrganizationsAccess或时CreatePortfolioShare， AWS Service Catalog 再次为您创建服务相关角色。

为 编辑服务相关角色 AWS Service Catalog

AWS Service Catalog 不允许您编辑AWSServiceRoleForServiceCatalogSync或与AWSServiceRoleForServiceCatalogOrgsDataSync服务相关的角色。创建服务相关角色后，将无法更改角色名称，因为可能有多个实体引用该角色。但是，您可以使用编辑角色的描述IAM。有关更多信息，请参阅《IAM用户指南》中的编辑服务相关角色。

删除 的服务相关角色 AWS Service Catalog

你可以使用IAM控制台， AWS CLI，或者 AWS API手动删除AWSServiceRoleForServiceCatalogSync或AWSServiceRoleForServiceCatalogOrgsDataSyn

为此，必须先手动删除所有使用服务相关角色的资源（例如，任何 AWS Service Catalog 已同步到外部存储库的产品），然后可以手动删除服务相关角色。

支持的区域 AWS Service Catalog 服务相关角色

AWS Service Catalog 支持在提供服务的所有地区使用服务相关角色。有关更多信息，请参阅 AWS 区域和终端节点。

区域名称	区域标识	Support in AWS Service Catalog
美国东部（弗吉尼亚州北部）	us-east-1	是
美国东部（俄亥俄州）	us-east-2	是
美国西部（加利福尼亚北部）	us-west-1	是
美国西部（俄勒冈州）	us-west-2	是
非洲（开普敦）	af-south-1	是
亚太地区（香港）	ap-east-1	是
亚太地区（雅加达）	ap-southeast-3	是
亚太地区（孟买）	ap-south-1	是
Asia Pacific (Osaka)	ap-northeast-3	是
Asia Pacific (Seoul)	ap-northeast-2	是
亚太地区（新加坡）	ap-southeast-1	是
亚太地区（悉尼）	ap-southeast-2	是
Asia Pacific (Tokyo)	ap-northeast-1	是
加拿大（中部）	ca-central-1	是
欧洲（法兰克福）	eu-central-1	是
欧洲地区（爱尔兰）	eu-west-1	是

区域名称	区域标识	Support in AWS Service Catalog
欧洲地区（伦敦）	eu-west-2	是
欧洲（米兰）	eu-south-1	是
欧洲（巴黎）	eu-west-3	是
欧洲地区（斯德哥尔摩）	eu-north-1	是
中东（巴林）	me-south-1	是
南美洲（圣保罗）	sa-east-1	是
AWS GovCloud （美国东部）	us-gov-east-1	否
AWS GovCloud （美国西部）	us-gov-west-1	否

故障排除 AWS Service Catalog 身份和访问权限

使用以下信息来帮助您诊断和修复在使用时可能遇到的常见问题 AWS Service Catalog 和IAM。

主题

- 我无权在以下位置执行操作 AWS Service Catalog
- 我无权执行 iam:PassRole
- 我想允许我以外的人进入 AWS 账户访问我的 AWS Service Catalog 资源

我无权在以下位置执行操作 AWS Service Catalog

如果 AWS Management Console 告诉您您无权执行某项操作，那么您必须联系管理员寻求帮助。管理员是向您提供登录凭证的人。当 mateojackson 用户尝试使用控制台查看虚构 my-example-widget资源的详细信息但没有虚构权限时，就会出现以下示例错误。`aws:GetWidget`

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

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他使用 `aws:GetWidget` 操作访问 `my-example-widget` 资源。

我无权执行 `iam:PassRole`

如果您收到错误消息，提示您无权执行 `iam:PassRole` 操作，则必须联系您的管理员寻求帮助。管理员是指提供用户名和密码的人员。要求该人更新您的政策，以允许您将角色传递给 AWS Service Catalog.

一段时间 AWS 服务允许您将现有角色传递给该服务，而不是创建新的服务角色或与服务相关的角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的用户尝试使用控制台在中执行操作时，会出现以下示例错误 AWS Service Catalog。 但是，该操作要求服务拥有服务角色授予的权限。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，Mary 要求她的管理员更新她的策略以允许她执行 iam: PassRole 操作。

我想允许我以外的人进入 AWS 账户访问我的 AWS Service Catalog 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解是否 AWS Service Catalog 支持这些功能，请参阅 AWS Identity and Access Management in AWS Service Catalog中的 AWS Service Catalog 管理员指南。
- 了解如何提供对您的资源的访问权限 AWS 您拥有的帐户，请参阅向其他IAM用户提供访问权限 AWS 您在《IAM用户指南》中拥有的帐户。
- 了解如何向第三方提供对您的资源的访问权限 AWS 帐户，请参阅提供访问权限 AWS 《IAM用户指南》中第三方拥有的账户。
- 要了解如何通过联合身份验证提供访问权限，请参阅《用户指南》中的向经过外部身份验证的用户提供访问权限（联合身份验证）。IAM
- 要了解使用角色和基于资源的策略进行跨账户访问的区别，请参阅用户指南中的IAM角色与基于资源的策略的IAM区别。

控制访问权限

a AWS Service Catalog portfolio 为您的管理员提供了对最终用户组进行一定级别的访问控制。将用户添加到某个产品组合后，这些用户可以浏览并启动其中的任何产品。有关更多信息，请参阅 the section called “管理产品组合”。

约束

约束控制在从特定产品组合启动产品时应用于最终用户的规则。您使用约束对产品进行限制，以便进行管理或控制成本。有关约束的更多信息，请参阅 the section called “使用约束”。

AWS Service Catalog 启动限制使您可以更好地控制最终用户所需的权限。当您的管理员为产品组合中的产品创建启动约束时，启动约束会关联最终用户从该产品组合中启动产品时使用的角色ARN。使用此模式，您可以控制访问权限 AWS 资源创建。有关更多信息，请参阅 the section called “启动约束”。

登录和监控 AWS Service Catalog

AWS Service Catalog 与 AWS CloudTrail，一项可捕获所有内容的服务 AWS Service Catalog API 调用日志文件并将其传送到您指定的 Amazon S3 存储桶。有关更多信息，请参阅日志 AWS Service Catalog API与... 通话 CloudTrail.

您还可以使用通知限制来设置有关堆栈事件的 Amazon SNS 通知。有关更多信息，请参阅 the section called “通知约束”。

合规性验证 AWS Service Catalog

第三方审计师评估以下各项的安全性和合规性 AWS Service Catalog 作为多个的一部分 AWS 合规计划，包括以下内容：

- 系统和组织控制 (SOC)
- 支付卡行业数据安全标准 (PCIDSS)
- 联邦风险和授权管理计划（美联储RAMP）
- Health Insurance 流通与责任法案 () HIPAA

有关清单 AWS 特定合规计划范围内的AWS服务，请参阅合规性计划范围内的服务。 有关一般信息，请参阅 AWS 合规计划 。

您可以使用以下方式下载第三方审计报告 AWS Artifact。 有关更多信息，请参阅中的下载报告 AWS Artic al。

您在使用时的合规责任 AWS Service Catalog 取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了在上部署以安全性和合规性为重点的基准环境的步骤 AWS.
- HIPAA安全与合规架构白皮书 — 本白皮书描述了公司如何使用 AWS 创建HIPAA兼容的应用程序。
- AWS 合规资源 — 此工作簿和指南集可能适用于您的行业和所在地。
- AWS Config— 这个 AWS 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 这个 AWS 服务可全面了解您的安全状态 AWS 它可以帮助您检查自己是否符合安全行业标准和最佳实践。

韧性在 AWS Service Catalog

这些区域有： AWS 全球基础设施是围绕着建立的 AWS 地区和可用区。 AWS 区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础架构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 地区和可用区，请参阅 AWS 全球基础设施。

除了 AWS 全球基础设施， AWS Service Catalog 优惠 AWS Service Catalog 自助操作。通过自助服务操作，客户可以减少管理维护和最终用户培训，同时合规且遵守安全措施。通过自助服务操作，作为管理员，您可以使最终用户能够执行操作任务，例如备份和恢复、解决问题、运行经批准的命令以及申请权限 AWS Service Catalog。 要了解更多信息，请参阅the section called “使用服务操作”。

中的基础设施安全 AWS Service Catalog

作为一项托管服务， AWS Service Catalog 受保护 AWS 全球网络安全。有关信息 AWS 安全服务及其方式 AWS 保护基础架构，请参阅 AWS 云安全。设计你的 AWS 使用基础设施安全最佳实践的环境，请参阅安全支柱中的基础设施保护 AWS 架构精良的框架。

你用 AWS 已发布的访问API呼叫 AWS Service Catalog 通过网络。客户端必须支持以下内容：

- 传输层安全 (TLS)。我们需要 TLS 1.2，建议使用 TLS 1.3。

- 具有完美前向保密性的密码套件 ()，例如（Ephemeral Diffie-HellmanPFS）或（Elliptic C DHE urve Ephemeral Diffie-Hellman）。ECDHE大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与IAM委托人关联的私有访问密钥对请求进行签名。或者你可以使用 AWS Security Token Service (AWS STS) 生成用于签署请求的临时安全证书。

随着 AWS Service Catalog，您可以控制存储数据的区域。产品组合和产品仅在您提供它们的区域中可用。您可以使用将产品复制`CopyProduct`API到另一个区域。

的安全最佳实践 AWS Service Catalog

AWS Service Catalog 提供了许多安全功能，供您在制定和实施自己的安全策略时考虑。以下最佳实践是一般指导原则，并不代表完整安全解决方案。这些最佳实践可能不适合环境或不满足环境要求，请将其视为有用的考虑因素而不是惯例。

您可以定义规则，限制用户在启动产品时输入的参数值。这些规则之所以称为模板约束，是因为它们限制了模板约束的方式 AWS CloudFormation 产品的模板已部署。您可以使用简单的编辑器创建模板约束，然后将其应用到各个产品。

AWS Service Catalog 在配置新产品或更新已在使用的产品时应用限制。在应用到组合和产品的所有约束中，它始终应用限制性最强的约束。例如，假设产品允许启动所有 Amazon EC2 实例，并且产品组合有两个限制：一个允许启动所有非GPU类型EC2实例，另一个限制仅允许启动 t1.micro 和 m1.small EC2 实例。在这个例子中， AWS Service Catalog 应用第二个限制性更强的约束（t1.micro 和 m1.small）。

您可以限制最终用户的访问权限 AWS 将IAM策略附加到启动角色时的资源。然后你使用 AWS Service Catalog 创建启动约束，以便在启动产品时使用该角色。

要了解有关托管策略的更多信息 AWS Service Catalog，请参阅 AWS 的托管策略 AWS Service Catalog.

管理目录

AWS Service Catalog 提供一个用于从管理员控制台管理产品组合、产品和约束的接口。

> **Note**
>
> 要执行本部分中的任一任务，您必须具有 AWS Service Catalog 的管理员权限。有关更多信息，请参阅 Identity and Access Management AWS Service Catalog。

任务

- 管理产品组合
- 管理产品
- 使用 AWS Service Catalog 约束
- AWS Service Catalog 服务操作
- 将 AWS Marketplace 产品添加到您的产品组合
- 使用 AWS CloudFormation StackSets
- 管理预算

管理产品组合

您可以在 AWS Service Catalog 管理员控制台的产品组合页面上创建、查看和更新产品组合。

任务

- 创建、查看和删除产品组合
- 查看产品组合详细信息
- 创建和删除产品组合
- 添加产品
- 添加约束
- 向用户授予访问权限
- 共享产品组合

- 共享和导入产品组合

创建、查看和删除产品组合

产品组合页面显示您在当前区域中创建的产品组合的列表。使用此页面可创建新的产品组合、查看产品组合的详细信息或从您的账户中删除产品组合。

查看产品组合页面

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 根据需要选择不同的区域。
3. 如果您是首次接触 AWS Service Catalog，您将会看到 AWS Service Catalog 起始页。选择 Get started 以创建产品组合。按照说明创建您的第一个产品组合，然后继续进入产品组合页面。

在使用 AWS Service Catalog 时，您可以随时返回产品组合页面，选择导航栏中的 Service Catalog，然后选择产品组合即可。

查看产品组合详细信息

在 AWS Service Catalog 管理员控制台中，产品组合详细信息页面列出了产品组合的设置。使用此页可以管理产品组合中的产品、授予用户对产品的访问权限以及应用 TagOptions 和限制。

查看 Portfolio details 页面

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择要管理的产品组合。

创建和删除产品组合

使用产品组合页面创建和删除产品组合。

创建新的产品组合

1. 从左侧导航菜单中，选择产品组合。
2. 选择创建产品组合。
3. 在创建产品组合页面输入所需信息。
4. 选择创建。AWS Service Catalog 创建产品组合并显示产品组合详细信息。

删除产品组合

> **Note**
>
> 您只能删除本地产品组合。您可以删除导入的（共享）产品组合，但不能删除导入的产品组合。

在删除投资组合之前，必须先移除其所有产品、约束、群组、角色、用户、共享和 TagOptions。为此，请打开一个产品组合以显示产品组合详细信息。然后选择选项卡将其删除。

> **Note**
>
> 为避免错误，请在删除任何产品之前从产品组合中删除约束。

1. 从左侧导航菜单中，选择产品组合。
2. 选择要删除的产品组合。
3. 选择删除。您只能删除本地产品组合。如果您正在尝试删除导入的（共享）产品组合，则操作菜单不可用。
4. 在确认窗口中，选择删除。

添加产品

您可以通过将新产品直接上传到现有产品组合或将目录中的现有产品与产品组合关联来将产品添加到产品组合。

> **Note**
>
> 创建 AWS Service Catalog 产品时，您可以上传 AWS CloudFormation 模板或 Terraform 配置文件。AWS CloudFormation 模板存储在 Amazon Simple Storage Service（Amazon S3）桶，且桶名称以“cf-templates-”开头。在预配置产品时，您还必须具有从其他存储桶中检索对象的权限。有关更多信息，请参阅创建产品。

添加新产品

您可以直接从产品组合详细信息页面添加新产品。在从该页面创建产品时，AWS Service Catalog 会将此产品添加到当前选定的产品组合。

添加新产品

1. 导航到产品组合页面，然后选择要将产品添加到的产品组合的名称。
2. 在产品组合详细信息页面上，展开产品部分，然后选择上传新产品。
3. 对于 Enter product details，输入以下内容：
   - 产品名称 - 产品的名称。
   - 产品描述（可选）- 产品描述。此描述显示在产品列表中，可帮助您选择正确的产品。
   - 描述 - 完整描述。此描述显示在产品列表中，可帮助您选择正确的产品。
   - 所有者或分销商 - 所有者的姓名或电子邮件地址。经销商的联系信息是可选的。
   - 供应商(可选) – 应用程序发布者的名称。利用此字段，您可以对其产品列表进行排序，以便更轻松地找到产品。
4. 在 Version details 页面上，输入以下内容：
   - 选择模板 — 对于 AWS CloudFormation 产品，请选择您自己的模板文件、本地驱动器中的AWS CloudFormation模板或指向 Amazon S3 中存储模板的 URL、现有的 AWS CloudFormation 堆栈 ARN 模板或存储在外部存储库中的模板文件。

     对于 Terraform 产品，请选择您自己的模板文件、本地驱动器中的 tar.gz 配置文件或指向 Amazon S3 中存储模板的 URL，或者存储在外部存储库中的 tar.gz 配置文件。
   - 版本名称（可选）– 产品版本的名称(例如，“v1”、“v2beta”)。不允许使用空格。
   - Description (可选) – 产品版本的说明，包括此版本与早期版本的区别。
5. 对于 Enter support details，输入以下内容：
   - Email contact (可选) – 用于报告与产品有关的问题的电子邮件地址。
   - 支持链接 (可选) – 用户可从中找到支持信息或文件票证的站点的 URL。URL 必须以 `http://` 或 `https://` 开头。管理员负责维护支持信息的准确性和可访问性。
   - 支持描述 (可选) – 有关用户应如何使用联系电子邮件和支持链接的描述。
6. 选择创建产品。

添加现有产品

您可以从三个位置将现有产品添加到产品组合：产品组合列表、产品组合详细信息页面或产品列表页面。

将现有产品添加到产品组合

1. 导航至产品组合页面。
2. 选择产品组合。然后选择操作 - 将产品添加到产品组合。
3. 选择产品，然后选择添加产品至产品组合。

从产品组合中删除产品

当您不再希望使用某个产品时，可将该产品从产品组合中删除。产品在产品页面的目录中仍然可用，并且您仍可将其添加到其他产品组合。可以一次性从产品组合中删除多个产品。

从产品组合中删除产品

1. 导航到产品组合页面，然后选择包含该产品的产品组合。产品组合详细信息页面随即打开。
2. 展开产品部分。
3. 选择一个或多个产品，然后选择删除。
4. 确认您的选择。

添加约束

您应该添加约束以控制用户与产品的互动方式。有关 AWS Service Catalog 支持的约束类型的更多信息，请参阅使用 AWS Service Catalog 约束。

可在产品放置到产品组合后向其添加约束。

将约束添加到产品

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择产品组合，然后选择一个产品组合。
3. 在产品组合详细信息页面上，展开创建约束部分，然后选择添加约束。
4. 对于产品，请选择要向其应用约束的产品。

5. 对于约束类型，选择以下选项之一：

   启动 — 允许您为用于预配置 AWS 资源的产品分配 IAM 角色。有关更多信息，请参阅 AWS Service Catalog 启动约束。

   通知 - 允许将产品通知流式传输到 Amazon SNS 主题。有关更多信息，请参阅 AWS Service Catalog 通知约束。

   模板 - 允许您限制最终用户在启动产品时可用的选项。模板包含 JSON 格式的文本文件，其中包含一个或多个规则。规则将添加到产品使用的 AWS CloudFormation 模板。有关更多信息，请参阅 模板约束规则。

   堆栈集-允许您使用配置跨账户和地区的产品部署AWS CloudFormation StackSets。有关更多信息，请参阅 AWS Service Catalog 堆栈集约束。

   标签更新 - 允许您在预配置产品后更新标签。有关更多信息，请参阅 AWS Service Catalog 标签更新约束。
6. 选择继续，然后输入所需信息。

编辑约束

1. 请登录 AWS Management Console 并通过以下网址打开 AWS Service Catalog 管理员控制台：https://console.aws.amazon.com/catalog/。
2. 选择产品组合，然后选择一个产品组合。
3. 在产品组合详细信息页面，展开创建约束部分，然后选择要编辑的约束。
4. 选择编辑约束。
5. 根据需要编辑约束，然后选择保存。

向用户授予访问权限

允许用户拥有通过组或角色对产品组合的访问权限。为多个用户提供对产品组合的访问权限的最佳方式是，将用户置于 IAM 用户组中并向该组授予访问权限。这样一来，您只需在组中添加或删除用户即可管理产品组合访问权限。有关更多信息，请参阅 IAM 用户指南 中的 IAM 用户和组。

除了对产品组合的访问权限外， 用户还必须具有对 AWS Service Catalog 最终用户控制台的访问权限。您可以通过在 IAM 中应用权限来授予对该控制台的访问权限。有关更多信息，请参阅 Identity and Access Management AWS Service Catalog。

如果您想与其他账户共享产品组合及其主体，可以将主体名称（群组、角色或用户）与投资组合相关联。主体名称与产品组合共享，并在收件人账户中用以向最终用户授予访问权限。

向用户或组授予产品组合访问权限

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在导航窗格中，选择管理，然后选择产品组合。
3. 选择要向群组、角色或用户授予访问权限的产品组合。AWS Service Catalog 定向至产品组合详细信息页面。
4. 在产品组合详细信息页面上，选择访问权限选项卡。
5. 在产品组合访问权限下，选择授予访问权限。
6. 对于类型，选择主体名称，然后选择组/、角色/或用户/类型。您最多可以添加 9 个主体名称。
7. 选择授予访问权限，关联主体与当前产品组合。

删除对产品组合的访问权限

1. 在产品组合详细信息页面上，选择组、角色和用户选项卡。
2. 然后选择删除访问权限。

共享产品组合

要使其他AWS账户的AWS Service Catalog管理员能够将您的产品分发给最终用户，请使用共享或与他们 account-to-account 共享您的产品AWS Service Catalog组合AWS Organizations。

当您使用共享或 Organizations account-to-account 共享作品集时，您就是在共享该作品集的引用。导入的产品组合中产品和约束与您对共享的产品组合（共享的原始产品组合）进行的更改保持同步。

收件人不能更改产品或约束，但可以为最终用户添加 AWS Identity and Access Management 访问权限。

> **Note**
>
> 您无法共享已共享的资源。这包括含有共享产品的产品组合。

A ccount-to-account 分享

要完成这些步骤，您必须获得目标 AWS 账户的账户 ID。您可以在目标账户 AWS Management Console 的我的账户页面上找到此 ID。

与 AWS 账户共享产品组合

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧导航菜单中，选择产品组合，然后选择要共享的产品组合。在操作菜单中，选择共享。
3. 在输入账户 ID 中，输入您要与之共享的 AWS 账户的账户 ID。（可选）选择TagOption 共享。然后选择共享。
4. 将 URL 发送到目标账户的 AWS Service Catalog 管理员。URL 将打开导入产品组合页面，并会自动提供共享产品组合的 ARN。

导入产品组合

如果其他 AWS 账户的 AWS Service Catalog 管理员与您共享了产品组合，将该产品组合导入到您的账户，以便将其产品分发给您的最终用户。

如果产品组合是通过 AWS Organizations 共享的，则无需导入产品组合。

要导入产品组合，您必须从管理员处获取产品 ID。

要查看所有导入的产品组合，请打开 AWS Service Catalog 控制台，网址为 https://console.aws.amazon.com/servicecatalog/。在产品组合页面上，选择已导入选项卡。查看导入的产品组合表。

与 AWS Organizations 共享

您可以使用 AWS Organizations 共享 AWS Service Catalog 产品组合。

首先，您必须决定是从管理账户还是从委托管理员账户进行共享。如果您不想从管理账户进行共享，请注册一个委托管理员账户并使用它进行共享。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的注册委托管理员。

接下来，您必须决定与谁共享。您可以与以下实体共享：

- 组织账户。
- 组织部门 (OU)。

- 组织本身。（这将与组织中的每个账户共享。）

从管理账户共享

当使用组织结构或输入组织节点的 ID 时，您可以与组织共享产品组合。

使用组织结构与组织共享产品组合

1. 打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicequotas/。
2. 在产品组合页面上，选择要共享的产品组合。在操作菜单中，选择共享。
3. 选择 AWS Organizations 并筛选到您的组织结构。

   您可以选择根节点与整个组织、父级组织单位 (OU)、子组织单位或 AWS 账户共享产品组合。

   共享给父级组织单位会将此产品组合共享给该父级组织单位中的所有账户和子组织单位。

   您可以选择仅查看 AWS 账户” 以查看组织中所有 AWS 账户的列表。

要通过输入组织节点的 ID与组织共享投资组合

1. 打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicequotas/。
2. 在产品组合页面上，选择要共享的产品组合。在操作菜单中，选择共享。
3. 选择组织节点。

   选择与整个组织、组织内的 AWS 账户或 OU 共享。

   输入您选择的组织节点的 ID，您可以在 AWS Organizations 控制台中找到此 ID：https://console.aws.amazon.com/organizations/。

从委托管理员账户共享

组织的管理账户可以将其他账户注册为组织的委托管理员，也可取消注册。

委托管理员可以像管理账户一样在其组织中共享 AWS Service Catalog 资源。他们有权创建、删除和共享产品组合。

要注册或取消注册委托管理员，您必须从管理账户中使用 API 或 CLI。有关更多信息，请参阅《AWS Organizations API 参考》中的 RegisterDelegatedAdministrator 和 DeregisterDelegatedAdministrator。

> **Note**
>
> 管理员必须先致电 `EnableAWSOrganizationsAccess`，然后才能指定委托 。

从委托管理员账户共享产品组合的过程与从管理账户共享相同，如 the section called “从管理账户共享” 中上述所示。

如果某个成员被取消注册为委托管理员，则会发生以下情况：

- 从该账户创建的产品组合共享将被删除。
- 他们不能再创建新的产品组合共享。

> **Note**
>
> 如果取消注册委托管理员后，委托管理员创建的产品组合和共享未被删除，请重新注册并取消注册委托管理员。此操作将删除由该账户创建的产品组合和共享。

在组织内移动账户

如果您在组织内移动账户，则与该账户共享的 AWS Service Catalog 产品组合可能会发生更改。

账户只能访问与其目标组织或组织单位共享的产品组合。

共享作品集 TagOptions 时共享

作为管理员，您可以创建要包含的共享 TagOptions。 TagOptions 是键值对，使管理员能够：

- 定义并强制执行标签分类法。
- 定义标签选项并将其与产品和产品组合相关联。
- 与其他账户共享与产品组合和产品关联的标签选项。

当您在主账户中添加或删除标签选项时，更改会自动显示在收件人账户中。在收款人账户中，当最终用户使用配置产品时 TagOptions，他们必须为标签选择值，这些标签将成为预配置产品上的标签。

在收款人账户中，管理员可以将其他本地账户关联 TagOptions 到其导入的投资组合，以强制执行特定于该账户的标记规则。

> **Note**
>
> 要共享投资组合，您需要消费端的 AWS 账户 ID。在控制台的我的账户中找到 AWS 账户 ID。

> **Note**
>
> 如果 a TagOption 只有一个值，则会在置备过程中AWS自动强制使用该值。

共享作品集 TagOptions 时共享

1. 从左侧导航菜单中，选择产品组合。
2. 在本地产品组合中，选择并打开产品组合。
3. 从上方的列表中选择共享，然后选择共享按钮。
4. 选择与其他 AWS 账户或组织共享。
5. 输入 12 位的账户 ID 号，选择启用，然后选择共享。

   您共享的账户显示在与之共享的账户部分中。它表示是否 TagOptions 已启用。

您也可以更新投资组合份额以包含在内 TagOptions。现在 TagOptions ，所有属于产品组合和产品的产品都将共享到该账户。

更新投资组合份额以包括 TagOptions

1. 从左侧导航菜单中，选择产品组合。
2. 在本地投资组合中，选择并打开投资组合。
3. 从上方的列表中选择共享。
4. 在与之共享的账户中，选择账户 ID，然后选择操作。
5. 选择更新取消共享或取消共享。

   选择 “更新取消共享” 时，选择 “启用” 以启动共享 TagOptions。您共享的账户显示在与之共享的账户部分中。

   选择取消共享时，请确认您不想再共享该账户。

共享产品组合时共享主体名称

作为管理员，您可以创建包含主体名称的产品组合共享。主体名称是群组、角色和用户的名称，管理员可以在产品组合中指定这些名称，然后与产品组合共享。当您共享产品组合时，AWS Service Catalog 会验证这些主体名称是否已经存在。如果确实存在，则 AWS Service Catalog 自动关联匹配的 IAM 主体和共享产品组合以向用户授予访问权限。

> **Note**
>
> 当您将主体与产品组合相关联时，当该产品组合随后与其他账户共享时，可能会出现潜在的权限提升路径。对于收件人账户中不是 AWS Service Catalog 管理员但仍能创建主体（用户/角色）的用户，该用户可以创建与产品组合的主体名称关联相匹配的 IAM 主体。尽管此用户可能不知道通过 AWS Service Catalog 关联了哪些主体名称，但他们可以猜出用户。如果这种潜在的提升路径是一个问题，则 AWS Service Catalog 建议使用 `PrincipalType` 作为 `IAM`。使用此配置，`PrincipalARN` 必须先存在于收件人账户中，然后才能将其关联。

当您在主账户中添加或删除主体名称时，AWS Service Catalog 会自动将这些更改应用到收件人账户中。然后，收件人账户中的用户可以根据其角色执行任务：

- 最终用户可以预配置、更新和终止产品组合的产品。
- 管理员可以将其他 IAM 主体关联到其导入的产品组合，以向特定于该账户的最终用户授予访问权限。

> **Note**
>
> 主体名称共享仅对 AWS Organizations 可用。

要在共享产品组合时共享主体名称

1. 从左侧导航菜单中，选择产品组合。
2. 在本地产品组合中，选择要共享的产品组合。
3. 在操作菜单中，选择共享。
4. 在 AWS Organizations 中选择一个组织。
5. 选择整个组织根目录、组织单位 (OU) 或组织成员。
6. 在共享设置中，启用主体共享选项。

您也可以更新产品组合共享，使其包含主体名称共享。这会与收件人账户共享属于该产品组合的所有主体名称。

要更新投资组合共享以启用或禁用主体名称

1. 从左侧导航菜单中，选择产品组合。
2. 在本地产品组合中，选择要更新的产品组合。
3. 选择共享 选项卡。
4. 选择要更新的共享，然后选择共享。
5. 选择 更新共享，然后选择启用以启动中体共享。然后 AWS Service Catalog 会在收件人账户中共享主体名称。

如果您想停止与收件人账户共享主体名称，请禁用委托人共享。

在共享主体名称时使用通配符

AWS Service Catalog 支持使用通配符（例如 “*” 或 “?”）向 IAM 主体（用户、群组或角色）名称授予产品组合访问权限。使用通配符模式可以同时覆盖多个 IAM 主体名称。ARN 路径和主体名称允许使用无限制通配符。

可接受的通配符 ARN 示例：

- **`arn:aws:iam:::role/ResourceName_*`**
- **`arn:aws:iam:::role/*/ResourceName_?`**

不可接受的通配符 ARN 示例：

- **`arn:aws:iam:::*/ResourceName`**

在 IAM 主体 ARN 格式 (**`arn:partition:iam:::resource-type/resource-path/resource-name`**) 中，有效值包括用户/、组/或角色/。“?”和“*”只能在 resource-id 段中的 resource-type 后使用。您可以在 resource-id 中的任何位置使用特殊字符。

“*”字符还与“/”字符匹配，从而允许在 resource-id 中形成路径。例如：

`arn:aws:iam:::role/*/ResourceName_?` 匹配 **`arn:aws:iam:::role/pathA/pathB/ResourceName_1`** 和 **`arn:aws:iam:::role/pathA/ResourceName_1`**。

共享和导入产品组合

要使 AWS Service Catalog 产品对不在您的 AWS 账户 中的用户可用，例如，其他组织中的用户或您组织中的其他 AWS 账户 中的用户，您可与其共享产品组合。您可以通过多种方式进行共享，包括 account-to-account 共享、组织共享和使用堆栈集部署目录。

在将产品和产品组合与其他账户共享之前，您必须决定是要共享目录的引用，还是要将目录的副本部署到每个收件人账户。请注意，如果部署副本，则当有要传播到收件人账户的更新时，必须重新部署。

您可以使用堆栈集同时将目录部署到多个账户。如果您想共享参考文献（与原始作品集保持同步的导入版本），则可以使用 account-to-account 共享或使用进行共享AWS Organizations。

要使用堆栈集部署目录的副本，请参阅如何设置公司标准 AWS Service Catalog 产品的多区域、多账户目录。

当您使用共享或 account-to-account 共享产品组合时AWS Organizations，您允许其他AWS账户的 AWS Service Catalog管理员将您的产品组合导入他们的账户，并将产品分发给该账户中的最终用户。

此导入的产品组合 不是独立副本。导入的产品组合中产品和约束与您对共享的产品组合（共享的原始产品组合）进行的更改保持同步。收件人管理员，即您与之共享产品组合的管理员，无法更改产品或约束，但可以为最终用户添加 AWS Identity and Access Management (IAM) 访问权限。有关更多信息，请参阅 向用户授予访问权限。

收件人管理员可通过以下方式将产品分发给属于其 AWS 账户的最终用户：

- 将用户、组或角色添加到导入的产品组合。
- 将导入的产品组合中的产品添加到本地产品组合（收件人管理员创建的、属于其 AWS 账户的独立产品组合）。然后，收件人管理员将用户、组和角色添加到本地产品组合。对共享的产品组合中的产品原本应用的任何约束也存在于本地产品组合中。本地产品组合收件人管理员可添加其他约束，但无法删除原本自共享产品组合中导入的约束。

当您在共享的产品组合中添加或删除产品或约束时，此更改会传播到该产品组合的所有导入实例。例如，如果从共享的产品组合中删除产品，则也会从导入的产品组合中删除该产品。还会从导入的产品所添加到的所有本地产品组合中删除该产品。如果最终用户在您删除产品之前已启动产品，则最终用户的预配置产品会继续运行，但该产品在将来无法启动。

如果将启动约束应用于共享的产品组合中的产品，则该启动约束会传播到该产品的所有导入实例。要覆盖此启动约束，收件人管理员需将产品添加到本地产品组合中，然后将不同的启动约束应用于该本地产品组合。生效的启动约束将设置产品的启动角色。

启动角色 是 AWS Service Catalog 在最终用户启动产品时用来配置 AWS 资源（如 Amazon EC2 实例或 Amazon RDS 数据库）的 IAM 角色。作为管理员，您可以选择指定特定启动角色 ARN 或本地角色名称。如果您使用角色 ARN，即使最终用户不属于拥有启动角色的 AWS 账户，也使用此启动角色。如果您使用本地角色名称，则将使用最终用户的账户中具有该名称的 IAM 角色。

有关启动约束和启动角色的更多信息，请参阅AWS Service Catalog 启动约束。拥有启动角色的 AWS 账户将配置 AWS 资源，并会产生这些资源的使用费。有关更多信息，请参阅AWS Service Catalog 定价。

该视频向您展示了如何在 AWS Service Catalog 中跨账户共享产品组合。

分享 (https://www.youtube.com/embed/BVSohYOppjk%22%3EShare) 在 AWS Service Catalog 中跨账户的产品组合。

> **Note**
>
> 您无法重新共享已导入或已共享的产品组合中的产品。

> **Note**
>
> 产品组合导入必须发生在管理账户和相关账户间的同一区域。

共享的产品组合和导入的产品组合之间的关系

此表汇总了导入的产品组合和共享的产品组合之间的关系，以及导入产品组合的管理员对该产品组合和其中的产品能执行和不能执行的操作。

共享的产品组合的元素	与导入的产品组合的关系	收件人管理员能够	收件人管理员不能
产品和产品版本	已继承。 如果产品组合创建者在共享的产品组合中添加或删除产品，则此更改会传播到导入的产品组合。	将导入的产品添加到本地产品组合。产品与共享的产品组合保持同步。	将产品上传或添加到导入的产品组合中，或从导入的产品组合中删除产品。

共享的产品组合的元素	与导入的产品组合的关系	收件人管理员能够	收件人管理员不能
启动约束	已继承。 如果产品组合创建者在共享的产品中添加或删除启动约束，则更改会传播到该产品的所有导入实例。 如果收件人管理员将导入的产品添加到本地产品组合，则导入的启动约束不会存在于共享产品组合中。	在本地产品组合中，管理员可以应用影响产品本地启动的启动限制。	在导入的产品组合中添加或删除启动约束。
模板约束	已继承。 如果产品组合创建者在共享的产品中添加或删除模板约束，则更改会传播到该产品的所有导入实例。 如果收件人管理员将导入的产品添加到本地产品组合，则导入的模板约束将不会应用于本地产品组合。	在本地产品组合中，管理员可以添加约束本地产品的模板约束。	删除导入的模板约束。
用户、组和角色	未继承。	添加管理员 AWS 账户中的用户、组和角色。	不适用。

管理产品

您可以通过创建基于更新后的模板的新版本来创建、更新产品，并将产品成组添加到产品组合中以便将其分发给用户。

产品的新版本将传播到有权通过产品组合访问产品的所有用户。在您分发更新时，最终用户可以更新现有预配置产品。

任务

- 查看产品页面
- 创建产品
- 将产品添加到产品组合
- 更新产品
- 将产品同步到 GitHub、 GitHub 企业版或 Bitbucket 中的模板文件
- 删除产品
- 管理版本

查看产品页面

您可以从 AWS Service Catalog 管理员控制台中的产品列表页面管理产品。

要查看产品列表页面

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择产品列表。

创建产品

您可以从 AWS Service Catalog 管理员控制台中的产品页面管理产品。

> **Note**
>
> 创建 Terraform 产品需要额外配置，包括 Terraform 预置引擎和启动角色。有关更多信息，请查看 开始使用 Terraform 产品。

创建新的 AWS Service Catalog 产品

1. 导航产品列表页面。
2. 选择创建产品，然后选择创建产品。
3. 产品详细信息 - 允许您选择要创建的产品类型。AWS Service Catalog 支持 AWS CloudFormation、Terraform 云和“外部”（支持 Terraform 社区版）产品类型。产品详细信息还包含您在列表或详细信息页面中搜索和查看产品时显示的元数据。输入以下信息：
   - 产品名称 - 产品的名称。
   - 产品描述 — 描述显示在产品列表中，可帮助您选择正确的产品。
   - 所有者 - 发布此产品的个人或组织。所有者可以是您 IT 组织或管理员的名称。
   - 分销商(可选) – 应用程序发布者的名称。利用此字段，您可以对其产品列表进行排序，以便更轻松地找到产品。
4. 版本详细信息使您可以添加模板文件并构建产品。输入以下信息：
   - 选择方法 - 有四种方法可以添加模板文件。
     - 使用本地模板文件 - 上传本地驱动器中的 AWS CloudFormation 模板或 Terraform tar.gz 配置文件。
     - 使用 Amazon S3 URL - 指定 URL，该 URL 指向 AWS CloudFormation 模板或存储在 Amazon S3 中的 Terraform tar.gz 配置文件。如果指定的是 Amazon S3 URL，则它必须以 `https://` 开头。
     - 使用外部存储库-指定您的 GitHub、 GitHub 企业版或 Bitbucket 代码存储库。 AWS Service Catalog允许您将产品同步到模板文件。对于 Terraform 产品，模板文件格式必须是在 Tar 中存档并在 Gzip 中压缩的单个文件。
     - 使用现有 CloudFormation 堆栈-输入现有 CloudFormation 堆栈的 ARN。此方法不支持 Terraform 云或外部产品。
   - 版本名称（可选）– 产品版本的名称(例如，“v1”、“v2beta”)。不允许使用空格。
   - 描述(可选) – 产品版本的描述，包括此版本与其他版本的区别。
   - 指南 - 在产品详细信息页面的版本选项卡中进行管理。创建产品版本时（在创建产品工作流程中），该版本的指南设置为默认值。要了解有关指南的更多信息，请参阅管理版本。
5. 支持详细信息可确定贵公司内部的组织，并提供支持联系人。输入以下信息：
   - Email contact (可选) – 用于报告与产品有关的问题的电子邮件地址。
   - 支持链接 (可选) – 用户可从中找到支持信息或文件票证的站点的 URL。URL 必须以 `http://` 或 `https://` 开头。管理员负责维护支持信息的准确性和可访问性。

- 支持描述 (可选) – 有关用户应如何使用联系电子邮件和支持链接的描述。

6. 管理标签（可选）— 除了使用标签对资源进行分类外，您还可以使用标签来验证您创建此资源的权限。
7. 创建产品 - 填写完表格后，选择创建产品。几秒钟后，产品会显示在产品列表页面上。您可能需要刷新浏览器来查看产品。

您还可以使用 CodePipeline 创建和配置管道，将产品模板部署到源存储库AWS Service Catalog并交付您在源存储库中所做的更改。有关更多信息，请参阅教程：创建部署到 AWS Service Catalog 的管道。

您可以在 AWS CloudFormation 或 Terraform 模板中定义参数属性，并在预配置期间强制执行这些规则。这些属性可以定义最小和最大长度、最小值和最大值、允许的值以及值的正则表达式。如果提供的值不符合参数属性，则 AWS Service Catalog 会在预配置期间发出警告。要了解有关参数属性的更多信息，请参阅《AWS CloudFormation 用户指南》中的参数。

故障排除

您必须拥有从 Amazon S3 存储桶中检索对象的权限。否则，启动或更新产品时，您可能会遇到以下错误。

```
Error: failed to process product version s3 access denied exception
```

如果您遇到此消息，请确保您拥有从以下存储桶中检索对象的权限：

- 存储桶储存着预配置构件模板。
- 存储桶以“cf-templates-*”开头，AWS Service Catalog 将其用于存储预配置构件模板。
- 内部存储桶以“sc-*”开头，AWS Service Catalog 将其用于存储元数据。您将无法在您的账户中看到此存储桶。

以下示例策略展示了在前述存储桶检索对象所需的最低权限。

```
{
          "Sid": "VisualEditor1",
          "Effect": "Allow",
          "Action": "s3:GetObject*",
          "Resource": [
              "arn:aws:s3:::YOUR_TEMPLATE_BUCKET",
```

```
            "arn:aws:s3:::YOUR_TEMPLATE_BUCKET/*",
            "arn:aws:s3:::cf-templates-*",
            "arn:aws:s3:::cf-templates-*/*",
            "arn:aws:s3:::sc-*",
            "arn:aws:s3:::sc-*/*"
        ]
    }
```

将产品添加到产品组合

您可以将产品添加到任意数量的产品组合。在产品更新时，所有包含产品的产品组合（包括共享产品组合）会自动接收新版本。

将产品从目录添加到产品组合

1. 导航产品列表页面。
2. 选择产品，然后选择操作。从下拉菜单中，选择将产品添加到产品组合。您将被定向到 “***name-of-product***添加到投资组合” 页面。
3. 选择产品组合，然后选择添加产品至产品组合。

当将 Terraform 产品添加到产品组合中时，该产品需要启动约束。您必须从账户中选择一个 IAM 角色，输入 IAM 角色 ARN 或输入角色名称。如果您指定角色名称，当账户使用启动约束时，将使用账户中具有该名称的 IAM 角色。这允许使用与账户无关的启动角色约束，确保您可以为每个共享账户创建更少的资源。有关详情和说明，请查看 步骤 6：为 Terraform 产品添加启动约束

一个产品组合可以包含多种产品，这些产品是 AWS CloudFormation 和 Terraform 产品类型的混合。

更新产品

当更新产品的模板时，您会创建产品的新版本。新产品版本将自动对有权访问包含此产品产品组合的所有用户可用。

> **Note**
>
> 更新现有产品时，您无法更改产品类型（AWS CloudFormation 或 Terraform）。例如，如果您更新的是 AWS CloudFormation 产品，则无法用 Terraform tar.gz 配置文件替换现有 AWS CloudFormation 模板。您必须使用新的 AWS CloudFormation 模板文件更新现有的 AWS CloudFormation 模板文件。

当前正在运行早期版本预配置产品的最终用户可将其预配置产品更新至新版本。当产品有新版本可用时，用户可以使用预配置产品列表或预配置产品详细信息页面上的更新预配置产品命令。

在创建产品的新版本之前，AWS Service Catalog 建议您测试 AWS CloudFormation 中或 Terraform 引擎中的产品更新以确保它们正常工作。

创建新产品版本

1. 导航产品页面。
2. 选择要更新的产品。您将被定向至产品详细信息页面。
3. 在产品详细信息页面上，展开版本选项卡，然后选择创建新版本。
4. 在版本详细信息下，执行以下操作：
   - 选择模板 - 有四种方法可以添加模板文件。

     使用本地模板文件 - 上传本地驱动器中的 AWS CloudFormation 模板或 Terraform tar.gz 配置文件。

     使用 Amazon S3 URL - 指定 URL，该 URL 指向 AWS CloudFormation 模板或存储在 Amazon S3 中的 Terraform tar.gz 配置文件。如果指定的是 Amazon S3 URL，则它必须以 https:// 开头。

     使用外部存储库-指定您的 GitHub、 GitHub 企业版或 Bitbucket 代码存储库。 AWS Service Catalog允许您将产品同步到模板文件。对于 Terraform 产品，模板文件格式必须是在 Tar 中存档并在 Gzip 中压缩的单个文件。

     使用现有 CloudFormation 堆栈-输入现有 CloudFormation 堆栈的 ARN。此方法不支持 Terraform 云或外部产品。
   - 版本标题 – 产品版本的名称(例如，“v1”、“v2beta”)。不允许使用空格。
   - 描述(可选) – 产品版本的描述，包括此版本与早期版本的区别。
5. 选择创建产品版本。

您还可以使用 CodePipeline 创建和配置管道，将产品模板部署到源存储库中AWS Service Catalog，并在源存储库中交付更改。有关更多信息，请参阅教程：创建部署到 AWS Service Catalog 的管道。

将产品同步到 GitHub、 GitHub 企业版或 Bitbucket 中的模板文件

AWS Service Catalog 允许您将产品同步到通过外部存储库提供商管理的模板文件。 AWS Service Catalog 将具有此类模板连接的产品称为 Git 同步产品。存储库选项包括 GitHub “ GitHub 企业版” 或 “Bitbucket”。 AWS 账户 使用外部存储库帐户授权后，您可以创建新 AWS Service Catalog 产品或更新现有产品以同步到存储库中的模板文件。当对模板文件进行更改并提交到存储库中时（例如，使用 git-push），AWS Service Catalog 会自动检测更改并创建新的产品版本（构件）。

主题

- 将产品同步至外部模板文件所需权限
- 创建账户连接
- 查看 Git 同步产品连接
- 更新 Git 同步产品连接
- 删除 Git 同步产品连接
- 将 Terraform 产品同步到来自 GitHub、Enterpr GitHub ise 或 Bitbucket 的模板文件
- AWS 区域 支持 Git 同步产品

将产品同步至外部模板文件所需权限

您可以使用以下 AWS Identity and Access Management (IAM) 策略作为模板，使 AWS Service Catalog 管理员能够将产品同步到外部存储库中的模板文件。此策略包括 CodeConnections 和的必需权限 AWS Service Catalog。 AWS Service Catalog 建议您复制下面的模板策略，并在启用存储库同步产品时也使用 AWS Service Catalog `AWSServiceCatalogAdminFullAccess`托管策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CodeStarAccess",
            "Effect": "Allow",
            "Action": [
            "codestar-connections:UseConnection",
            "codestar-connections:PassConnection",
            "codestar-connections:CreateConnection",
            "codestar-connections:DeleteConnection",
            "codestar-connections:GetConnection",
            "codestar-connections:ListConnections",
            "codestar-connections:ListInstallationTargets",
```

```
            "codestar-connections:GetInstallationUrl",
            "codestar-connections:StartOAuthHandshake",
            "codestar-connections:UpdateConnectionInstallation",
            "codestar-connections:GetIndividualAccessToken"
            ],
            "Resource": "arn:aws:codestar-connections:*:*:connection/*"
        },
        {
        "Sid": "CreateSLR",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/
sync.servicecatalog.amazonaws.com/AWSServiceRoleForServiceCatalogArtifactSync",
        "Condition": {
        "StringLike": {
        "iam:AWSServiceName": "sync.servicecatalog.amazonaws.com"
                }
            }
        }
    ]
}
```

创建账户连接

在将模板文件同步到 AWS Service Catalog 产品之前，必须创建并授权一次性 account-to-account 连接。您可以使用此连接来指定包含所需模板文件的存储库的详细信息。您可以使用 AWS Service Catalog 控制台、控制 CodeConnections 台、 AWS Command Line Interface (CLI) 或 CodeConnections API 创建连接。

建立连接后，您可以使用 AWS Service Catalog 控制台、 AWS Service Catalog API 或 CLI 来创建同步 AWS Service Catalog 产品。 AWS Service Catalog 管理员可以根据存储库和分支中的模板文件创建新 AWS Service Catalog 产品或更新现有产品。如果在存储库中提交了更改，则 AWS Service Catalog 会自动检测更改并创建新的产品版本。先前产品版本仍然不可超过规定的版本限制，并被指定为已弃用状态。

此外，在创建连接后 AWS Service Catalog 自动创建服务相关角色 (SLR)。此 SLR 允许 AWS Service Catalog 检测提交到存储库的任何模板文件更改。单反相机还 AWS Service Catalog 允许自动为同步产品创建新的产品版本。有关 SLR 权限和功能的更多信息，请参阅 AWS Service Catalog的服务相关角色。

要创建新的 Git 同步产品

1. 在左侧导航面板中，选择产品列表，然后选择创建产品。
2. 输入产品详细信息。
3. 在 “版本详细信息” 中，选择 “使用 AWS CodeStar 提供程序指定您的代码存储库”，然后选择 “创建新 AWS CodeStar 连接” 链接。
4. 创建连接后，刷新连接列表，然后选择新连接。指定存储库详细信息，包括存储库、分支和模板文件路径。

   有关使用 Terraform 配置文件的更多信息，请参阅 将 Terraform 产品同步到来自 GitHub、Enterpr GitHub ise 或 Bitbucket 的模板文件 。

   a. （创建新 AWS Service Catalog 产品资源时可选）在 Suppor t Det ails 部分，添加产品的元数据。
   b. （创建新 AWS Service Catalog 产品资源时可选）在 “标签” 部分，选择 “添加新标签”，然后输入 “密钥” 和 “值” 对。
5. 选择创建新产品。

要创建多个 Git 同步产品

1. 在 AWS Service Catalog 控制台左侧导航面板中，选择产品列表，然后选择创建多个 git 托管产品。
2. 输入通用产品详细信息。
3. 在外部存储库详细信息中，选择一个 AWS CodeStar 连接，然后指定存储库和分支。
4. 在“添加产品”窗格中，输入模板文件路径和产品名称。选择添加新项目，然后根据需要继续添加产品。
5. 添加所有所需产品后，选择批量创建产品。

将现有 AWS Service Catalog 产品连接到外部存储库

1. 在 AWS Service Catalog 控制台左侧导航面板中，选择 “产品列表”，然后选择 “将产品连接到外部存储库”。
2. 在选择产品页面上，选择要连接到外部存储库的产品，然后选择下一步。
3. 在 “指定源详细信息” 页面上，选择现有 AWS CodeStar 连接，然后指定存储库、分支和模板文件路径。

4. 选择下一步。
5. 在查看并提交页面上，验证连接详细信息，然后选择将产品连接到外部存储库。

查看 Git 同步产品连接

您可以使用 AWS Service Catalog 控制台、API 或 AWS CLI 来查看存储库连接的详细信息。对于链接到模板文件的 AWS Service Catalog 产品，您可以从 “上次同步状态” 中检索有关存储库连接以及模板上次与产品同步的时间的信息。

> **Note**
>
> 您可以在产品级别查看存储库信息和上次同步状态。用户必须拥有 CodeConnections API 中的 IAM 权限才能查看存储库的详细信息。有关这些 IAM 权限所需策略的更多信息，请参阅将 AWS Service Catalog 产品同步到模板文件所需的权限。

要查看连接和存储库的详细信息，请使用 AWS Management Console

1. 从左侧导航面板中，选择产品列表。
2. 从列表中选择产品。
3. 在产品页面上，导航至产品源详细信息部分。
4. 要查看产品版本的源代码修订 ID，请选择上次创建的版本链接。版本详细信息部分显示源代码修订 ID。

要查看连接和存储库的详细信息，请使用 AWS CLI

从中 AWS CLI，运行以下命令：

```
$ aws servicecatalog describe-product-as-admin

$ aws servicecatalog describe-provisioning-artifact

$ aws servicecatalog search-product-as-admin

$ aws servicecatalog list-provisioning-artifacts
```

更新 Git 同步产品连接

您可以使用 AWS Service Catalog 控制台、 AWS Service Catalog API 或更新现有账户连接和 Git 同步产品。 AWS CLI

要了解如何将现有 AWS Service Catalog 产品连接到模板文件，请参阅创建新的 Git 同步产品连接。

要将现有产品更新为 Git 同步产品

1. 在左侧的导航面板中，选择产品列表，然后选择以下选项之一：
   - 要更新单个产品，请选择该产品，导航至产品源详细信息部分，然后选择编辑详细信息。
   - 要更新多个产品，请选择将产品连接到外部存储库，选择最多十个产品，然后选择下一步。
2. 在产品源详细信息部分，执行以下更新：
   - 指定连接。
   - 指定存储库。
   - 指定分支。
   - 命名模板文件。
3. 选择保存更改。

> **Note**
>
> 对于尚未连接到外部存储库的产品，您可以在选择产品后使用产品信息页面顶部的提醒中显示的连接到外部存储库选项。

你也可以使用 AWS Service Catalog 控制台 AWS CLI 或

- 将现有 AWS Service Catalog 产品连接到外部存储库中的模板文件
- 更新产品元数据，包括产品名称、描述和标签。
- 重新配置（更新同步以使用其他存储库来源）先前连接 AWS Service Catalog 产品的连接。

使用 AWS Service Catalog 控制台更新连接和存储库详细信息

1. 在 AWS Service Catalog 控制台左侧导航面板中，选择产品列表，然后选择当前连接到外部存储库的产品。

2. 在产品源详细信息部分，选择编辑产品源。
3. 在产品源详细信息部分，指定新的所需存储库。
4. 选择 保存更改。

要更新连接和存储库的详细信息，请使用 AWS CLI

从中 AWS CLI 运行`$ aws servicecatalog update-product`和`$ aws servicecatalog update-provisioning-artifact`命令。

删除 Git 同步产品连接

您可以使用 AWS Service Catalog 控制台、 CodeConnections API 或删除 AWS Service Catalog 产品与模板文件之间的连接 AWS CLI。当您断开产品与模板文件的连接时，已同步 AWS Service Catalog 的产品会切换到常规管理的产品。断开产品连接后，如果模板文件更改并提交到先前连接的存储库中，则这些更改不会反映出来。要将 AWS Service Catalog 产品重新连接到外部存储库中的模板文件，请参阅更新连接和同步 AWS Service Catalog产品。

使用控制台断开 Git 同步产品的连接 AWS Service Catalog

1. 在中 AWS Management Console，从左侧导航面板中选择产品列表。
2. 从列表中选择产品。
3. 在产品页面上，导航至产品源详细信息部分。
4. 选择断开连接。
5. 确认操作，然后选择断开连接。

要断开 Git 同步产品的连接，请使用 AWS CLI

从中 AWS CLI，运行`$ aws servicecatalog update-product`命令。在`ConnectionParameters` 输入中，移除指定的连接。

要使用 CodeConnections API 删除连接或 AWS CLI

在 CodeConnections API 或中 AWS CLI，运行`$ aws codestar-connections delete-connection`命令。

将 Terraform 产品同步到来自 GitHub、Enterpr GitHub ise 或 Bitbucket 的模板文件

使用 Terraform 配置文件创建 Git 同步产品时，文件路径仅接受 tar.gz 格式。在文件路径中不能接受 Terraform 文件夹格式。

AWS 区域 支持 Git 同步产品

AWS Service Catalog 支持中 AWS 区域 与 Git 同步的产品，如下表所示。

AWS 区域 名字	AWS 区域 身份	支持与 Git 同步的产品
美国东部（弗吉尼亚州北部）	us-east-1	支持
美国东部（俄亥俄州）	us-east-2	支持
美国西部（北加利福尼亚）	us-west-1	支持
美国西部（俄勒冈州）	us-west-2	支持
非洲（开普敦）	af-south-1	不支持
亚太地区（香港）	ap-east-1	不支持
亚太地区（雅加达）	ap-southeast-3	不支持
亚太地区（孟买）	ap-south-1	支持
亚太地区（大阪）	ap-northeast-3	不支持
亚太地区（首尔）	ap-northeast-2	支持
亚太地区（新加坡）	ap-southeast-1	支持
亚太地区（悉尼）	ap-southeast-2	支持
亚太地区（东京)	ap-northeast-1	支持
加拿大（中部）	ca-central-1	支持
欧洲地区（法兰克福）	eu-central-1	支持
欧洲地区（爱尔兰）	eu-west-1	支持
欧洲地区（伦敦）	eu-west-2	支持
欧洲地区（米兰）	eu-south-1	不支持

AWS 区域 名字	AWS 区域 身份	支持与 Git 同步的产品
欧洲地区（巴黎）	eu-west-3	支持
欧洲地区（斯德哥尔摩）	eu-north-1	支持
中东（巴林）	me-south-1	不支持
South America（São Paulo）	sa-east-1	支持
AWS GovCloud （美国东部）	us-gov-east-1	不支持
AWS GovCloud （美国西部）	us-gov-west-1	不支持

删除产品

您删除产品时，AWS Service Catalog 会从每个包含该产品的产品组合中删除该产品的所有版本。

AWS Service Catalog 允许您使用 AWS Service Catalog 控制台或 AWS CLI 删除产品。要成功删除产品，您必须先取消关联该产品的所有资源。产品资源关联的示例包括产品组合关联 TagOptions、预算和服务操作。

> **Important**
>
> 删除产品后，您无法将其恢复。

要使用 AWS Service Catalog 控制台删除产品

1. 导航至产品组合页面，然后选择包含要删除的产品的产品组合。
2. 选择要删除的产品，然后选择产品窗格右上角的删除。
3. 对于没有关联资源的产品，请在文本框中输入删除，确认要删除的产品，然后选择删除。

   对于关联资源的产品，请继续执行步骤 4。
4. 在删除产品窗口中，查看关联表，该表显示了该产品的所有关联资源。当您删除产品时，AWS Service Catalog 会尝试解除这些资源的关联。
5. 在文本框中输入删除，确认您要删除该产品并删除其所有关联资源。
6. 选择取消关联并删除。

如果 AWS Service Catalog 无法解除所有产品资源关联，则不会删除该产品。删除产品窗口会显示取消关联的失败次数以及每个失败的描述。有关在删除产品时解决资源取消关联失败问题的更多信息，请参阅下方的删除产品时解决资源取消关联失败的问题。

主题

- 使用 AWS CLI 删除产品
- 解决删除产品时资源取消关联失败的问题

使用 AWS CLI 删除产品

AWS Service Catalog 允许您使用 AWS Command Line Interface(AWS CLI) 从您的产品组合中删除产品。AWS CLI 是一种开源工具，让您能够在命令行 Shell 中使用命令与 AWS 服务进行交互。AWS Service Catalog 的强制删除功能需要 AWS CLI 别名，这是您可以在 AWS CLI 中创建的、用于缩短您常用的命令和脚本的快捷方式。

先决条件

- 安装和配置 AWS CLI。有关更多信息，请参阅安装或更新最新版本的 AWS CLI 和配置基础知识 。使用最低 AWS CLI 版本 1.11.24 或 2.0.0。
- 删除产品 CLI 别名需要与 bash 兼容的终端和 JQ 命令行 JSON 处理器。有关安装命令行 JSON 处理器的更多信息，请参阅下载 jq。
- 借助创建 AWS CLI 别名以批处理 `Disassociation` API 调用，您能够使用单个命令删除产品。

要成功删除产品，您必须先取消关联该产品的所有资源。产品资源关联的示例包括产品组合关联、预算、标签选项和服务操作。使用 CLI 删除产品时，CLI `force-delete-product` 别名允许您调用 `Disassociate` API 来取消关联任何会阻止 `DeleteProduct` API 的资源。这样可以避免为单独解除关联进行独立调用。

> **Note**
>
> 以下步骤中显示的文件路径可能会有所不同，具体取决于您用来执行这些操作的操作系统。

创建 AWS CLI 别名以删除 AWS Service Catalog 产品

使用 AWS CLI 删除 AWS Service Catalog 产品时，CLI `force-delete-product` 别名允许您调用 `Disassociate` API 来取消关联任何会阻止 `DeleteProduct` 调用的资源。

在 AWS CLI 配置文件夹中创建 **alias** 文件

1. 在 AWS CLI 控制台中，导航到配置文件夹。默认情况下，配置文件夹路径为 `~/.aws/`（Linux 或 macOS）或 `%USERPROFILE%\.aws\` (Windows)。
2. 使用文件导航或在首选终端中输入以下命令来创建名为 `cli` 的子文件夹：

```
$ mkdir -p ~/.aws/cli
```

   生成的 `cli` 文件夹默认路径为 `~/.aws/cli/`（Linux 或 macOS）或 `%USERPROFILE%\.aws\cli` (Windows)。
3. 在新建的 `cli` 文件夹中，创建不带扩展名的名为 `alias` 的文本文件。您可以使用文件导航或在首选终端中输入以下命令来创建 `alias` 文件：

```
$ touch ~/.aws/cli/alias
```

4. 在第一行输入 `[toplevel]`。
5. 保存该文件。

接下来，您可以通过手动将 force-delete-product 别名脚本粘贴到`alias`文件中，或者在终端窗口中使用命令来将别名添加到文件中。

手动将 force-delete-product 别名添加到您的**alias**文件中

1. 在 AWS CLI 控制台中，导航到您的 AWS CLI 配置文件夹并打开 `alias` 文件。
2. 在文件的 `[toplevel]` 行下方输入以下代码别名：

```
[command servicecatalog]
force-delete-product =
  !f() {
    if [ "$#" -ne 1 ]; then
        echo "Illegal number of parameters"
        exit 1
    fi
```

```
                if [[ "$1" != prod-* ]]; then
                   echo "Please provide a valid product id."
                   exit 1
                fi

                productId=$1
                describeProductAsAdminResponse=$(aws servicecatalog describe-
product-as-admin --id $productId)
                listPortfoliosForProductResponse=$(aws servicecatalog list-
portfolios-for-product --product-id $productId)

                tagOptions=$(echo "$describeProductAsAdminResponse" | jq -r
 '.TagOptions[].Id')
                budgetName=$(echo "$describeProductAsAdminResponse" | jq -r
 '.Budgets[].BudgetName')
                portfolios=$(echo "$listPortfoliosForProductResponse" | jq -r
 '.PortfolioDetails[].Id')
                provisioningArtifacts=$(echo "$describeProductAsAdminResponse" | jq
 -r '.ProvisioningArtifactSummaries[].Id')
                provisioningArtifactServiceActionAssociations=()

                for provisioningArtifactId in $provisioningArtifacts; do
                  listServiceActionsForProvisioningArtifactResponse=$(aws
 servicecatalog list-service-actions-for-provisioning-artifact --product-id
 $productId --provisioning-artifact-id $provisioningArtifactId)
                  serviceActions=$(echo
 "$listServiceActionsForProvisioningArtifactResponse" | jq -r
 '[.ServiceActionSummaries[].Id] | join(",")')
                  if [[ -n "$serviceActions" ]]; then
                    provisioningArtifactServiceActionAssociations
+=("${provisioningArtifactId}:${serviceActions}")
                  fi
                done

                echo "Before deleting a product, the following associated resources
 must be disassociated. These resources will not be deleted. This action may take
 some time, depending on the number of resources being disassociated."

                echo "Portfolios:"
                for portfolioId in $portfolios; do
                  echo "\t${portfolioId}"
                done

                echo "Budgets:"
```

```
                if [[ -n "$budgetName" ]]; then
                  echo "\t${budgetName}"
                fi

                echo "Tag Options:"
                for tagOptionId in $tagOptions; do
                  echo "\t${tagOptionId}"
                done

                echo "Service Actions on Provisioning Artifact:"
                for association in
 "${provisioningArtifactServiceActionAssociations[@]}"; do
                  echo "\t${association}"
                done

                read -p "Are you sure you want to delete ${productId}? y,n "
                if [[ ! $REPLY =~ ^[Yy]$ ]]; then
                   exit
                fi

                for portfolioId in $portfolios; do
                  echo "Disassociating ${portfolioId}"
                  aws servicecatalog disassociate-product-from-portfolio --product-
id $productId --portfolio-id $portfolioId
                done

                if [[ -n "$budgetName" ]]; then
                  echo "Disassociating ${budgetName}"
                  aws servicecatalog disassociate-budget-from-resource --budget-
name "$budgetName" --resource-id $productId
                fi

                for tagOptionId in $tagOptions; do
                  echo "Disassociating ${tagOptionId}"
                  aws servicecatalog disassociate-tag-option-from-resource --tag-
option-id $tagOptionId --resource-id $productId
                done

                for association in
 "${provisioningArtifactServiceActionAssociations[@]}"; do
                  associationPair=(${association//:/ })
                  provisioningArtifactId=${associationPair[0]}
                  serviceActionsList=${associationPair[1]}
                  serviceActionIds=${serviceActionsList//,/ }
```

```
                  for serviceActionId in $serviceActionIds; do
                    echo "Disassociating ${serviceActionId} from
 ${provisioningArtifactId}"
                    aws servicecatalog disassociate-service-action-from-
provisioning-artifact --product-id $productId --provisioning-artifact-id
 $provisioningArtifactId --service-action-id $serviceActionId
                  done
                done

                echo "Deleting product ${productId}"
                aws servicecatalog delete-product --id $productId

              }; f
```

3. 保存该文件。

使用终端窗口将 force-delete-product 别名添加到**alias**文件中

1. 打开终端窗口，并运行以下命令

   ```
   $ cat >> ~/.aws/cli/alias
   ```

2. 将别名脚本粘贴到终端窗口，然后按 CTRL+D 退出 `cat` 命令。

给 force-delete-product 别名打电话

1. 在设备上的终端窗口中运行以下命令以调用删除产品别名

   ```
   $ aws servicecatalog force-delete-product {product-id}
   ```

   下面的示例展示了 `force-delete-product` 别名命令及其生成的响应

   ```
   $ aws servicecatalog force-delete-product prod-123
   ```

   ```
   Before deleting a product, the following associated resources must
 be disassociated. These resources will not be deleted. This action may take some
 time, depending on the number of resources being disassociated.
   Portfolios:
   ```

```
                  port-123
                Budgets:
                    budgetName
                Tag Options:
                    tag-123
                Service Actions on Provisioning Artifact:
                    pa-123:act-123
                Are you sure you want to delete prod-123? y,n
```

2. 输入 y 以确认您要删除产品。

成功删除产品后，终端窗口将显示如下结果

```
Disassociating port-123
Disassociating budgetName
Disassociating tag-123
Disassociating act-123 from pa-123
Deleting product prod-123
```

其他资源

有关 AWS CLI 和使用别名和删除 AWS Service Catalog 产品的更多信息，请查看以下资源：

- AWS Command Line Interface (CLI) 用户指南中的创建和使用 AWS CLI 别名。
- AWS CLI别名存储库 git 存储库。
- 删除 AWS Service Catalog 产品。
- AWSre: Invent 2016：《有效用户》开启AWS CLI。YouTube

解决删除产品时资源取消关联失败的问题

如果由于资源解除关联异常而导致您之前尝试删除产品失败，请查看下面的异常列表及其解决方案。

> **Note**
>
> 如果您在收到资源解除关联失败消息之前关闭了删除产品窗口，则可以按照后续删除产品部分中的步骤一至三再次打开该窗口。

要解决资源解除关联失败的问题

在删除产品窗口中，查看关联表的状态列。确定失败的资源解除关联异常和建议解决方案：

状态异常类型	原因	解决方案
Product prod-****	AWS Service Catalog 无法删除产品，因为产品仍有关联的预算 TagOptions，至少有一个`ProvisioningArtifact` 与之关联的操作，产品仍分配给产品组合，产品有用户，或者产品有约束条件。	再次尝试删除产品。
用户：`username` 未获得授权执行：	尝试删除产品的用户没有取消关联产品资源的必要权限。	AWS Service Catalog 建议联系您的账户管理员，了解有关取消关联您当前无权取消关联的产品资源的更多信息。

管理版本

您可以在创建产品时分配产品版本，并且您可以随时更新产品版本。

版本具有 AWS CloudFormation 模板、标题、描述、状态和指南。

版本状态

一个版本可以有三种状态之一：

- 活动 - 活动版本出现在版本列表中，并允许用户启动该版本。
- 非活动 - 非活动版本从版本列表中隐藏。从此版本启动的现有预配置产品不会受到影响。
- Deleted (已删除) - 被删除的某个版本将从版本列表中删除。删除版本的操作无法撤消。

版本指南

您可以设置版本指南，以便向最终用户提供有关产品版本的信息。版本指南仅影响活动的产品版本。

版本指南有两个选项：

- 无 - 默认情况下，产品版本没有任何指导。最终用户可以使用该版本更新和启动预配置产品。
- 已淘汰-用户无法使用已淘汰的产品版本启动新的预配置产品。如果之前启动的预配置产品使用的是现已淘汰的版本，则用户只能使用现有版本或新版本更新该预配置产品。

更新版本

您可以在创建产品时分配产品版本，也可以随时更新版本。有关创建产品的更多信息，请参阅创建产品。

更新产品版本

1. 在 AWS Service Catalog 控制台中，选择产品。
2. 从产品列表中，选择要更新其版本的产品。
3. 在产品详细信息页面上，选择版本选项卡，然后选择要更新的版本。
4. 在版本详细信息页面上，编辑产品版本，然后选择保存更改。

使用 AWS Service Catalog 约束

您可以应用约束，以控制当最终用户启动特定组合中的产品时所应用的规则。当最终用户启动产品时，他们将看到您已使用约束应用的规则。您可以在产品放入产品组合后将约束应用于产品。约束一经创建便立即生效，并且它们已应用于尚未启动的产品的所有当前版本。

约束

- AWS Service Catalog 启动约束
- AWS Service Catalog 通知约束
- AWS Service Catalog 标签更新约束
- AWS Service Catalog 堆栈集约束
- AWS Service Catalog 模板约束

AWS Service Catalog 启动约束

启动约束指定 AWS Identity and Access Management（IAM）角色，用户启动、更新或终止产品时 AWS Service Catalog 将担任此角色。IAM 角色是一个权限集合，用户或 AWS 服务可临时利用这些权限来使用 AWS 服务。有关介绍性示例，请参阅：

- AWS CloudFormation 产品类型：步骤 6：添加启动约束以分配 IAM 角色
- Terraform 开源或 Terraform 云产品类型：步骤 5：创建启动角色

启动约束适用于产品组合中的产品（产品-产品组合关联）。启动约束不适用于产品组合层面，也不适用于跨所有产品组合的某个产品。要将启动约束与产品组合中的所有产品相关联，您必须将启动约束分别应用于每个产品。

如果没有启动约束，最终用户必须使用自己的 IAM 凭证启动和管理产品。为此，他们必须具有针对 AWS CloudFormation、 产品所使用的 AWS 服务和 AWS Service Catalog 的权限。通过使用启动角色，您可改为将最终用户的权限限定为他们对该产品所需的最小权限。有关最终用户权限的更多信息，请参阅Identity and Access Management AWS Service Catalog。

要创建和分配 IAM 角色，您必须拥有以下 IAM 管理权限：

- `iam:CreateRole`
- `iam:PutRolePolicy`
- `iam:PassRole`
- `iam:Get*`
- `iam:List*`

配置启动角色

您向产品分配的作为启动约束的 IAM 角色必须拥有使用以下项的权限：

对于 Cloudformation 产品

- `arn:aws:iam::aws:policy/AWSCloudFormationFullAccess` AWS CloudFormation 托管策略
- 产品 AWS CloudFormation 模板中的服务
- 自服务拥有的 Amazon S3 存储桶中读取 AWS CloudFormation 模板的访问权限。

对于 Terraform 产品

- 产品的 Amazon S3 模板中使用的服务
- 自服务拥有的 Amazon S3 存储桶中读取 Amazon S3 模板的访问权限。
- `resource-groups:Tag` 用于在 Amazon EC2 实例中进行标记（Terraform 预置引擎在执行配置操作时担任）
- `resource-groups:CreateGroup` 用于资源组标记（AWS Service Catalog 用于创建资源组和分配标签时担任）

IAM 角色的信任策略必须允许 AWS Service Catalog 担任该角色。在以下过程中，您选择 AWS Service Catalog 作为角色类型时，将自动设置信任策略。如果您不使用控制台，请参阅如何将信任策略与 IAM 角色一起使用中的为担任角色的 AWS 服务创建信任策略部分。

> **Note**
>
> 无法在启动角色中分配 `servicecatalog:ProvisionProduct`、`servicecatalog:TerminateProvisionedProduct` 和 `servicecatalog:UpdateProvisionedProduct` 权限。您必须使用 IAM 角色，如授予 AWS Service Catalog 最终用户权限部分中的内联策略步骤所示。

> **Note**
>
> 要在 AWS Service Catalog 控制台中查看预配置的 Cloudformation 产品和资源，最终用户需要 AWS CloudFormation 读取权限。在控制台中查看预配置产品和资源不会使用启动角色。

创建启动角色

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。

   Terraform 产品需要额外的启动角色配置。有关更多信息，请查看 Terraform 开源产品入门中的步骤 5：创建启动角色。
2. 选择 角色。
3. 选择创建新角色。
4. 输入角色名称并选择 Next Step。

5. 在 AWS Service Catalog 旁边的AWS服务角色下，选择选择。
6. 在 Attach Policy 页面上，选择 Next Step。
7. 要创建角色，请选择 Create Role。

将策略附加到新角色

1. 选择您创建的角色以查看角色详细信息页面。
2. 选择 Permissions 选项卡，展开 Inline Policies 部分。然后选择 click here。
3. 选择 Custom Policy，然后选择 Select。
4. 输入策略的名称，然后将以下内容粘贴到 Policy Document 编辑器中：

```
          "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "s3:ExistingObjectTag/servicecatalog:provisioning":"true"
            }
         }
   ]
}
```

> **Note**
>
> 为启动约束配置启动角色时，您必须使用以下字符串：`"s3:ExistingObjectTag/servicecatalog:provisioning":"true"`。

5. 为产品使用的每个额外服务的策略添加一行。例如，要为 Amazon Relational Database Service (Amazon RDS) 添加权限，请在 `Action` 列表中的最后一行的末尾键入逗号，然后添加以下行：

```
"rds:*"
```

6. 选择应用策略。

应用启动约束

您配置启动角色后，将该角色作为启动约束分配给产品。此操作将通知 AWS Service Catalog 在最终用户启动产品时担任角色。

将角色分配给产品

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择包含产品的产品组合。
3. 选择约束选项卡并选择创建约束。
4. 自产品中选择产品，然后在约束类型下选择启动。选择继续。
5. 在启动约束 部分，您可以从自己的账户中选择 IAM 角色、输入 IAM 角色 ARN 或输入角色名称。

   如果您指定角色名称，当账户使用启动约束时，将使用账户中具有该名称的 IAM 角色。此方法允许使用与账户无关的启动角色约束，因此您可以为每个共享账户创建更少的资源。

   > **Note**
   > 在创建启动约束的账户中，以及使用此启动约束启动产品的用户账户中，都必须存在给定的角色名称。

6. 指定 IAM 角色后，选择 创建。

在启动限制中添加混淆代理

AWS Service Catalog 支持为使用担任角色请求运行的 API 提供混淆代理保护。添加启动约束时，您可以使用启动角色信任策略中的 `sourceAccount` 和 `sourceArn` 条件来限制启动角色访问权限。它可确保启动角色由可信来源调用。

在以下示例中，AWS Service Catalog 最终用户属于账户 111111111111。AWS Service Catalog 管理员为产品创建 `LaunchConstraint` 时，最终用户可以在启动角色信任策略中指定以下条件，将担任角色限制为账户 111111111111。

```
"Condition":{
   "ArnLike":{
      "aws:SourceArn":"arn:aws:servicecatalog:us-east-1:111111111111:*"
   },
   "StringEquals":{
```

```
        "aws:SourceAccount":"111111111111"
    }

}
```

使用 `LaunchConstraint` 预配置产品的用户必须具有相同的 `AccountId` (111111111111)。否则，为防止滥用启动角色，操作将出现 `AccessDenied` 错误并失败。

以下 AWS Service Catalog API 已通过混淆代理保护得到保护：

- `LaunchConstraint`
- `ProvisionProduct`
- `UpdateProvisionedProduct`
- `TerminateProvisionedProduct`
- `ExecuteProvisionedProductServiceAction`
- `CreateProvisionedProductPlan`
- `ExecuteProvisionedProductPlan`

对 AWS Service Catalog 的 `sourceArn` 保护仅支持模板化 ARN，例如“`arn:<aws-partition>:servicecatalog:<region>:<accountId>:`”。它不支持特定的资源 ARN。

验证启动约束

要确认 AWS Service Catalog 使用角色启动产品，从 AWS Service Catalog 控制台启动产品并成功预配置了产品。要在将约束发布给用户前对其进行测试，请创建包含相同产品的测试产品组合，并使用该产品组合测试约束。

启动产品

1. 在 AWS Service Catalog 控制台的菜单中，选择 Service Catalog、最终用户。
2. 选择产品以打开产品详细信息页面。在启动选项表中，确认已显示角色的 Amazon 资源名称 (ARN)。
3. 选择启动产品。
4. 继续执行启动步骤，填入任何所需信息。
5. 确认产品已成功启动。

AWS Service Catalog 通知约束

> **Note**
>
> AWS Service Catalog 不支持 Terraform 开源或 Terraform 云产品的通知约束。

通知约束指定 Amazon SNS 主题以接收有关堆栈事件的通知。

使用以下过程创建 SNS 主题并订阅它。

创建 SNS 主题和订阅

1. 通过 https://console.aws.amazon.com/sns/v3/home 打开 Amazon SNS 控制台。
2. 选择创建主题。
3. 键入主题名称，然后选择 创建主题。
4. 选择创建订阅。
5. 对于协议，请选择电子邮件。对于 Endpoint，请键入您用于接收通知的电子邮件地址。选择创建订阅。
6. 您将收到一封包含主题行 `AWS Notification - Subscription Confirmation` 的确认电子邮件。打开电子邮件，然后按照说明操作以完成订阅。

通过以下过程使用您在上一过程中创建的 SNS 主题应用通知约束。

向产品应用通知约束

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择包含产品的产品组合。
3. 展开约束并选择添加约束。
4. 从产品中选择产品并将约束类型设置为通知。选择继续。
5. 选择 Choose a topic from your account，然后选择您根据 Topic Name 创建的 SNS 主题。
6. 选择提交。

AWS Service Catalog 标签更新约束

> **Note**
>
> AWS Service Catalog 不支持对 Terraform 开源产品使用标签更新约束。

使用标签更新约束，AWS Service Catalog 管理员可以允许或禁止最终用户更新与预配置产品关联的资源上的标签。如果允许更新标签，则与产品或产品组合关联的新标签将在预配置产品更新期间应用于预配置的资源。

启用产品的标签更新

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择包含要更新的产品的产品组合。
3. 选择约束 选项卡并选择添加约束。
4. 在约束类型下，选择标签更新。
5. 从产品中选择产品，然后选择继续。
6. 在标签更新页面上，选择启用标签更新。
7. 选择提交。

AWS Service Catalog 堆栈集约束

> **Note**
>
> - AWS Service Catalog 不支持对 Terraform 开源产品的堆栈集约束。
> - AutoTags 目前不支持AWS CloudFormation StackSets。

堆栈集约束允许您使用配置产品部署选项AWS CloudFormation StackSets。您可以为产品启动指定多个账户和区域。最终用户可以管理这些账户，并确定产品的部署位置和部署顺序。

向产品应用堆栈集约束

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。

2. 选择包含所需产品的产品组合。
3. 选择约束选项卡并选择创建约束。
4. 在产品中，选择产品。在约束类型中，选择堆栈集。
5. 为您的堆栈集约束配置账户、区域和权限。
   - 在账户设置中，确定要在其中创建产品的账户。
   - 在区域设置中，选择要部署产品的地理区域以及在这些区域中您希望的产品部署顺序。
   - 在权限中，选择一个 IAM StackSet 管理员角色来管理您的目标账户。如果您不选择角色，则 StackSets使用默认 ARN。了解有关设置堆栈集权限的更多信息。
6. 选择创建。

AWS Service Catalog 模板约束

> **Note**
>
> AWS Service Catalog 不支持 Terraform 开源或 Terraform 云产品的模板约束。

要限制最终用户在启动产品时可用的选项，请应用模板约束。应用模板约束可确保最终用户可在不违反组织的合规性要求的情况下使用产品。您可将模板约束应用于 AWS Service Catalog 产品组合中的产品。产品组合必须先包含一个或多个产品，然后才能定义模板约束。

模板约束包含一个或多个规则，这些规则可缩小在产品的基础 AWS CloudFormation 模板中定义的参数的允许值。AWS CloudFormation 模板中的参数定义用户在创建堆栈时指定的值的集合。例如，参数可定义用户在启动包含 EC2 实例的堆栈时可从中选择的多种实例类型。

如果模板中的参数值集对于产品组合的目标受众来说太广泛，则可定义模板约束来限制用户在启动产品时可选择的值。例如，如果模板参数包含的 EC2 实例类型对于只应使用小型实例类型 (例如，`t2.micro` 或 `t2.small`) 的用户来说过大，则可添加模板约束以限制最终用户可选择的实例类型。有关 AWS CloudFormation 参数的更多信息，请参阅《AWS CloudFormation用户指南》中的参数。

模板约束绑定于产品组合内。如果将模板约束应用于一个产品组合的某个产品，然后您将此产品包含在另一个产品组合中，则这些约束将不会应用于第二个产品组合的此产品。

如果将模板约束应用于已与用户共享的产品，则约束会立即应用到随后启动的所有产品和产品组合中所有版本的产品。

通过使用规则编辑器或通过在 AWS Service Catalog 管理员控制台中将规则编写为 JSON 文本来定义模板约束规则。有关规则的更多信息（包括语法和示例），请参阅模板约束规则。

要在将约束发布给用户前对其进行测试，请创建包含相同产品的测试产品组合，并使用该产品组合测试约束。

将模板约束应用于产品

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在产品组合页面，选择包含要将模板约束应用于其上的产品的产品组合。
3. 展开约束部分并选择添加约束。
4. 在选择产品和类型窗口中，对于产品，选择您要定义模板约束的产品。然后，对于约束类型项，选择模板。选择继续。
5. 在模板约束生成器页面，使用 JSON 编辑器或规则生成器界面来编辑约束规则。

   - 要编辑规则的 JSON 代码，请选择约束文本编辑器选项卡。此选项卡上提供了多个示例来帮助您开始操作。

     要使用规则生成器界面构建规则，请选择规则生成器选项卡。在此选项卡上，您可以选择产品的模板中指定的任何参数，并可以为该参数指定允许值。根据参数的类型，可通过选择清单中的项目、指定数量或指定逗号分隔列表中的一组值来指定允许值。

     在构建完规则后，选择添加规则。规则会显示在规则生成器选项卡上的表中。要查看和编辑 JSON 输出，请选择约束文本编辑器选项卡。
6. 在编辑完约束的规则后，选择提交。要查看约束，请转到产品组合详细信息页，然后展开约束。

模板约束规则

在 AWS Service Catalog 产品组合中定义模板约束的规则描述了最终用户可以使用模板的时间以及可以为用于创建其尝试使用的产品的 AWS CloudFormation 模板中声明的参数指定的值。规则可用于防止最终用户无意中指定错误的值。例如，您可以添加规则以验证最终用户是否在给定 VPC 中指定了有效子网或是否将 `m1.small` 实例类型用于测试环境。AWS CloudFormation 先使用规则验证参数值，然后再为产品创建资源。

每个规则包含两个属性：规则条件（可选）和断言（必需）。规则条件确定规则的生效时间。断言描述用户可为特定参数指定的值。如果您未定义规则条件，则规则的断言始终生效。要定义规则条件和断言，可使用特定于规则的内部函数，只能在模板的 `Rules` 部分中使用这些函数。您可以嵌套函数，但规则条件或断言的最终结果必须为 true 或 false。

例如，假设您在 Parameters 部分中声明了 VPC 和子网参数。您可以创建一个规则来验证给定子网是否位于特定的 VPC 中。因此，当用户指定 VPC 时，AWS CloudFormation 将评估断言以检查在创建或更新堆栈之前，子网参数值是否在 VPC 中。如果参数值无效，则 AWS CloudFormation 将无法创建或更新堆栈。如果用户未指定 VPC，则 AWS CloudFormation 不会检查子网参数值。

语法

模板的Rules 部分由后跟冒号的密钥名称 Rules 组成。所有规则声明都被括在括号里。如果您声明多个规则，则可用逗号将它们分隔开。对于每个规则，您必须声明一个用引号引起来的逻辑名称，后跟冒号以及将规则条件和断言括起来的括号。

规则可以包含 RuleCondition 属性，且必须包含 Assertions 属性。对于每个规则，您可以仅定义一个规则条件；您可以在 Assertions 属性内定义一个或多个断言。可以通过使用特定于规则的内部函数定义规则条件和断言，如以下伪模板所示：

```
"Rules":{
   "Rule01":{
      "RuleCondition":{
         "Rule-specific intrinsic function"
      },
      "Assertions":[
         {
            "Assert":{
               "Rule-specific intrinsic function"
            },
            "AssertDescription":"Information about this assert"
         },
         {
            "Assert":{
               "Rule-specific intrinsic function"
            },
            "AssertDescription":"Information about this assert"
         }
      ]
   },
   "Rule02":{
      "Assertions":[
         {
            "Assert":{
               "Rule-specific intrinsic function"
            },
            "AssertDescription":"Information about this assert"
```

```
        }
      ]
    }
}
```

此伪模板显示包含两个分别名为 Rules 和 Rule01 的规则的 Rule02 部分。Rule01 包含一个规则条件和两个断言。如果规则条件中的函数的计算结果为 true，则将计算和应用每个断言中的函数。如果规则条件为 false，则此规则不会生效。由于 Rule02 没有规则条件（这意味着始终计算和应用断言），因此它始终生效。

有关用于定义规则条件和断言的特定于规则的内置函数信息，请参阅AWS CloudFormation《用户指南》中的AWS规则函数。

示例：按条件验证参数值

以下两个规则检查 InstanceType 参数的值。根据环境参数（test 或 prod）的值，用户必须为 m1.small 参数指定 m1.large 或 InstanceType。必须在同一模板的 InstanceType 部分中声明 Environment 和 Parameters 参数。

```
"Rules" : {
  "testInstanceType" : {
    "RuleCondition" : {"Fn::Equals":[{"Ref":"Environment"}, "test"]},
    "Assertions" : [
      {
        "Assert" :  { "Fn::Contains" : [ ["m1.small"], {"Ref" : "InstanceType"} ] },
        "AssertDescription" : "For the test environment, the instance type must be
 m1.small"
      }
    ]
  },
  "prodInstanceType" : {
    "RuleCondition" : {"Fn::Equals":[{"Ref":"Environment"}, "prod"]},
    "Assertions" : [
      {
        "Assert" :  { "Fn::Contains" : [ ["m1.large"], {"Ref" : "InstanceType"} ] },
        "AssertDescription" : "For the prod environment, the instance type must be
 m1.large"
      }
    ]
  }
}
```

AWS Service Catalog 服务操作

> **Note**
>
> AWS Service Catalog 不支持 Terraform 开源或 Terraform 云产品的服务操作。

AWS Service Catalog 使您能够减少管理维护和最终用户培训，同时合规且遵守安全措施。利用服务操作，作为管理员，您可以允许最终用户在 AWS Service Catalog 中执行操作任务、排查问题、运行批准的命令或请求权限。使用 AWS Systems Manager 文档定义服务操作。AWS Systems Manager 文档提供对实施 AWS 最佳实践的预定义操作（如 Amazon EC2 停止和重启）的访问权限，您也可以定义自定义操作。

在本教程中，您为最终用户提供重启 Amazon EC2 实例的功能。添加必要的权限，定义服务操作，将服务操作与产品关联，将操作用于预配置产品来测试最终用户体验。

先决条件

本教程假定您具有完整的 AWS 管理员权限，已熟悉 AWS Service Catalog 并且您已具有一组基础的产品、产品组合和用户。如果您不熟悉 AWS Service Catalog，请在使用本教程之前完成设置和 入门 任务。

主题

- 步骤 1：配置最终用户权限
- 步骤 2：创建服务操作
- 步骤 3：将服务操作与产品版本关联
- 步骤 4：测试最终用户体验
- 步骤 5：使用 AWS CloudFormation 管理服务操作
- 步骤 6：问题排查

步骤 1：配置最终用户权限

最终用户必须拥有必需的权限才能查看和执行特定服务操作。在本示例中，最终用户需要权限才能访问 AWS Service Catalog 服务操作功能和执行 Amazon EC2 重启。

更新权限

1. 通过以下网址打开 AWS Identity and Access Management (IAM) 控制台：https://console.aws.amazon.com/iam/。
2. 从菜单中找到用户群组。
3. 选择最终用户将用于访问 AWS Service Catalog 资源的群组。在本示例中，我们选择最终用户组。在您自己的实现中，选择相关最终用户使用的组。
4. 在组的详细信息页面的权限选项卡上，创建新策略或编辑现有策略。在本示例中，我们通过选择为组的 AWS Service Catalog 预置权限和终止权限创建的自定义策略来向现有策略添加权限。
5. 在策略页面上，选择编辑策略以添加必要的权限。您可以使用可视化编辑器或 JSON 编辑器来编辑策略。在本示例中，我们使用 JSON 编辑器添加权限。在本教程中，向策略添加以下权限：

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "Stmt1536341175150",
   "Action": [
    "servicecatalog:ListServiceActionsForProvisioningArtifact",
    "servicecatalog:ExecuteprovisionedProductServiceAction",
    "ssm:DescribeDocument",
    "ssm:GetAutomationExecution",
    "ssm:StartAutomationExecution",
    "ssm:StopAutomationExecution",
    "cloudformation:ListStackResources",
    "ec2:DescribeInstanceStatus",
    "ec2:StartInstances",
    "ec2:StopInstances"
   ],
   "Effect": "Allow",
   "Resource": "*"
  }
 ]
}
```

6. 编辑策略之后，审查并批准对策略的更改。最终用户组中的用户现在拥有在 AWS Service Catalog 中执行 Amazon EC2 重启操作所需的权限。

步骤 2：创建服务操作

接下来，您要创建一个服务操作来重启 Amazon EC2 实例。

1. 打开AWS Service Catalog控制台，网址为 https://console.aws.amazon.com/sc/。
2. 从菜单中，选择服务操作。
3. 在服务操作页面上，选择创建新操作。
4. 在创建操作页面上，选择 AWS Systems Manager 文档以定义服务操作。Amazon EC2 实例重启操作由 AWS Systems Manager 文档定义，因此我们保留下拉菜单上的默认选项 Amazon 文档。
5. 搜索并选择 AWS-RestartEC2Instance 操作。
6. 请为操作提供一个对您的环境和团队有意义的名称和描述。最终用户将看到此描述，从而选择帮助他们了解操作作用的内容。
7. 在参数和目标配置下，选择将作为操作目标的 SSM 文档参数（例如，实例 ID），然后选择参数的目标。选择添加参数以添加其他参数。
8. 在权限下，选择一个角色。我们在此示例中使用默认权限。还可以在此页面上执行和定义其他权限配置。
9. 审查配置之后，选择创建操作。
10. 在下一页上，操作创建完成后且可用时，将出现确认。

步骤 3：将服务操作与产品版本关联

定义操作后，您必须将产品与定义的操作关联。

1. 在服务操作 页面上，选择 AWS-RestartEC2instance，然后选择关联操作。
2. 在关联操作页面上，选择您希望您的最终用户在其上执行服务操作的产品。在本示例中，我们选择Linux 桌面。
3. 选择产品版本。请注意，您可以使用顶部的复选框选择所有版本。
4. 选择关联操作。
5. 在下一页上，将显示一条确认消息。

您现在已经在 AWS Service Catalog 中创建了服务操作。本教程的下一步是，以最终用户身份使用服务操作。

步骤 4：测试最终用户体验

最终用户可以在预配置产品上执行服务操作。在本教程中，最终用户至少必须具有一个预配置产品。预配置产品应从您在上一步中与服务操作关联的产品版本启动。

以最终用户身份访问服务操作

1. 以最终用户身份登录 AWS Service Catalog 控制台。
2. 在 AWS Service Catalog 控制面板上的导航窗格中，选择预配置产品列表。此列表将显示为最终用户账户预配置的产品。
3. 在预配置产品列表 页面上，选择已预配置的实例。
4. 在预配置产品详细信息页面上，选择右上角的操作，然后选择 AWS-RestartEC2instance 操作。
5. 确认您要执行自定义操作。您收到操作已发送的确认。

步骤 5：使用 AWS CloudFormation 管理服务操作

您可以使用 AWS CloudFormation 资源创建服务操作及其关联。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的以下内容：

- AWS::ServiceCatalog::CloudFormation产品 ProvisioningArtifactProperties
- AWS::ServiceCatalog::ServiceAction协会

> **Note**
>
> 如果您要管理 AWS CloudFormation 资源的服务操作关联，请不要通过 AWS Command Line Interface 或 AWS Management Console 添加或删除服务操作。当您执行堆栈更新时，在 AWS CloudFormation 之外对服务操作所做的任何更改都将被替换。

步骤 6：问题排查

如果您的服务操作执行失败，则可以在预配置产品页面上服务操作执行事件的输出部分中找到错误消息。您可以在下面看到常见错误消息的说明。

> **Note**
> 错误消息的确切文本可能会发生更改，因此您应避免在任何类型的自动执行的过程中使用这些文本。

内部故障

AWS Service Catalog 遇到了内部错误。请稍后重试。如果错误仍存在，请与客户支持联系。

调用 StartAutomationExecution 操作时出错 (ThrottlingException)

服务操作执行受后端服务（例如 SSM）的限制。

代入角色时拒绝访问

AWS Service Catalog 无法代入在服务操作定义中指定的角色。请确保在角色的信任策略中将 servicecatalog.amazonaws.com 主体或区域主体（如 servicecatalog.us-east-1.amazonaws.com）列入允许列表。

调用 StartAutomationExecution 操作时出错 (AccessDeniedException)：用户无权在资源 StartAutomationExecution 上执行:ssm:。

服务操作定义中指定的角色无权调用 ssm: StartAutomationExecution。确保角色具有适当的 SSM 权限。

*TargetType*在预配置产品中找不到任何类型为的资源

预配置产品不包含任何与在 SSM 文档中指定的目标类型匹配的资源，例如 AWS::EC2::实例。检查您的预配置产品是否有这些资源，或确认文档是否正确。

Document with that name does not exist (具有该名称的文档不存在)

在服务操作定义中指定的文档不存在。

Failed to describe SSM Automation document (无法描述 SSM Automation 文档)

AWS Service Catalog 在尝试描述指定文档时遇到来自 SSM 的未知异常。

Failed to retrieve credentials for role (无法检索角色的凭证)

AWS Service Catalog 在代入指定的角色时遇到未知错误。

在 ***{ValidValue1}#{ValidValue2}*** 中找不到参数的值 ***InvalidValue***“”

传递给 SSM 的参数值不在文档的允许值列表中。确认提供的参数有效，然后重试。

参数类型错误。为提供的值***ParameterName***不是有效的字符串。

传递给 SSM 的参数值对文档上的类型无效。

Parameter is not defined in service action definition (未在服务操作定义中定义参数)

已将一个未在服务操作定义中定义的参数传递给 AWS Service Catalog。您只能使用在服务操作定义中定义的参数。

执行/取消操作时，步骤失败。*#####*有关更多诊断详细信息，请参阅 Automation 服务问题排查指南。

SSM Automation 文档中的步骤失败。请参阅消息中的错误以进一步排除问题。

不允许使用以下参数值，因为它们不在预配置产品中：***InvalidResourceId***

用户已请求对不在预配置产品中的资源执行操作。

TargetType 未为 SSM 自动化文档定义

服务操作需要 SSM 自动化文档进行 TargetType 定义。检查您的 SSM Automation 文档。

将 AWS Marketplace 产品添加到您的产品组合

您可以将 AWS Marketplace 产品添加到您的产品组合来使这些产品对 AWS Service Catalog 最终用户可用。

AWS Marketplace 是一个在线商店，您可以从其中查找、订阅和快速开始使用大量可选软件和服务。AWS Marketplace 中的产品类型包括数据库、应用程序服务器、测试工具、监控工具、内容管理工具和商业智能软件。AWS Marketplace 可在 https://aws.amazon.com/marketplace 处获取。请注意，您不能将软件即服务（SaaS）产品从添加 AWS Marketplace 到 AWS Service Catalog。

您可以复制带有 AWS CloudFormation 模板的产品到 AWS Service Catalog，从而将 AWS Marketplace 产品分发给 AWS Service Catalog 最终用户，然后将该产品添加到产品组合中。

> **Note**
>
> AWS Service Catalog 不支持使用 Terraform 开源或 Terraform 云产品模板向 AWS Service Catalog 最终用户分发 AWS Marketplace 产品。

AWS Marketplace 支持直接使用 AWS Service Catalog 或通过手动选项订阅和添加产品。我们建议您使用专为 AWS Service Catalog 设计的功能添加产品。

使用 AWS Marketplace 管理 AWS Service Catalog 产品

您可以使用自定义界面直接向 AWS Marketplace 添加订阅的 AWS Service Catalog 产品。在 AWS Marketplace 中，选择 Service Catalog。有关更多信息，请参阅 AWS Marketplace 帮助和常见问题解答中的将产品复制到 AWS Service Catalog。

手动管理和添加 AWS Marketplace 产品

完成以下步骤可订阅 AWS Marketplace 产品、在 AWS CloudFormation 模板中定义该产品并将模板添加到 AWS Service Catalog 产品组合。

订阅 AWS Marketplace 产品

1. 前往位于 https://aws.amazon.com/marketplace 的 AWS Marketplace。
2. 浏览产品或执行搜索以查找要添加到 AWS Service Catalog 产品组合的产品。选择产品以查看产品详细信息页面。
3. 选择继续以查看完成页面，然后选择手动启动选项卡。

   完成页面上的信息包括支持的 Amazon Elastic Compute Cloud (Amazon EC2) 实例类型、支持的 AWS 区域 以及产品对每个 AWS 区域使用的亚马逊机器映像（AMI）ID。注意，有些选择可能会影响到成本。您将在后续步骤中使用此信息来自定义 AWS CloudFormation 模板。
4. 选择 Accept Terms 订阅产品。

   订阅产品后，您可以随时访问 AWS Marketplace 中产品完成页面上的信息，选择您的软件，然后选择产品即可。

在 AWS Marketplace 模板中定义 AWS CloudFormation 产品

要完成以下步骤，您将使用一个 AWS CloudFormation 示例模板作为起始点，并将自定义此模板以使其代表您的 AWS Marketplace 产品。要访问示例模板，请参阅《AWS CloudFormation 用户指南》中的示例模板。

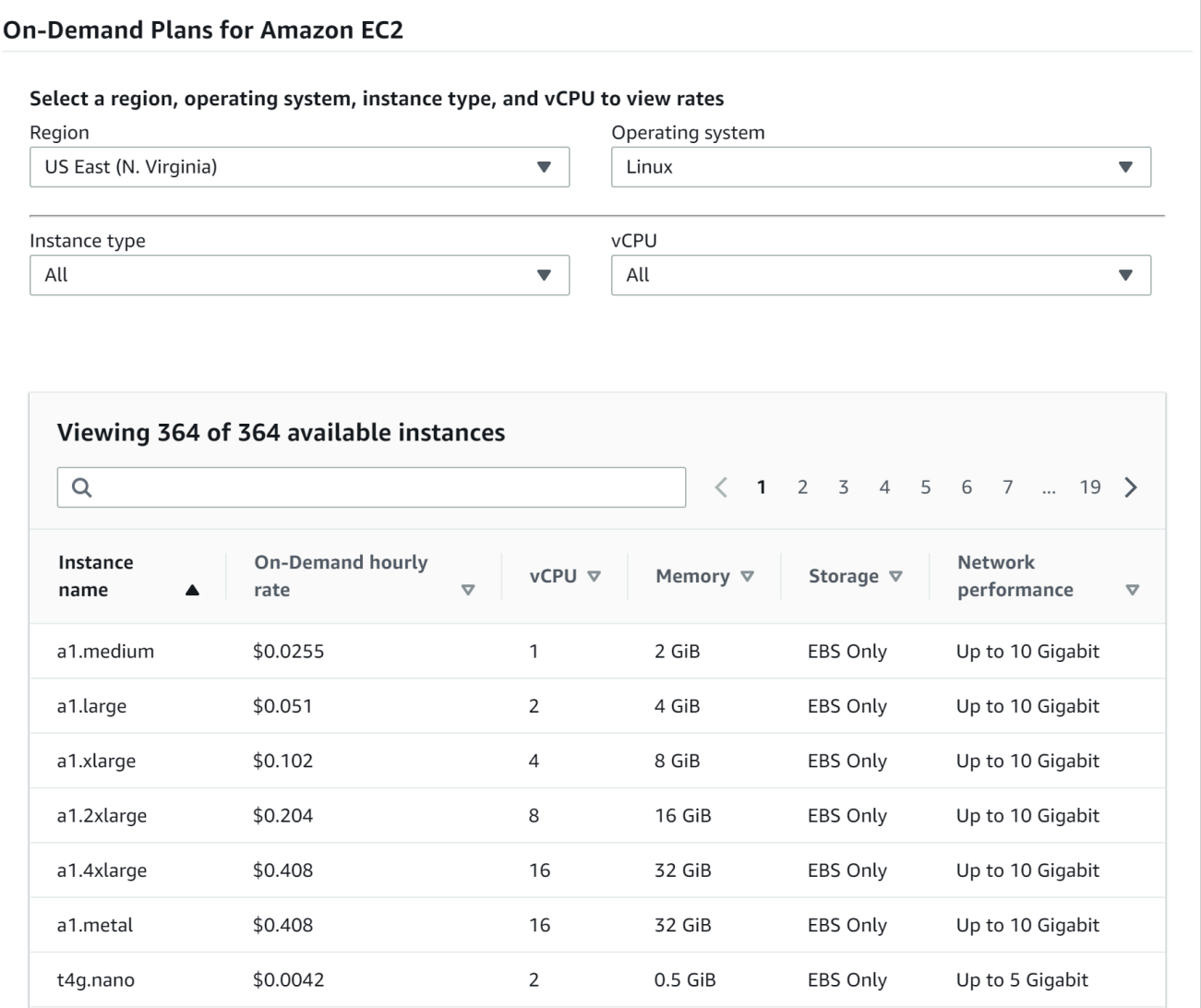
1. 在《AWS CloudFormation 用户指南》中的示例模板页面上，选择产品将用于的 AWS 区域。AWS 区域必须受 AWS Marketplace 产品的支持。您可以在 AWS Marketplace 中的产品完成页面上查看受支持的区域。

2. 要查看适合该区域的服务示例模板的列表，请选择服务链接。
3. 您可以使用满足您需求的任一示例作为起始点。此过程中的步骤将使用 Amazon EC2 instance in a security group 模板。要查看示例模板，请选择 View，然后在本地保存模板的副本，以便您能对其进行编辑。本地文件的扩展名必须为 `.template`。
4. 在文本编辑器中打开模板文件。
5. 自定义模板顶部的描述。您的描述可能与以下示例类似：

   ```
   "Description": "Launches a LAMP stack from AWS Marketplace",
   ```

6. 自定义 `InstanceType` 参数，使其仅包括产品支持的 EC2 实例类型。如果模板包含不受支持的 EC2 实例类型，则最终用户无法启动产品。

   a. 在 AWS Marketplace 中的产品完成页面上，查看定价详细信息部分中受支持的 EC2 实例类型。



Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
a1.medium	$0.0255	1	2 GiB	EBS Only	Up to 10 Gigabit
a1.large	$0.051	2	4 GiB	EBS Only	Up to 10 Gigabit
a1.xlarge	$0.102	4	8 GiB	EBS Only	Up to 10 Gigabit
a1.2xlarge	$0.204	8	16 GiB	EBS Only	Up to 10 Gigabit
a1.4xlarge	$0.408	16	32 GiB	EBS Only	Up to 10 Gigabit
a1.metal	$0.408	16	32 GiB	EBS Only	Up to 10 Gigabit
t4g.nano	$0.0042	2	0.5 GiB	EBS Only	Up to 5 Gigabit

b. 在您的模板中，将默认实例类型更改为所选的受支持的 EC2 实例类型。

c. 编辑 AllowedValues 列表，使其仅包含产品所支持的 EC2 实例类型。

d. 删除您不希望最终用户在从 AllowedValues 列表启动产品时使用的 EC2 实例类型。

编辑完 InstanceType 参数后，此参数可能类似于以下示例：

```
    "InstanceType" : {
      "Description" : "EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro", "m1.small", "m1.medium", "m1.large",
"m1.xlarge", "m2.xlarge", "m2.2xlarge", "m2.4xlarge", "c1.medium", "c1.xlarge",
"c3.large", "c3.large", "c3.xlarge", "c3.xlarge", "c3.4xlarge", "c3.8xlarge" ],
      "ConstraintDescription" : "Must be a valid EC2 instance type."
    },
```

7. 在模板的 Mappings 部分中，编辑 AWSInstanceType2Arch 映射，使其仅包含受支持的 EC2 实例类型和架构。

   a. 通过删除 AllowedValues 参数的 InstanceType 列表中未包含的所有 EC2 实例类型来编辑映射的列表。

   b. 编辑将作为产品支持的架构类型的每个 EC2 实例类型的 Arch 值。有效值包括 PV64、HVM64 和 HVMG2。要了解产品支持的架构类型，请参阅 AWS Marketplace 中的产品详细信息页面。要了解 EC2 实例系列支持哪些架构，请参阅 Amazon Linux AMI 实例类型矩阵。

编辑完 AWSInstanceType2Arch 映射后，此映射可能类似于以下示例：

```
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "PV64"  },
      "m1.small"    : { "Arch" : "PV64"  },
      "m1.medium"   : { "Arch" : "PV64"  },
      "m1.large"    : { "Arch" : "PV64"  },
      "m1.xlarge"   : { "Arch" : "PV64"  },
      "m2.xlarge"   : { "Arch" : "PV64"  },
      "m2.2xlarge"  : { "Arch" : "PV64"  },
      "m2.4xlarge"  : { "Arch" : "PV64"  },
      "c1.medium"   : { "Arch" : "PV64"  },
      "c1.xlarge"   : { "Arch" : "PV64"  },
```

```
      "c3.large"    : { "Arch" : "PV64"  },
      "c3.xlarge"   : { "Arch" : "PV64"  },
      "c3.2xlarge"  : { "Arch" : "PV64"  },
      "c3.4xlarge"  : { "Arch" : "PV64"  },
      "c3.8xlarge"  : { "Arch" : "PV64"  }
    }
,
```

8. 在模板的 `Mappings` 部分中，编辑 `AWSRegionArch2AMI` 映射以将每个 AWS 区域与产品的相应架构和 AMI ID 关联。

   a. AWS Marketplace 中的产品完成页面上，查看产品对每个 AWS 区域使用的 AMI ID，如以下示例中所示：

Region	ID	
US East (N. Virginia)	ami-	Launch with EC2 Console
US West (Oregon)	ami-	Launch with EC2 Console
US West (N. California)	ami-	Launch with EC2 Console
EU (Frankfurt)	ami-	Launch with EC2 Console
EU (Ireland)	ami-	Launch with EC2 Console
Asia Pacific (Singapore)	ami-	Launch with EC2 Console
Asia Pacific (Sydney)	ami-	Launch with EC2 Console
Asia Pacific (Tokyo)	ami-	Launch with EC2 Console
South America (Sao Paulo)	ami-	Launch with EC2 Console

   b. 在模板中，删除您不支持的任何 AWS 区域的映射。

   c. 编辑每个区域的映射以移除不受支持的架构（`PV64`、`HVM64` 或 `HVMG2`）及其关联的 AMI ID。

   d. 对于每个其余的 AWS 区域和架构映射，请在 AWS Marketplace 中的产品详细信息页面中指定相应的 AMI ID。

编辑完 `AWSRegionArch2AMI` 映射后，您的节点可能类似于以下示例：

```
    "AWSRegionArch2AMI" : {
      "us-east-1"        : {"PV64" : "ami-nnnnnnnn"},
      "us-west-2"        : {"PV64" : "ami-nnnnnnnn"},
      "us-west-1"        : {"PV64" : "ami-nnnnnnnn"},
      "eu-west-1"        : {"PV64" : "ami-nnnnnnnn"},
      "eu-central-1"     : {"PV64" : "ami-nnnnnnnn"},
```

```
    "ap-northeast-1"   : {"PV64" : "ami-nnnnnnnn"},
    "ap-southeast-1"   : {"PV64" : "ami-nnnnnnnn"},
    "ap-southeast-2"   : {"PV64" : "ami-nnnnnnnn"},
    "sa-east-1"        : {"PV64" : "ami-nnnnnnnn"}
  }
```

现在您可以使用模板将产品添加到 AWS Service Catalog 产品组合。如果要进行其他更改，请参阅使用 AWS CloudFormation 模板以了解有关模板的更多信息。

将 AWS Marketplace 产品添加到 AWS Service Catalog 产品组合

1. 通过以下网址登录至 AWS Management Console 并导航至 AWS Service Catalog 管理员控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在产品组合页面上，选择要将 AWS Marketplace 产品添加到的产品组合。
3. 在产品组合详细信息页面上，选择上传新产品。
4. 键入请求的产品和支持详细信息。
5. 在 Version details 页面上，依次选择 Upload a template file、Browse 和模板文件。
6. 键入版本标题和描述。
7. 选择下一步。
8. 在审核页面上，确认摘要是正确的，然后选择确认并上传。产品将添加到产品组合。产品现在对有权访问产品组合的最终用户可用。

使用 AWS CloudFormation StackSets

> **Note**
>
> AutoTags 目前不支持AWS CloudFormation StackSets。

您可以使用AWS CloudFormation StackSets 在多个AWS 区域和账户中发布AWS Service Catalog产品。您可以指定产品在 AWS 区域 内部署的顺序。在账户之间，产品是并行部署的。在启动时，用户可以指定容错能力以及并行部署的最大账户数。有关更多信息，请参阅使用 AWS CloudFormation StackSets。

堆栈集和堆栈实例

利用堆栈集，您可使用一个 AWS CloudFormation 模板在 AWS 账户中跨 AWS 区域创建堆栈。

堆栈实例指的是 AWS 区域内的目标账户中的堆栈，并且仅与一个堆栈集相关联。

有关更多信息，请参阅 StackSets 概念。

堆栈集约束

在 AWS Service Catalog 中，您可以使用堆栈集约束来配置产品部署选项。

AWS Service Catalog支持两种产品的堆栈集限制AWS GovCloud (US) Regions：AWS GovCloud（美国西部）和 AWS GovCloud （美国东部）。

有关更多信息，请参阅 AWS Service Catalog 堆栈集约束。

管理预算

您可以使用 AWS 预算来跟踪 AWS Service Catalog 中的服务成本和使用情况。您可以将预算与 AWS Service Catalog 产品和产品组合相关联。

> **Note**
>
> AWS Service Catalog 不支持 Terraform 开源产品的预算。

借助 AWS 预算，您可以设置自定义预算，让系统在您的成本或使用量超过（或预测会超过）预算金额时提醒您。有关 AWS 预算的信息，请访问 https://aws.amazon.com/aws-cost-management/aws-budgets。

任务

- 先决条件
- 创建预算
- 关联预算
- 查看预算
- 取消关联预算

先决条件

在使用 AWS 预算之前，您需要在 AWS Billing and Cost Management 控制台中激活成本分配标签。有关更多信息，请参阅《AWS Billing and Cost Management 用户指南》中的激活用户定义的成本分配标签。

> **Note**
>
> 标签最多需要 24 小时才能激活。

您还需要为将使用“预算”功能的任何用户或组启用用户对 AWS Billing and Cost Management 控制台的访问权限。您可以通过为用户创建新策略来执行此操作。

要允许 用户创建预算，您还必须允许用户查看账单信息。如果要使用 Amazon SNS 通知，则可以为用户提供创建 Amazon SNS 通知的功能，如下面的策略示例所示。

创建预算策略

1. 打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择策略。
3. 在内容窗格中，选择创建策略。
4. 选择 JSON 选项卡，然后复制以下 JSON 策略文档中的文本。将该文本粘贴到 JSON 文本框中。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1435216493000",
            "Effect": "Allow",
            "Action": [
                "aws-portal:ViewBilling",
                "aws-portal:ModifyBilling",
                "budgets:ViewBudget",
                "budgets:ModifyBudget"
            ],
            "Resource": [
                "*"
            ]
```

```
        },
        {
            "Sid": "Stmt1435216552000",
            "Effect": "Allow",
            "Action": [
                "sns:*"
            ],
            "Resource": [
                "arn:aws:sns:us-east-1"
            ]
        }
    ]
}
```

5. 完成后，选择查看策略。策略验证程序将报告任何语法错误。
6. 在审核 页面上，为您的策略命名。查看策略摘要以查看您的策略授予的权限，然后选择创建策略以保存您的工作。

   新策略将显示在托管策略列表中，并且已准备好附加到您的用户和组。有关更多信息，请参阅《AWS Identity and Access Management 用户指南》中的创建和附加客户管理型策略。

创建预算

在 AWS Service Catalog 管理员控制台中，产品列表和产品组合页面列出了有关现有产品和产品组合的信息，并允许您对其进行操作。要创建预算，请先确定要将预算与哪个产品或产品组合相关联。

创建预算

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择产品列表或产品组合。
3. 选择要添加预算的产品或产品组合。
4. 打开操作菜单，然后选择创建预算。
5. 在预算创建 页面上，将一种标签类型与您的预算相关联。

   标签有两种类型： AutoTags 和 TagOptions。 AutoTags 确定推出产品的产品组合、产品和用户。 AWS Service Catalog自动将这些标签应用于已配置的资源。A TagOption 是管理员定义的键值对，在中进行管理。AWS Service Catalog

为了使产品组合或产品上的支出能够反映相关预算，它们必须具有相同的标签。请注意，首次使用的标签键可能需要 24 小时才能激活。有关更多信息，请参阅 the section called “先决条件”。

6. 选择在 AWS Budgets 中创建。您会被定向至设置预算页面。按照创建预算中的步骤继续进行预算设置。

> **Note**
>
> 创建预算后，您必须将其与产品或产品组合相关联。

关联预算

每个产品组合或产品可以有一个与之关联的预算。每个预算可以与多个产品组合和产品相关联。

当您将预算与产品或产品组合相关联时，您将能够从该产品或产品组合的详细信息页面查看有关预算的信息。为了使产品或产品组合上的支出能够反映在预算中，您必须为预算和产品或产品组合关联相同的标签。

> **Note**
>
> 如果您从 AWS Budgets 中删除预算，则与 AWS Service Catalog 产品和产品组合的现有关联仍然存在，但 AWS Service Catalog 将无法显示有关已删除预算的任何信息。

关联预算

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择产品列表或产品组合。
3. 选择您想要与预算关联的产品或产品组合。
4. 打开操作菜单，然后选择关联预算。
5. 在预算关联页面上，选择现有预算，然后选择继续。
6. 产品或产品组合表现在将包含您刚刚添加的预算的数据。

查看预算

如果预算与产品相关联，您可以在产品详细信息和产品列表页面上查看有关预算的信息。如果预算与产品组合相关联，您可以在产品组合和产品组合详细信息页面上查看有关预算的信息。

产品组合和产品列表页面显示现有资源的预算信息。您可以看到显示当前与预算和预测与预算的列。

当选择产品或产品组合时，您会被定向至详细信息页面。产品组合详细信息和产品详细信息页面中包含一些部分，记录了有关关联预算的详细信息。您可以查看预算金额、当前支出和预测支出。您还可以选择查看预算详细信息并编辑预算。

取消关联预算

您可以将预算与产品组合或产品取消关联。

> **Note**
>
> 如果从 AWS 预算中删除预算，则与 AWS Service Catalog 产品和产品组合的现有关联仍然存在，但 AWS Service Catalog 将无法显示有关已删除预算的任何信息。

要取消关联预算

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 选择产品列表或产品组合。
3. 选择要取消预算关联的产品或产品组合。
4. 选择操作。从下拉列表中选择取消关联预算。此时会显示确认提醒。
5. 在您确认要取消关联产品或产品组合的预算后，选择确认。

管理预配置产品

AWS Service Catalog 提供了一个用于管理预配置产品的界面。您可以基于访问级别查看、更新和终止您的目录中的所有预配置产品。有关示例流程，请参阅下面几节。

主题

- 以管理员的身份管理预配置产品
- 更改预配置产品的所有者
- 更新预配置产品的模板
- 教程：确定用户资源分配
- 管理 Terraform 开源产品状态错误
- 管理 Terraform 开源产品状态文件

以管理员的身份管理预配置产品

要管理账户的所有预配置产品，您需要具备 `AWSServiceCatalogAdminFullAccess` 或与预配置产品写入访问权限的同等 IAM 权限。有关更多信息，请参阅 Identity and Access Management AWS Service Catalog。

> **Tip**
>
> 对于静态预配置产品链接，在配置预配置产品之前，您必须在产品构件模板中引用预配置产品输出。有关更多信息及示例，请参阅以下内容：
>
> - AWS::ServiceCatalog::CloudFormationProvisionedProduct（在 AWS CloudFormation 用户指南中）。
> - DescribeProvisioningParameters (ProvisioningArtifactOutputKeys) 在《AWS Service Catalog开发者指南》中。

查看和管理所有预配置产品

1. 打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicequotas/。

   如果您已经登录到 AWS Service Catalog 控制台，请选择 Service Catalog，然后选择“最终用户”。

2. 如有必要，向下滚动到预配置产品部分。
3. 在预配置产品部分中，选择查看：列表，然后选择要查看的访问级别：用户、角色或账户。此操作会显示目录中的所有预配置产品。
4. 选择一个预配置产品进行查看、更新或终止。有关此方面信息的更多信息，请参阅查看预配置产品信息。

更改预配置产品的所有者

您可以随时更改预配置产品的所有者。您需要知道要设置为新所有者的用户或角色的 ARN。

默认情况下，此功能可供使用 `AWSServiceCatalogAdminFullAccess` 托管策略的管理员使用。您可以在 AWS Identity and Access Management (IAM) 中向最终用户授予 `servicecatalog:UpdateProvisionedProductProperties` 权限，从而为他们启用该功能。

更改预配置产品的所有者

1. 在 AWS Service Catalog 控制台中，选择预配置产品列表。
2. 找到要更新的预配置产品，然后选择其旁边的三个点并选择更改预配置产品所有者。您还可以在已配置产品的详情页面的操作菜单中找到更改所有者选项。
3. 在对话框中，输入要设置为新所有者的用户或角色的 ARN。ARN 以 `arn:` 开头并包含其他信息，这些信息之间用冒号或斜杠分隔，例如 `arn:aws:iam::123456789012:user/NewOwner`。
4. 选择提交。更新所有者后，您将看到一条成功消息。

另请参阅

- UpdateProvisionedProductProperties

更新预配置产品的模板

您可以将预配置产品的当前模板更改为其他模板。例如，如果您在 Service Catalog 中有一个 EC2 产品，则可以更新该 EC2 产品，保留相同的预配置产品 ID，但将模板更改为 S3 存储桶。

> **Note**
>
> 已预配置的 Terraform 开源或 Terraform Cloud 产品不支持更新模板。如果您想为现有 Terraform 产品使用不同的模板，则必须删除该产品，然后使用希望使用的模板创建新产品。

更新预配置产品模板

1. 从左侧的导航菜单中选择预配置产品。
2. 在预配置产品中，选择预配置产品，然后选择操作、更新。

   请注意，您也可以在预配置产品详细信息页面中选择操作、更新。
3. （可选）在产品详细信息中，选择更改产品。

   在更改产品中，请注意以下警告：

   更改产品会将此预配置产品更新为不同的产品模板。这可能会终止资源并创建新资源。

   您可以将预配置产品更新至同一产品中的其他版本。
4. （可选）在产品中，选择要更新至其他模板的产品。然后选择更改。

   在产品详细信息中，请注意以下警告：

   [Product name] 将从 [current template name] 更新为 [new template name]。但是，您的预配置产品的名称 [Provisioned Product name] 不会更改。

   您可以将预配置产品更新至同一产品中的其他版本。
5. 在产品版本中，选择所需的产品版本。
6. 在参数中，选择适当的参数。
7. 选择更新。

   在预配置产品详细信息中，您可以看到更新的详细信息。预配置的产品名称不会更改，但更新后预配置产品具有不同的模板。

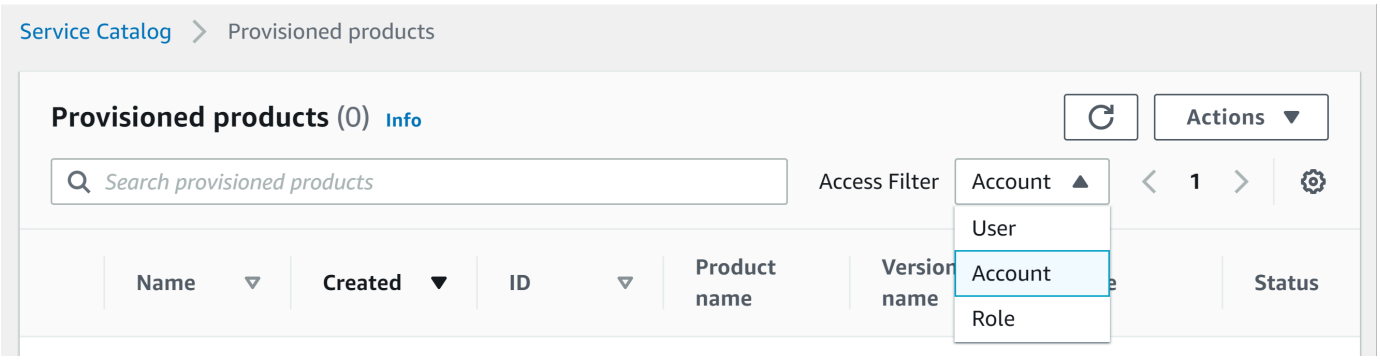
教程：确定用户资源分配

您可以借助 AWS Service Catalog 控制台确定预配置产品的用户及与产品关联的资源。本教程将帮助您将该示例转换为您自己的特定预配置产品。

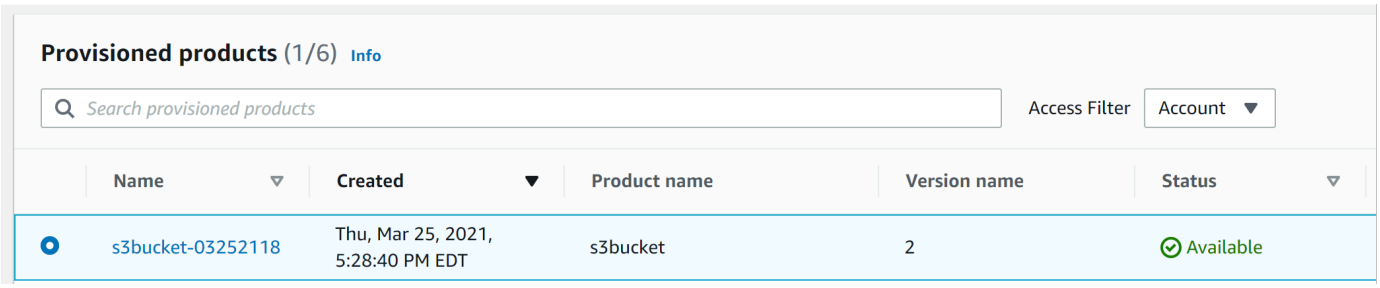
要管理账户的所有预配置产品，您需要具备 `AWSServiceCatalogAdminFullAccess` 或对预配置产品的同等写入访问权限。有关更多信息，请参阅《AWS Service Catalog用户指南》中的身份和访问权限管理。

确定预配置产品的用户和关联的资源

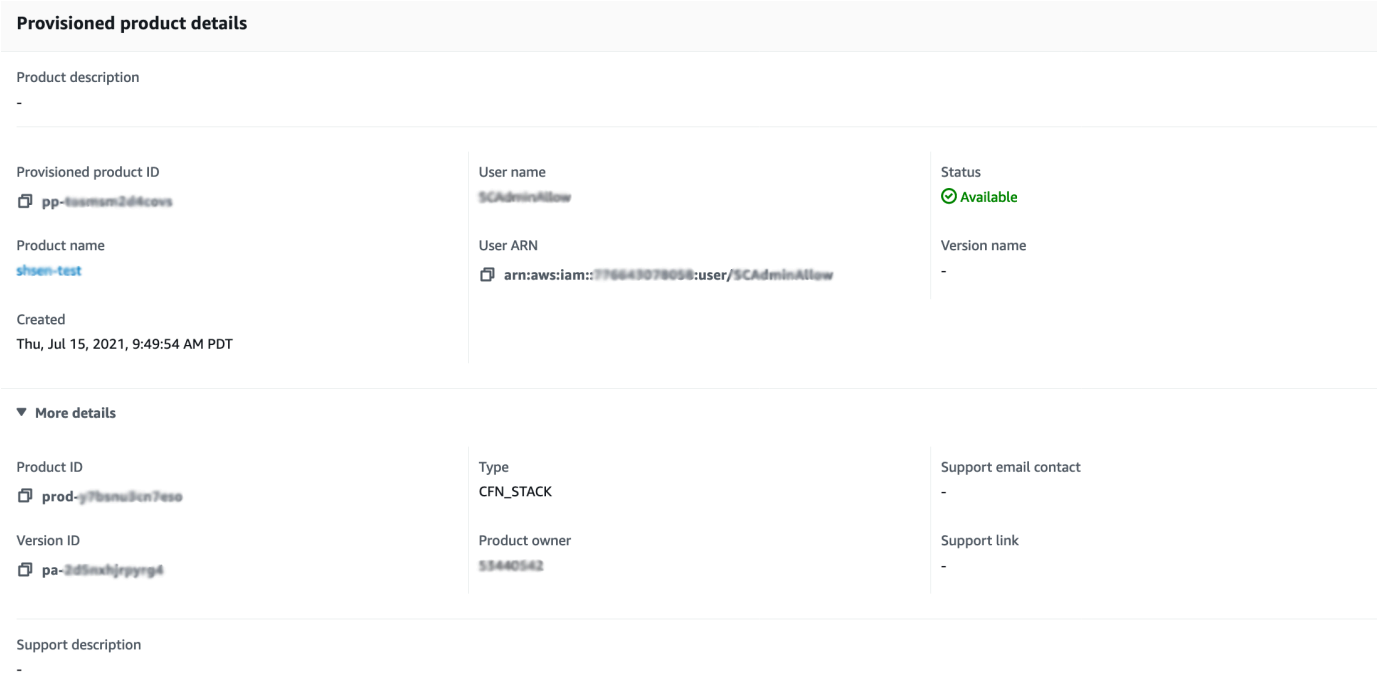
1. 打开 https://console.aws.amazon.com/servicecatalog。
2. 从左侧的导航菜单中选择预配置产品。
3. 在访问筛选条件下拉菜单中，选择账户。



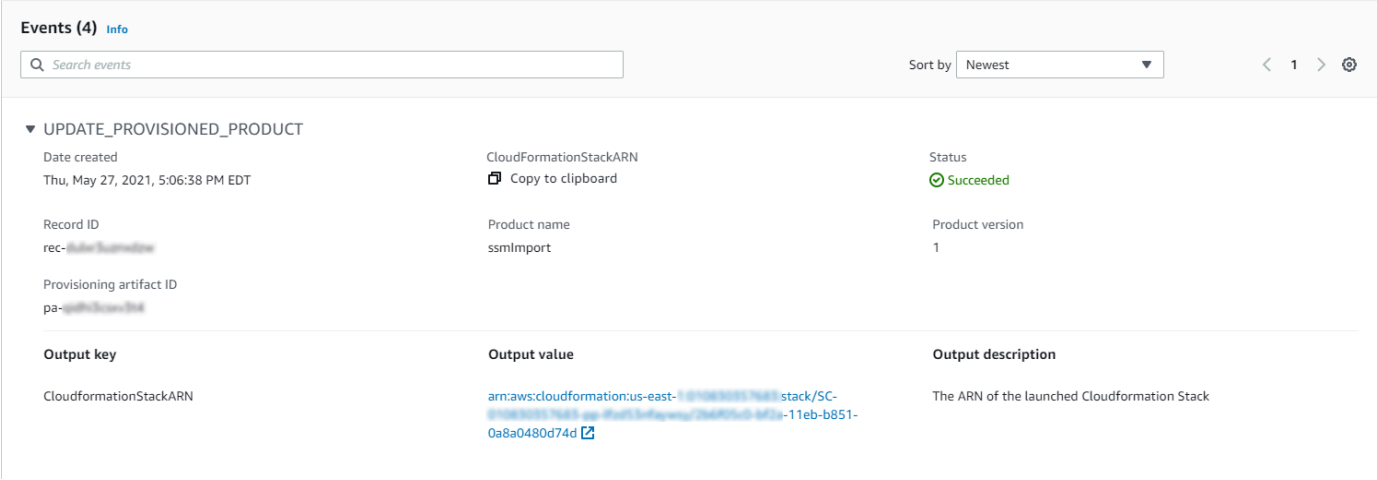
4. 在账户视图中，选择并打开预配置产品以显示其详细信息。



您可以查看预配置产品详细信息。



5. 向下滚动到事件部分。请注意 `Provisioned product ID` 和 `CloudformationStackARN` 的值。



6. 借助预配置产品 ID 确定与此启动对应的 AWS CloudTrail 记录及请求用户（这通常是您在联合身份验证时输入的电子邮件地址）。在本示例中，请求用户为“steve”。

```
{
  "eventVersion":"1.03","userIdentity":
  {
    "type":"AssumedRole",
    "principalId":"[id]:steve",
    "arn":"arn:aws:sts::[account number]:assumed-role/SC-usertest/steve",
    "accountId":[account number],
```

```
    "accessKeyId":[access key],
    "sessionContext":
    {
      "attributes":
      {
        "mfaAuthenticated":[boolean],
        "creationDate":[timestamp]
      },
      "sessionIssuer":
      {
        "type":"Role",
        "principalId":"AROAJEXAMPLELH3QXY",
        "arn":"arn:aws:iam::[account number]:role/[name]",
        "accountId":[account number],
        "userName":[username]
      }
    }
  },
  "eventTime":"2016-08-17T19:20:58Z","eventSource":"servicecatalog.amazonaws.com",
  "eventName":"ProvisionProduct",
  "awsRegion":"us-west-2",
  "sourceIPAddress":[ip address],
  "userAgent":"Coral/Netty",
  "requestParameters":
  {
    "provisioningArtifactId":[id],
    "productId":[id],
    "provisioningParameters":[Shows all the parameters that the end user entered],
    "provisionToken":[token],
    "pathId":[id],
    "provisionedProductName":[name],
    "tags":[],
    "notificationArns":[]
  },
  "responseElements":
  {
    "recordDetail":
    {
      "provisioningArtifactId":[id],
      "status":"IN_PROGRESS",
      "recordId":[id],
      "createdTime":"Aug 17, 2016 7:20:58 PM",
      "recordTags":[],
      "recordType":"PROVISION_PRODUCT",
```

```
        "provisionedProductType":"CFN_STACK",
        "pathId":[id],
        "productId":[id],
        "provisionedProductName":"testSCproduct",
        "recordErrors":[],
        "provisionedProductId":[id]
      }
    },
    "requestID":[id],
    "eventID":[id],
    "eventType":"AwsApiCall",
    "recipientAccountId":[account number]
  }
```

7. 使用 `CloudformationStackARN` 值识别 AWS CloudFormation 事件，以查找有关所创建资源的信息。您也可以使用 AWS CloudFormation API 获取此信息。有关更多信息，请参阅 AWS CloudFormation API 参考。

您可以使用 AWS Service Catalog API 或 AWS CLI 执行步骤 1 至 4。有关更多信息，请参阅AWS Service Catalog《开发人员指南》中的AWS Service Catalog命令行参考。

管理 Terraform 开源产品状态错误

Terraform Open Source `ProvisionProduct` 故障会路由到 `TAINTED` 状态，进而允许每个预配置的产品继续进行 `UpdateProvisionedProduct`。发生这种情况时：

- `UpdateProvisionedProduct` 不会尝试更新或更正标签，也不会尝试创建或修改资源组。
- `UpdateProvisionedProduct` 在决定是否应将预配置产品设置为 `AVAILABLE` 或 `TAINTED` 时，不会考虑先前配置操作的失败。

AWS Service Catalog 仅在 `ProvisionProduct` 期间应用标签。由于 `ProvisionProduct` 操作失败而导致的任何标记失败都不会自动解决。

状态错误示例

示例 1：AWS Service Catalog 在 **`ProvisionProduct`** 期间未创建资源组

在以下场景中，即使没有支持资源组，也没有对资源应用任何标签，您仍有预配置产品处于 `AVAILABLE` 状态。

1. 您的操作启动 ProvisionProduct。
2. Terraform 预置引擎会以工作流程故障对 ProvisionProduct 做出响应，但不提供 ResourceIdentifier。
3. ProvisionProduct 工作流程不会创建资源组，并随后将预配置的产品状态设置为 ERROR。
4. 然后，您可启动 UpdateProvisionedproduct 操作。
5. Terraform 预置引擎会做出表示“成功”的回复。
6. 因此，UpdateprovisionedProduct 工作流程将预配置产品的状态设置为 AVAILABLE，但不会创建资源组，也不会尝试应用任何标签。

示例 2：在 **UpdateProvisionedProduct** 期间 AWS Service Catalog 创建新资源

在以下场景中，即使新资源未应用任何标签，您仍有预配置产品处于 AVAILABLE 状态。

1. 您的操作启动 ProvisionProduct。
2. Terraform 预置引擎会做出表示“成功”的回复，并提供 ResourceIdentifier。
3. ProvisionProduct 工作流程创建资源组并将标签应用于所有已识别的资源。
4. 您启动 UpdateProvisionedProduct 一个创建新资源的新构件。
5. Terraform 预置引擎会做出表示“成功”的回复。
6. UpdateProvisionedProduct 工作流程将预配置产品状态设置为 AVAILABLE，但不会尝试将任何其他标签应用于新资源。

状态错误解决方案

AWS Service Catalog 确保为所有自 ProvisionProduct 设置为 TAINTED 的预配置产品创建资源组。如果 Terraform 预置引擎未返回 ResourceIdentifier，或者 AWS Service Catalog 创建资源组失败，则预配置产品将设置为 ERROR 状态，迫使您终止。

管理 Terraform 开源产品状态文件

每个 Terraform 开源预配置产品都有一个单状态文件。预配置产品与其状态文件之间存在一一对应关系。文件存储在名为 sc-terraform-engine-state-${AWS::AccountId}-${AWS::Region} 的 Amazon S3 存储桶中。状态文件保存在 AccountID 或 ProvisionedProductID 对象键下。

状态文件的访问权限仅限于 GetStateFile AWS Lambda 和 Amazon EC2 启动模板。AWS Service Catalog 管理员无法直接访问 Amazon S3 中的状态文件。管理员必须使用 Amazon EC2 访问这些文

件。默认情况下，AWS Service Catalog 管理员可以看到状态文件列表，但无法读取或写入文件内容。只有 Terraform 预置引擎可以读取或写入文件内容。

在 AWS Service Catalog 中管理标签

AWS Service Catalog 提供了标签，因此您可以对资源进行分类。标签有两种类型： AutoTags 和 TagOptions。

AutoTags 是标识中已置备资源来源信息的标签，AWS Service Catalog并自动应用于已置备AWS Service Catalog的资源。

TagOptions 是在中管理的键值对AWS Service Catalog，用作创建AWS标签的模板。

主题

- AWS Service Catalog AutoTags
- AWS Service Catalog TagOption 图书馆

AWS Service Catalog AutoTags

> **Note**
>
> AWS Service Catalog不支持 Terra AutoTags form 开源产品。

AutoTags 是标识中已置备资源来源信息的标签，AWS Service Catalog并自动应用于已置备AWS Service Catalog的资源。

AutoTags 包括产品组合、产品、用户、产品版本和预配置产品的唯一标识符的标签。这提供了一组标签，可反映客户在目录中配置的 AWS Service Catalog 结构。 AutoTags 不要计入买家的 50 个标签上限。

> **Note**
>
> AWS Service Catalog不支持 Terra AutoTags form 开源产品。

AWS Service Catalog AutoTags 可以帮助为您的资源提供一致的标记，这在为产品组合、产品或用户设置预算时非常有用。您还可以使用 AutoTags 来识别用于启动后操作（例如设置AWS Config规则）的资源。 AutoTags 对于您的预配置资源，可以在用于预配置的下游服务（例如 Amazon EC2 和 Amazon S3）的 “标签” 部分中查看。AWS CloudFormation

> **Note**
>
> AWS Service Catalog AutoTags 在您申请 AutoTags 已配置资源后不会更新。如果您将预配置产品更新为其他产品、预配置对象或新的启动路径，则现有产品 AutoTags 仍会显示原始值。

AutoTag 详情

- aws:servicecatalog:portfolioArn - 从中启动预配置产品的产品组合的 ARN。
- aws:servicecatalog:productArn - 从中启动预配置产品的产品的 ARN。
- aws: servicecatalog:-provisioningPrincipalArn 创建预配置产品的配置委托人（用户）的 ARN。
- aws: servicecatalog:-预配置provisionedProductArn的产品 AR N。
- aws: servicecatalog: provisioningArtifactIdentifier-原始配置工件（产品版本）的 ID。

AWS Service Catalog TagOption 图书馆

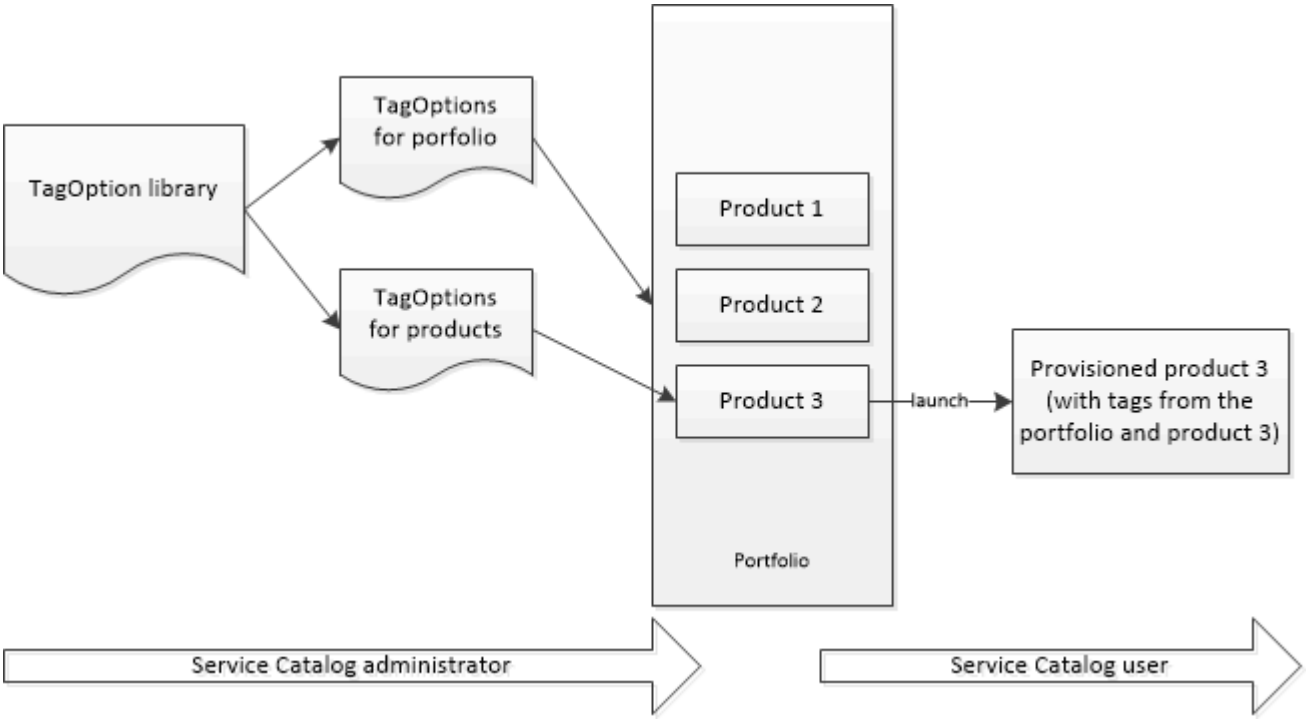
为了让管理员能够轻松管理已配置产品的标签，AWS Service Catalog提供了一个 TagOption 库。A TagOption 是在中管理的键值对。AWS Service Catalog它不是AWS标签，而是用作根据创建AWS标签的模板 TagOption。

AWS Service Catalog不支持 TagOptions Terraform 开源或 Terraform Cloud 产品。

该 TagOption 库使强制执行以下内容变得更加容易：

- 一致的分类
- 适当地标记 AWS Service Catalog 资源
- 为允许的标签定义用户可选的选项

管理员可以 TagOptions 与产品组合和产品关联。在产品发布（配置）期间，AWS Service Catalog汇总关联的产品组合和产品 TagOptions，并将其应用于预配置的产品，如下图所示。



使用该 TagOption 库，您可以停用 TagOptions 和保留它们与产品组合或产品的关联，并在需要时重新激活它们。这种方法不仅有助于维护库的完整性，还允许您管理 TagOptions 可能间歇性使用或仅在特殊情况下使用的库。

您可以使用控制 TagOptions AWS Service Catalog台或 TagOption 库 API 进行管理。有关更多信息，请参阅 Service Catalog API 参考。

内容

- 使用发布产品 TagOptions
- 管理 TagOptions
- TagOptions 与AWS Organizations标签策略一起使用

使用发布产品 TagOptions

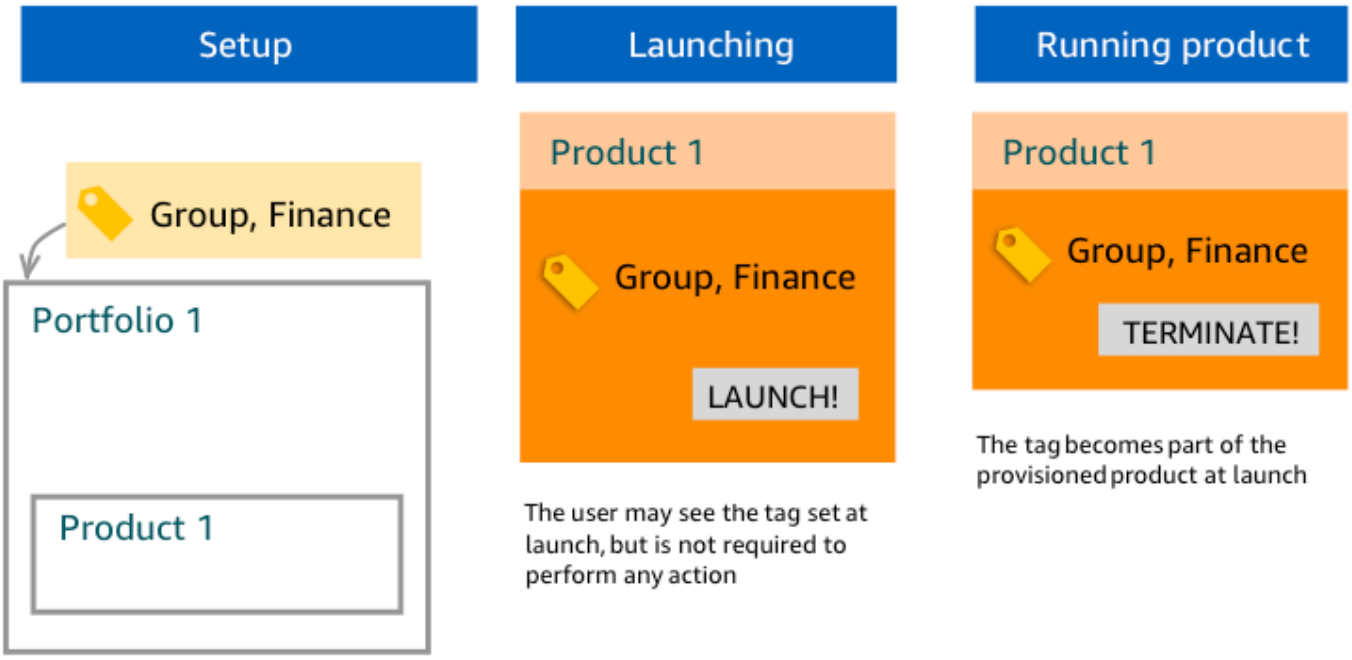
当用户启动的产品时 TagOptions，AWS Service Catalog会代表您执行以下操作：

- 收集产品和发布产品组合的所有 TagOptions 信息。
- 确保在预配置产品的标签中仅使用 TagOptions 具有唯一密钥的标签。用户获取一个键的多选值列表。在用户选择一个值后，它将成为预配置产品上的一个标签。
- 允许用户在预配置期间向产品添加不冲突的标签。

以下用例演示了启动期间 TagOptions 的工作原理。

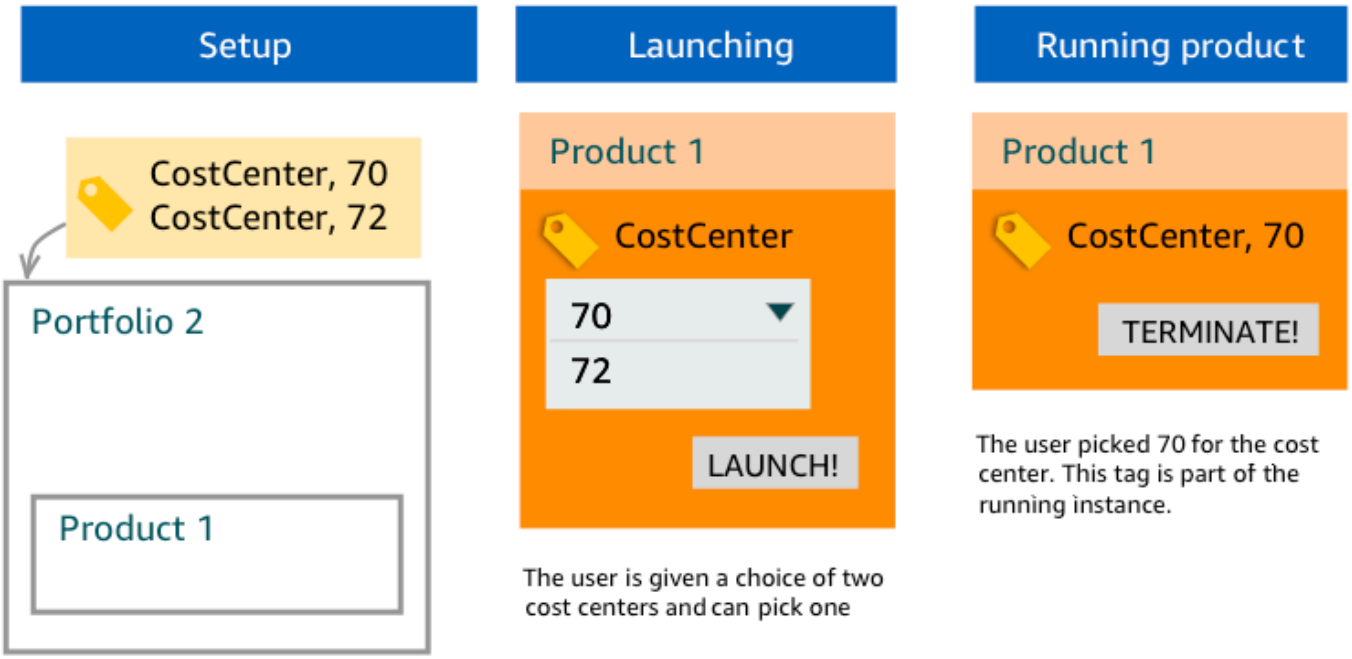
示例 1：唯一 TagOption 密钥

管理员创建 TagOption[Group=Finance] 并将其与 Portfolio1 关联起来，Product1 中有 Product1 而不是。 TagOptions当用户启动预配置产品时，单个产品将 TagOption 变为标签 [Group=Finance]，如下所示：



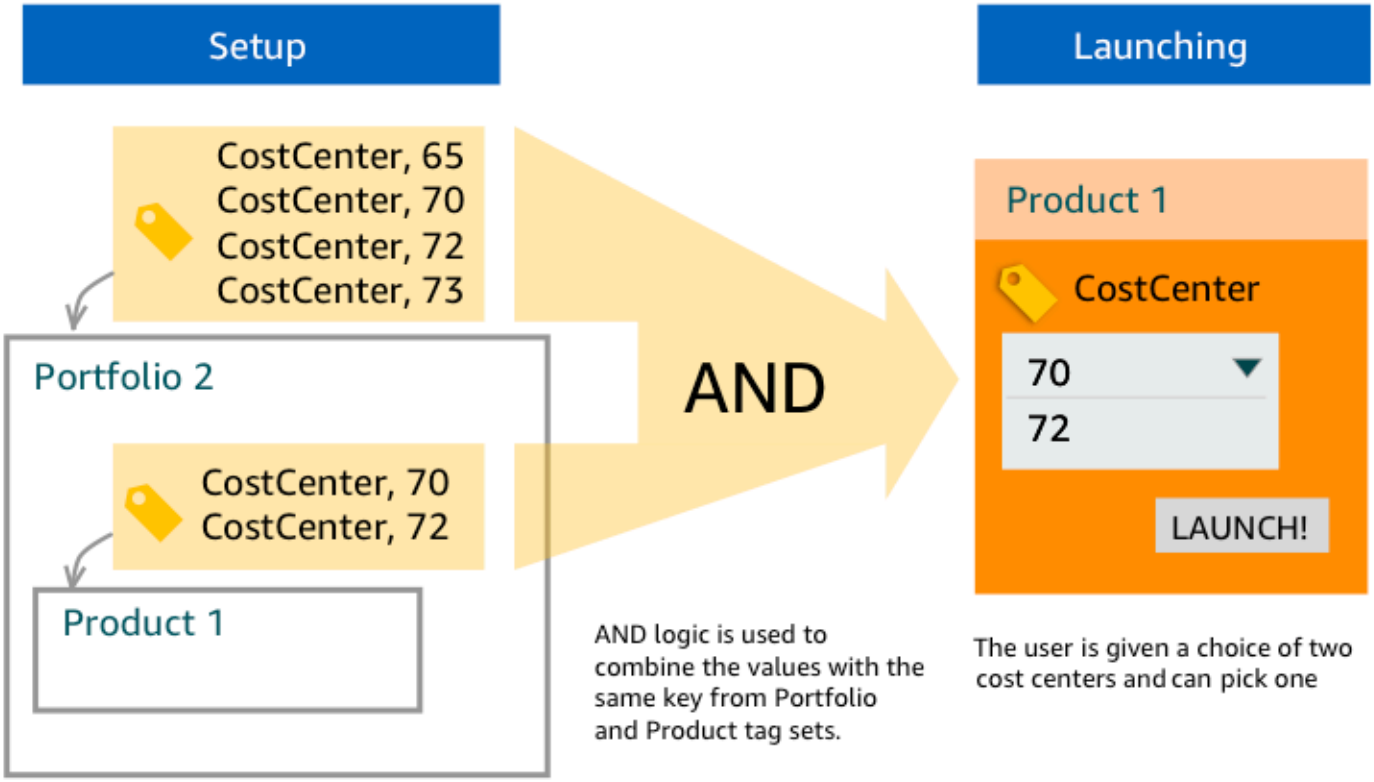
示例 2：投资组合上 TagOptions 具有相同密钥的集合

管理员在产品组合中放置了两个 TagOptions 具有相同密钥的产品，而在该产品组合中的任何产品上都没有 TagOptions 相同的密钥。在启动期间，用户必须选择与该键相关联的两个值之一。然后，预配置产品将使用该键和用户选择的值进行标记。



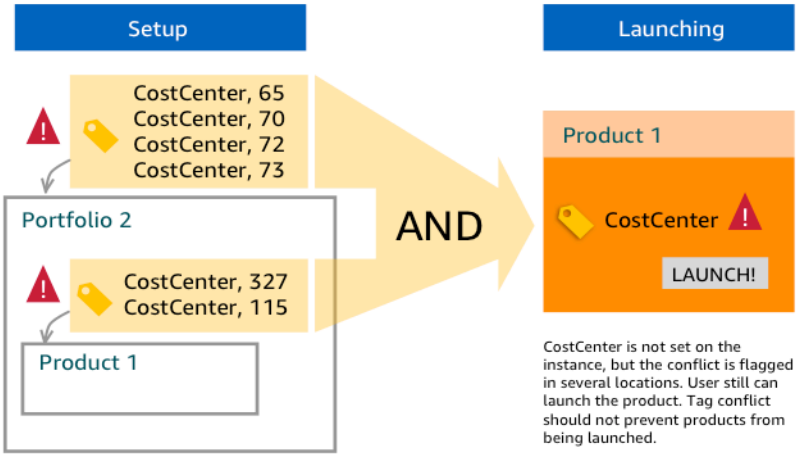
示例 3：一组在产品组合和该产品组合中的产品上都 TagOptions 具有相同密钥的

管理员在产品组合中放置了多个 TagOptions 具有相同密钥的产品，在该产品组合中，还有几个 TagOptions 使用相同密钥的产品上。 AWS Service Catalog根据的聚合（逻辑 AND 运算）创建一组值 TagOptions。当用户启动产品时，他或她会看到这组值并从中选择。预配置产品将使用该键和用户选择的值进行标记。



示例 4： TagOptions 具有相同密钥且值相互冲突的多个

管理员在产品组合中放置了多个 TagOptions 具有相同密钥的产品，在该产品组合中，还有几个 TagOptions 具有相同密钥的产品上。 AWS Service Catalog根据的聚合（逻辑 AND 运算）创建一组值 TagOptions。如果聚合找不到该键的值，AWS Service Catalog 将使用相同键和值 `sc-tagconflict-`*`portfolioid`*`-`*`productid` 创建一个标签，其中 *`portfolioid`* 和 *`productid`* 分别是产品组合和产品的 ARN。这可确保预配置产品使用正确的键以及管理员可以找到和更正的值进行标记。



管理 TagOptions

作为管理员，您可以在 TagOptions 库 TagOptions 中执行以下操作进行管理：

- 创建或删除
- 激活或停用
- 关联或取消关联
- 编辑

在控制台 TagOptions 中创建

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧导航菜单中，选择TagOptions资源库。
3. 在 “新建” 中 TagOption，输入键和值，然后选择 “添加”。

   创建新 TagOption 内容后，按键值对对其进行分组，并在列表中按字母顺序排序。TagOptions

   要 TagOption 使用 AWS Service Catalog API 创建，请参阅CreateTagOption。

在控制台 TagOptions 中删除

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧导航菜单中，选择TagOptions 库，然后选择操作。
3. 选择删除并确认删除。

在控制台中激活或停用一个或 TagOptions 多个

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧导航菜单中，选择TagOptions 库，然后选择操作。
3. 要激活，请选择 TagOption 您想要的非活动状态。然后选择操作并从下拉菜单中选择激活，然后确认您的选择。

   要停用，请选择 TagOption 所需的激活项。然后选择操作并从下拉菜单中选择停用，然后确认您的选择。

在控制台中将一个或多个产品组合 TagOptions 与投资组合关联或取消关联

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧导航菜单中，选择产品组合，然后打开要关联或取消关联的产品组合。
3. 选择TagOptions选项卡，然后选择一个或多个 TagOptions 要与投资组合关联或取消关联。
4. 选择操作。然后选择关联或取消关联并确认您的选择。

在控制台中将一个或多个产品 TagOptions 与产品关联或取消关联

1. 打开 AWS Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧导航菜单的管理项下，选择产品。然后打开您希望关联或取消关联的产品。
3. 选择TagOptions选项卡，然后选择一个或多个 TagOptions 要与投资组合关联或取消关联。
4. 选择操作。然后选择关联或取消关联并确认您的选择。

> **Note**
>
> 要使用 AWS Service Catalog API TagOptions 与产品组合或产品关联，请参阅AssociateTagOptionWithResource。
> 要 TagOptions 使用 AWS Service Catalog API 移除（取消关联），请参阅DisassociateTagOptionFromResource。

在控制台 TagOptions 中编辑的值

1. 通过以下网址打开 Service Catalog 控制台：https://console.aws.amazon.com/servicecatalog/。
2. 在左侧导航菜单中，选择TagOptions资源库。
3. 选择 a TagOption 并打开该值。（此值包含超链接。）然后选择编辑。
4. 在值字段中编辑此值并选择保存更改。

TagOptions 与AWS Organizations标签策略一起使用

本主题简要概述了 TagOptions 针对AWS Organizations和的标签策略AWS Service Catalog。它还就如何在同时使用这两个功能时防止标记冲突提出了建议。

TagOptions AWS Service Catalog适用于预配置产品（CloudFormation堆栈），而标签策略AWS Organizations适用于AWS账户和组织单位 (OU) 或组织根目录。例如，如果您将标签策略附加到 OU，则相同的标签策略将应用于该 OU 中的所有账户。如果您同时使用这两种标记功能，则应对其进行配置，避免其发生冲突。

标签策略

标签策略允许您在 AWS Organizations 中定义如何对账户中的 AWS 资源使用标签的规则。您可以使用标签策略来创建和维护在账户级别标记 AWS 资源的一致方法。

标签策略提供了一种简便方法，可确保用户应用一致的标签、审计已标记的资源并维护正确的资源分类。您还可以定义标签键的大写方式和允许的值。例如，您可以要求账户中的所有 EC2 实例必须将标签键设置为 **`CostCenter`** 并将该标签的值设置为 **`Data Insights`** 或 **`Marketing`**。

借助标签策略，您可以选择用于强制执行标记规则的选项，防止对标签进行不合规的操作，以及指定要应用强制措施的资源类型。如果您不选择强制执行选项，则标签策略允许您创建或转换不合规的标签，但会在 AWS Organizations 控制台中将其报告为不合规。

有关如何设置账户级别标记强制执行的更多信息，请参阅 AWS Organizations 中的标签策略。

TagOptions

TagOptions 是一项标记功能，AWS Service Catalog适用于 CloudFormation 堆栈级别的预配置产品（如果这些产品应用于关联产品）。 AWS Service Catalog提供了一个 TagOptions 库，您可以在其中定义要与您的AWS Service Catalog产品关联的键值对。启动AWS Service Catalog产品时，必须为与该产品组合或产品关联的现有 TagOption 密钥选择 TagOption 值才能启动该产品。由于您在产品组合或产品级别进行设置 TagOptions ，因此可以强制使用一致的分类法来标记跨账户和区域共享的产品组合。

有关如何在中进行设置的更多信息 TagOptions AWS Service Catalog，请参阅AWS Service Catalog TagOption 资源库。

避免AWS Organizations标签策略与之间的冲突 AWS Service Catalog TagOptions

如果您为组织中的账户配置 AWS Organizations 标签策略，我们建议您执行以下操作：

- 与同时管理AWS Service Catalog产品组合和产品的管理员共享对合规标签 TagOptions的要求。
- 与可能在 AWS Service Catalog 中启动产品并能在启动产品时附加可选的最终用户标签的最终用户共享对合规标签的要求。

假设您要在中启动使用AWS Service Catalog该 TagOption 密钥的产品`city`，并且您的标签策略要求标签密钥具有`city`美国城市的标签值，例如**`AtlantaSan Francisco`**、或**`Austin`**。 AWS Service Catalog不允许您在没有为产品所需 TagOption 密钥选择 TagOption 值的情况下启动产品。

在这种情况下，如果 TagOption 密钥的 TagOption 值`city`包含南美城市，例如**`Rio de Janeiro`**或**`Buenos Aires`**，则AWS Service Catalog不会启动该产品。相反，您必须在发布期间选择一个包含美国城市的 TagOption 值，以符合标签政策。

下表提供了一些场景，这些场景描述了如何解决您在使用标签策略时可能遇到的标签冲突问题。TagOptions

场景	Reason	解决方案
在标签策略中勾选了标签强制执行，由于标签不合规，产品无法启动。	TagOptions 使用您尚未添加到标签策略中允许的合规标签列表中的密钥和值进行指定。 添加了不符合您标签策略的可选自定义标签。	如果您在标签策略标签密钥大写强制中配置了特定的大写架构，请确保您的 TagOptions 标签密钥和可选的自定义标签密钥与您在标签策略中指定的内容一致。 请注意，如果在标签策略中取消选中标签密钥大小写强制复选框，则所有小写标签密钥都合规，并确保您的 TagOptions 标签密钥和可选的自定义标签密钥与您在标签策略中的要求一致（例如全部为小写）。
由于标签键大小写不合规，产品无法启动。	在 TagOptions 密钥中指定与标签策略大写强制规则不一致的大小写。	正确配置标签策略。如果您未指定标签键大小写合规性，则默认的标签键大小写均为小写。 此外，如果您未在标签策略中指定标签密钥大小写合规性，请确保中的 TagOptions 标签

场景	Reason	解决方案
		密钥全部AWS Service Catalog为小写以遵守执法规则。 如果您使用的标签策略未启用大小写合规性，则该标签策略仅会将全小写标签键视为合规。
由于标签值不兼容，产品无法启动。	为产品发布选择不在 TagOptions 标签策略标签值合规性允许列表中的标签值。	关联 TagOptions 您的产品和产品组合，这些产品和产品组合符合您在列表标签策略中的要求标签值合规性允许的标签值。

外部引擎用于 AWS Service Catalog

在中 AWS Service Catalog，外部引擎通过EXTERNAL产品类型表示。该EXTERNAL产品类型允许集成第三方配置引擎，例如 Terraform。您可以使用外部引擎将 Service Catalog 的功能扩展到原生 AWS CloudFormation 模板之外，从而允许使用其他基础设施即代码 (IaC) 工具。

该EXTERNAL产品类型允许您使用熟悉的 Service Catalog 界面管理和部署资源，同时利用所选 IaC 工具的特定功能和语法。

要在 Service Catalog 中启用EXTERNAL产品类型，您必须在账户中定义一组标准资源。这些资源被称为引擎。Service Catalog 在工件解析和置备操作的特定时刻将任务委派给引擎。

配置工件代表 Service Catalog 中产品的特定版本，允许您管理和部署一致的资源。

当您为EXTERNAL产品类型的配置对象调用 AWS Service CatalogDescribeProvisioningArtifact或DescribeProvisioningParameters操作时，Service Catalog 会调用引擎中的一个 AWS Lambda 函数。这是从提供的置备工件中提取参数列表并将其返回到所必需的 AWS Service Catalog。这些参数稍后将作为配置过程的一部分使用。

当您通过调用EXTERNAL配置对象时 ProvisionProduct，Service Catalog 首先在内部执行一些操作，然后向引擎中的 Amazon SQS 队列发送消息。接下来，引擎扮演提供的启动角色（您作为启动约束分配给产品的 IAM 角色），根据提供的配置项目配置资源，并调用 NotifyProvisionProductEngineWorkflowResultAPI 来报告成功或失败。

对UpdateProvisionedProduct和的调TerminateProvisionedProduct用处理方式类似，每个调用都有不同的队列和通知 API：

- NotifyProvisionProductEngineWorkflowResult
- NotifyUpdateProvisionedProductEngineWorkflowResult
- NotifyTerminateProvisionedProductEngineWorkflowResult.

主题

- 注意事项
- 参数解析
- 预置
- Updating
- 正在终止

- 标记

注意事项

每个中心账户只能使用一个外部引擎

每个 Service Catalog 中心账户只能使用一个EXTERNAL配置引擎。Service Catalog hub-and-spoke模型允许中心账户创建基准产品并共享产品组合，而分支账户则可以导入产品组合并利用产品。

此限制是因为EXTERNAL只能路由到一个账户中的一个引擎。如果管理员想要拥有多个外部引擎，则管理员必须在不同的中心帐户中设置外部引擎（以及产品组合和产品）。

外部引擎仅支持有启动限制的启动角色

EXTERNAL配置对象仅支持使用启动限制指定的启动角色进行置备。启动约束指定 Service Catalog 在最终用户启动、更新或终止产品时扮演的 IAM 角色。有关启动限制的更多信息，请参阅AWS Service Catalog 启动限制。

参数解析

EXTERNAL配置对象可以是任何格式。这意味着，在创建EXTERNAL产品类型时，引擎需要从提供的置备工件中提取参数列表并将其返回到 Service Catalog。这是通过在您的账户中创建一个 Lambda 函数来完成的，该函数可以接受以下请求格式、处理配置项目并返回以下响应格式。

> **Important**
>
> 必须为 Lambda 函数命名。`ServiceCatalogExternalParameterParser`

请求语法：

```
{
    "artifact": {
        "path": "string",
        "type": "string"
    },
    "launchRoleArn": "string"
}
```

字段	类型	必填	描述
项目	对象	是	要解析的工件的详细信息。
工件/路径	字符串	是	解析器下载工件的位置。例如，对于AWS_S3，这是亚马逊 S3 的 URI。
工件/类型	字符串	是	神器类型。允许的值：AWS_S3。
LaunchRole	字符串	否	下载项目时要扮演的启动角色的 Amazon 资源名称 (ARN)。如果未提供启动角色，则使用 Lambda 的执行角色。

响应语法：

```
{
    "parameters": [
        {
            "key": "string"
            "defaultValue": "string",
            "type": "string",
            "description": "string",
            "isNoEcho": boolean
        },
    ]
}
```

字段	类型	必填	描述
parameters	list	是	Service Catalog 要求最终用户在置备产品

字段	类型	必填	描述
			或更新预配置产品时提供的参数列表。如果在构件中未定义任何参数，则返回一个空列表。
key	字符串	是	参数键。
defaultValue	字符串	否	如果最终用户未提供值，则为参数的默认值。
type	字符串	是	引擎参数值的预期类型。例如，字符串、布尔值或映射。允许的值因每个引擎而异。Service Catalog 将每个参数值作为字符串传递给引擎。
description	字符串	否	参数的描述。建议将其设置为用户友好型。
isNoEcho	布尔值	否	确定参数值是否未在日志中回显。默认值为 false（参数值会被回显）。

预置

对于该ProvisionProduct操作，Service Catalog 将资源的实际配置委托给引擎。该引擎负责与您选择的 IaC 解决方案（例如 Terraform）接口，以配置工件中定义的资源。引擎还负责将结果通知 Service Catalog。

Service Catalog 会将所有配置请求发送到您的账户中名为的 Amazon SQS 队列。`ServiceCatalogExternalProvisionOperationQueue`

请求语法：

```
{
    "token": "string",
    "operation": "string",
    "provisionedProductId": "string",
    "provisionedProductName": "string",
    "productId": "string",
    "provisioningArtifactId": "string",
    "recordId": "string",
    "launchRoleArn": "string",
    "artifact": {
        "path": "string",
        "type": "string"
    },
    "identity": {
        "principal": "string",
        "awsAccountId": "string",
        "organizationId": "string"
    },
    "parameters": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ]
}
```

字段	类型	必填	描述
令牌	字符串	是	标识此操作的标记。必须将令牌返回到

字段	类型	必填	描述
			Service Catalog 才能通知执行结果。
operation	字符串	是	此字段必须`PROVISION_PRODUCT` 用于此操作。
provisionedProductId	字符串	是	预配置产品的 ID。
provisionedProductName	字符串	是	预配置产品的名称。
产品编号	字符串	是	产品的 ID。
provisioningArtifactId	字符串	是	置备工件的 ID。
recordId	字符串	是	此操作的 Service Catalog 记录的 ID。
launchRoleArn	字符串	是	用于配置资源的 IAM 角色的亚马逊资源名称 (ARN)。
项目	对象	是	定义资源配置方式的工件的详细信息。
工件/路径	字符串	是	引擎下载工件的位置。例如，对于`AWS_S3`，这是亚马逊 S3 的 URI。
工件/类型	字符串	是	神器类型。允许的值：`AWS_S3`。
identity	字符串	否	该字段目前未被使用。

字段	类型	必填	描述
parameters	list	是	用户输入到 Service Catalog 作为此操作输入的参数键值对列表。
标签	list	是	作为标签输入到 key-value-pairs Service Catalog 的用户列表，以应用于已配置的资源。

工作流程结果通知：

使用 NotifyProvisionProductEngineWorkflowResult API 详细信息页面上指定的响应对象调用 API。

Updating

对于该UpdateProvisionedProduct操作，Service Catalog 将资源的实际更新委托给引擎。引擎负责与您选择的 IaC 解决方案（例如 Terraform）接口，以更新工件中定义的资源。引擎还负责将结果通知 Service Catalog。

Service Catalog 将所有更新请求发送到您名为的账户中的亚马逊 SQS 队列。`ServiceCatalogExternalUpdateOperationQueue`

请求语法：

```
{
    "token": "string",
    "operation": "string",
    "provisionedProductId": "string",
    "provisionedProductName": "string",
    "productId": "string",
    "provisioningArtifactId": "string",
    "recordId": "string",
    "launchRoleArn": "string",
    "artifact": {
        "path": "string",
```

```
        "type": "string"
    },
    "identity": {
        "principal": "string",
        "awsAccountId": "string",
        "organizationId": "string"
    },
    "parameters": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ]
}
```

字段	类型	必填	描述
令牌	字符串	是	标识此操作的标记。必须将令牌返回到 Service Catalog 才能通知执行结果。
operation	字符串	是	此字段必须UPDATE_PROVISION_PRODUCT 用于此操作。
provisionedProductId	字符串	是	预配置产品的 ID。
provisionedProductName	字符串	是	预配置产品的名称。
产品编号	字符串	是	产品的 ID。

字段	类型	必填	描述
provisioningArtifactId	字符串	是	置备工件的 ID。
recordId	字符串	是	此操作的 Service Catalog 记录的 ID。
launchRoleArn	字符串	是	用于配置资源的 IAM 角色的亚马逊资源名称 (ARN)。
项目	对象	是	定义资源配置方式的工件的详细信息。
工件/路径	字符串	是	引擎下载工件的位置。例如，对于`AWS_S3`，这是亚马逊 S3 的 URI。
工件/类型	字符串	是	神器类型。允许的值：`AWS_S3`。
identity	字符串	否	该字段目前未被使用。
parameters	list	是	用户输入到 Service Catalog 作为此操作输入的参数键值对列表。
标签	list	是	作为标签输入到 key-value-pairs Service Catalog 的用户列表，以应用于已配置的资源。

工作流程结果通知：

使用 NotifyUpdateProvisionedProductEngineWorkflowResultAPI 详细信息页面上指定的响应对象调用 API。

正在终止

对于该TerminateProvisionedProduct操作，Service Catalog 将资源的实际终止委托给引擎。引擎负责与您选择的 IaC 解决方案（例如 Terraform）接口，以终止工件中定义的资源。引擎还负责将结果通知 Service Catalog。

Service Catalog 将所有终止请求发送到您的账户中名为的亚马逊 SQS 队列。`ServiceCatalogExternalTerminateOperationQueue`

请求语法：

```
{
    "token": "string",
    "operation": "string",
    "provisionedProductId": "string",
    "provisionedProductName": "string",
    "recordId": "string",
    "launchRoleArn": "string",
    "identity": {
        "principal": "string",
        "awsAccountId": "string",
        "organizationId": "string"
    }
}
```

字段	类型	必填	描述
令牌	字符串	是	标识此操作的标记。必须将令牌返回到 Service Catalog 才能通知执行结果。
operation	字符串	是	此字段必须`TERMINATE_PROVISIO`

字段	类型	必填	描述
			N_PRODUCT 用于此操作。
provisionedProductId	字符串	是	预配置产品的 ID。
provisionedProductName	字符串	是	预配置产品的名称。
recordId	字符串	是	此操作的 Service Catalog 记录的 ID。
launchRoleArn	字符串	是	用于配置资源的 IAM 角色的亚马逊资源名称 (ARN)。
identity	字符串	否	该字段目前未被使用。

工作流程结果通知：

使用 NotifyTerminateProvisionedProductEngineWorkflowResultAPI 详细信息页面上指定的响应对象调用 API。

标记

要通过 Resource Groups 管理标签，您的启动角色需要以下其他权限声明：

```
{
    "Effect": "Allow",
    "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:ListGroupResources"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
```

```
        "tag:GetResources",
        "tag:GetTagKeys",
        "tag:GetTagValues",
        "tag:TagResources",
        "tag:UntagResources"
    ],
    "Resource": "*"
}
```

> **Note**
>
> 启动角色还需要对构件中特定资源的标记权限，例如`ec2:CreateTags`。

AWS Service Catalog 中的监控

您可以使用 Amazon 监控您的AWS Service Catalog资源 CloudWatch，Amazon 会从中收集原始数据并将其处理AWS Service Catalog为可读的指标。这些统计数据会保存两周，从而使您能够访问历史信息，并能够更好地了解您服务的执行情况。AWS Service Catalog 指标数据以 1 分钟为间隔自动发送到 CloudWatch 。有关的更多信息 CloudWatch，请参阅 Amazon CloudWatch 用户指南。

有关可用指标和维度的列表，请参阅AWS Service Catalog CloudWatch 指标。

监控是保持 AWS Service Catalog 和您的 AWS 解决方案的可靠性、可用性和性能的重要方面。您应从 AWS 解决方案的所有部分收集监控数据，以便更轻松地调试出现的多点故障。在开始监控 AWS Service Catalog 之前，您应该创建一个监控计划，其中包括以下问题的答案：

- 监控目的是什么？
- 您将监控哪些资源？
- 监控这些资源的频率如何？
- 您将使用哪些监控工具？
- 谁负责执行监控任务？
- 出现错误时应通知谁？

监控工具

AWS 提供各种可以用来监控 AWS Service Catalog 的工具。您可以配置其中的一些工具来为您执行监控任务，但有些工具需要手动干预。建议您尽可能实现监控任务自动化。

自动监控工具

您可以使用 Amazon CloudWatch 警报来监控AWS Service Catalog和报告中断情况。

CloudWatch 警报在您指定的时间段内监视单个指标，并根据该指标在多个时间段内相对于给定阈值的值执行一项或多项操作。该操作是发送到亚马逊简单通知服务 (Amazon SNS) Simple Notification Scaling 主题或亚马逊 EC2 Auto Scaling 策略的通知。 CloudWatch 警报不会仅仅因为它们处于特定状态就调用操作；该状态必须已更改并保持了指定的时间段。要了解如何创建警报，请参阅创建 Amazon CloudWatch 警报。有关将 Amazon CloudWatch 指标与配合使用的更多信息AWS Service Catalog，请参阅AWS Service Catalog CloudWatch 指标。

AWS Service Catalog CloudWatch 指标

您可以使用 Amazon 监控您的AWS Service Catalog资源 CloudWatch，Amazon 会收集原始数据并将其处理AWS Service Catalog为可读的指标。这些统计数据会保存两周，从而使您能够访问历史信息，并能够更好地了解您服务的执行情况。AWS Service Catalog 指标数据以 1 分钟为间隔自动发送到 CloudWatch 。有关的更多信息 CloudWatch，请参阅 Amazon CloudWatch 用户指南。

主题

- 启用 CloudWatch 指标
- 可用指标和维度
- 查看 AWS Service Catalog 指标

启用 CloudWatch 指标

默认情况下，亚马逊 CloudWatch 指标处于启用状态。

可用指标和维度

下面列AWS Service Catalog出了发送给 Amazon CloudWatch 的指标和维度。

AWS Service Catalog 指标

`AWS/ServiceCatalog` 命名空间包括以下指标。

指标	描述
`ProvisionedProductLaunch`	在指定的时间段内为给定的产品和预配置项目启动的预配置产品的数量。 单位：计数 有效统计数据：Minimum、Maximum、Sum、Average

AWS Service Catalog 指标的维度

AWS Service Catalog向 Amazon 发送以下尺寸 CloudWatch。

维度	描述
`State`	该维度筛选您为以这一指定状态启动的所有预配置产品请求的数据。这样有助于您按启动状态给数据分类。 有效的状态：SUCCEEDED、FAILED
`ProductId`	此维度仅筛选您为已识别的产品 ID 请求的数据。这可帮助您准确找出从中启动的确切产品。
`ProvisioningArtifactId`	此维度仅筛选您为已识别的预配置项目 ID 请求的数据。这可帮助您准确找出从中启动的确切产品版本。

查看 AWS Service Catalog 指标

您可以在亚马逊控制台中查看亚马逊 CloudWatch 指标，该 CloudWatch 控制台可以精细且可自定义地显示您的资源以及服务中正在运行的任务数量。

主题

- 在 Amazon CloudWatch 控制台中查看AWS Service Catalog指标

在 Amazon CloudWatch 控制台中查看AWS Service Catalog指标

您可以在 Amazon CloudWatch 控制台中查看AWS Service Catalog指标。Amazon CloudWatch 控制台提供AWS Service Catalog指标的详细视图，您可以根据需要定制视图。有关亚马逊的更多信息 CloudWatch，请参阅亚马逊 CloudWatch 用户指南。

在 Amazon CloudWatch 控制台中查看指标

1. 打开亚马逊 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/。
2. 在左侧导航窗格的指标部分，选择服务目录。
3. 选择要查看的指标。

使用 AWS CloudTrail 记录 AWS Service Catalog API 调用

AWS Service Catalog与AWS CloudTrail一项服务集成，该服务提供用户、角色或AWS服务在中执行的操作的记录AWS Service Catalog。 CloudTrail 将所有 API 调用捕获AWS Service Catalog为事件。捕获的调用包含来自 AWS Service Catalog 控制台和代码的 AWS Service Catalog API 操作调用。如果您创建了跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括的事件AWS Service Catalog。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “事件历史记录” 中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向哪个请求发出AWS Service Catalog、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅《AWS CloudTrail用户指南》。

AWS Service Catalog信息在 CloudTrail

CloudTrail 在您创建AWS账户时已在您的账户上启用。当活动发生在中时AWS Service Catalog，该活动会与其他AWS服务 CloudTrail 事件一起记录在事件历史记录中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录 AWS 账户中的事件（包括 AWS Service Catalog 的事件），请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，在使用控制台创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Amazon S3 桶。此外，您可以配置其他AWS服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 创建跟踪记录概览
- AWS CloudTrail 支持的服务和集成
- 为 AWS CloudTrail 配置 Amazon SNS 通知
- 从多个区域接收 AWS CloudTrail 日志文件和从多个账户中接收 AWS CloudTrail 日志文件

CloudTrail 记录所有AWS Service Catalog操作。例如，调用`CreateProduct`和`UpdateProvisionedProduct`操作会在 CloudTrail 日志文件中生成条目。`CreatePortfolio`

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 AWS Identity and Access Management（IAM）用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。

- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 AWS Service Catalog 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。以下示例显示了演示 `CreateApplication` API 的 CloudTrail 日志条目。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "account",
        "arn": "arn:aws:iam::12345789012:user/dev-haw",
        "accountId": "12345789012",
        "accessKeyId": "keyId",
        "userName": "dev-haw"
    },
    "eventTime": "2020-09-23T21:07:58Z",
    "eventSource": "servicecatalog-appregistry.amazonaws.com",
    "eventName": "CreateApplication",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "205.251.233.48",
    "userAgent": "aws-cli/1.18.140 Python/3.6.11
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.63",
    "requestParameters": {
        "name": "hawTestCT",
        "clientToken": "6f36d650-a086-47cf-810a-fbfab2f8ad33"
    },
    "responseElements": {
        "application": {
            "applicationArn": "arn:aws:servicecatalog:us-
east-1:12345789012:application/app-02ocuq2cie2328pv64ya78e22f",
            "applicationId": "app-02ocuq2cie2328pv64ya78e22f",
            "creationTime": 1600895277.775,
            "lastUpdateTime": 1600895277.775,
            "name": "hawTestCT",
            "tags": {}
        }
```

```
    },
    "requestID": "1b6ad353-3b06-421b-bcb4-00075a782762",
    "eventID": "0a2ca224-cdfd-4c4b-a4ed-163218ff5e2d",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "12345789012"
}
```

控制台品牌首选项

AWS Service Catalog 允许管理员为账户指定控制台品牌首选项。管理员可以使用控制台品牌来为各种站点组件指定公司名称、徽标图像以及主要和强调（前景）颜色。管理员和最终用户在使用控制台时均可看到这些品牌首选项。

控制台品牌首选项可增强账号的外观并实现以下目标：

- 创建控制台和内部应用程序之间的无缝视觉过渡
- 区分同一公司内不同内部团队使用的账户
- 区分多个环境（例如开发、暂存或生产）中的账户

> **Note**
>
> 管理员在账户级别指定控制台品牌首选项。

要指定控制台品牌首选项

1. 在左侧导航菜单中，选择首选项。
2. 选择编辑，确定浅色模式或深色模式品牌首选项。
3. 上传徽标，输入品牌名称，然后选择主要颜色和强调颜色。
4. 选择保存。

有关 AWS Service Catalog 支持控制台品牌的区域列表，请查看AWS 区域对控制台品牌的支持。

AWS 区域 支持控制台品牌首选项

AWS Service Catalog 支持下表所列的 AWS 区域 控制台品牌首选项。

AWS 区域 名称	AWS 区域 身份
美国东部（弗吉尼亚州北部）	us-east-1
美国东部（俄亥俄州）	us-east-2

AWS 区域 名称	AWS 区域 身份
美国西部 (加利福尼亚北部)	us-west-1
美国西部（俄勒冈州）	us-west-2
非洲（开普敦）	af-south-1
亚太地区（香港）	ap-east-1
亚太地区（雅加达）	ap-southeast-3
亚太地区（孟买）	ap-south-1
亚太地区（大阪）	ap-northeast-3
亚太地区（首尔）	ap-northeast-2
亚太地区（新加坡）	ap-southeast-1
亚太地区（悉尼）	ap-southeast-2
亚太地区（东京）	ap-northeast-1
加拿大（中部）	ca-central-1
欧洲地区（法兰克福）	eu-central-1
欧洲地区（爱尔兰）	eu-west-1
欧洲地区（伦敦）	eu-west-2
欧洲地区（米兰）	eu-south-1
欧洲地区（巴黎）	eu-west-3
欧洲地区（斯德哥尔摩）	eu-north-1
中东（巴林）	me-south-1
南美洲（圣保罗）	sa-east-1

AWS 区域 名称	AWS 区域 身份
AWS GovCloud（美国东部）	us-gov-east-1
AWS GovCloud（美国西部）	us-gov-west-1

文档历史记录

下表描述了对的文档所做的重要更改 AWS Service Catalog。要获得本文档的更新通知，您可以订阅 RSS 源。

- API 版本：2014-11-12
- 最新文档更新：2024 年 5 月 16 日

变更	说明	日期
外部引擎 AWS Service Catalog	AWS Service Catalog 为外部引擎添加了新的文档。外部引擎通过`EXTERNAL`产品类型表示。该`EXTERNAL`产品类型允许集成第三方配置引擎，例如 Terraform。您可以使用外部引擎将 Service Catalog 的功能扩展到原生 AWS CloudFormation 模板之外，从而允许使用其他基础设施即代码 (IaC) 工具。有关更多信息，请参阅的外部引擎 AWS Service Catalog。	2024 年 5 月 16 日
安全 IAM 更新	AWS Service Catalog 将`AWSServiceCatalogSyncServiceRolePolicy` 政策更新`codestar-connections` 为`codeconnections` 。有关更多信息，请参阅适用于 AWS Service Catalog AppRegistry的AWS 托管策略。	2024 年 5 月 7 日

早期更新

下表描述了 2024 年 4 月 25 日 AWS Service Catalog 之前的文档发布历史。

功能	描述	发行日期
AWS Service Catalog	要了解 Hashicorp 对 Terraform 许可的更改以及对外部产品类型的更新，请查看将现有的 Terraform 开源产品和预配置产品更新为外部产品类型。	2023 年 10 月 20 日
AWS Service Catalog	要了解如何与之共享投资组合 AWS Organizations并 AWS Service Catalog 允许与之同步 AWS Organizations，请参阅AWSServiceCatalogOrgsDataSyncServiceRolePolicy策略和AWSServiceRoleForServiceCatalogOrgsDataSync服务相关角色。	2023 年 4 月 14 日
AWS Service Catalog	要了解如何管理 git-connected 产品以及 AWS Service Catalog 允许将外部存储库中的模板同步到您的 AWS Service Catalog 产品，请参阅AWSServiceCatalogSyncServiceRolePolicy策略和AWSServiceRoleForServiceCatalogSync服务相关角色。	2022 年 11 月 18 日
AWS Service Catalog AppRegistry	要了解如何 AppRegistry 帮助存储 AWS 应用程序、其关联的资源集合和应用程序属	2022 年 6 月 15 日

功能	描述	发行日期
	性组，请参阅AWS Service Catalog AppRegistry。	
AWS 服务管理连接器	要了解适用于 Jira 服务管理的连接器以及 ServiceNow，请参阅AWS 服务管理连接器。	2022 年 6 月 9 日
Jira 服务管理连接器	要了解有关 Jira 服务管理连接器的更新，请参阅AWS Jira 服务管理连接器。	2021 年 5 月 25 日
连接器用于 ServiceNow	要了解有关连接器的更新 ServiceNow，请参阅的AWS 服务管理连接器 ServiceNow。	2021 年 4 月 7 日
连接器用于 ServiceNow	要了解有关连接器的更新 ServiceNow，请参阅的AWS 服务管理连接器 ServiceNow。	2020 年 9 月 24 日
AWS 服务配额	要了解如何 AWS Service Catalog 使用 S AWS ervice Quotas，请参阅AWS Service Catalog 默认服务配额。	2020 年 3 月 24 日
入门库	要了解由 AWS Service Catalog提供的架构良好的产品模板库，请参阅 入门库	2020 年 3 月 10 日
版本指南	要了解有关产品版本指南的信息，请参阅版本指南。	2019 年 12 月 17 日
Jira 服务台连接器	要开始使用 Jira 服务台连接器，请参阅 AWS Jira 服务台服务管理连接器。	2019 年 11 月 21 日

功能	描述	发行日期
连接器用于 ServiceNow	要了解有关连接器的更新 ServiceNow，请参阅的AWS 服务管理连接器 ServiceNow。	2019 年 11 月 18 日
新增安全性章节	要了解中的安全性 AWS Service Catalog，请参阅中的安全 AWS Service Catalog。	2019 年 10 月 31 日
更改预配置产品的所有者	要了解如何更改预配置产品的所有者，请参阅更改预配置产品的所有者。	2019 年 10 月 31 日
新资源更新约束	要了解如何使用 RESOURCE_UPDATE 约束更新预配置产品中的标签，请参阅AWS Service Catalog 标签更新约束。	2019 年 4 月 17 日
连接器用于 ServiceNow	要开始使用连接器 ServiceNow，请参阅AWS 服务管理连接器 ServiceNow。	2019 年 3 月 19 日
Support AWS CloudFormation StackSets	要开始使用 AWS CloudFormation StackSets，请参阅使用 AWS CloudFormation StackSets。	2018 年 11 月 14 日
自助服务操作	要开始使用自助服务操作，请参阅AWS CloudFormation 服务操作。	2018 年 10 月 17 日
亚马逊 CloudWatch 指标	要了解有关 Amazon CloudWatch 指标的信息，请参阅AWS Service CatalogAmazon CloudWatch。	2018 年 9 月 26 日

功能	描述	发行日期
Support TagOptions	要管理标签，请参阅AWS Service Catalog TagOption资源库。	2017 年 6 月 28 日
导入产品组合	要导入从其他 AWS 账户共享的投资组合，请参阅导入投资组合。	2016 年 2 月 16 日
对权限信息的更新	要授予对最终用户控制台视图的访问权限，请参阅最终用户控制台访问权限。	2016 年 2 月 16 日
初始版本	这是《 AWS Service Catalog 管理员指南》的初始版本。	2015 年 7 月 9 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。