



管理员指南

AWS Service Catalog



AWS Service Catalog: 管理员指南

Copyright © 2026 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：简介 AWS Service Catalog	1
概述	2
Users	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
Permissions	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
设置步骤	7
注册获取 AWS 账户	7
向管理员授予权限	7
向最终用户授予权限	9
安装和配置 Terraform 预配置引擎	10
队列确定	10
将代理混淆添加到您的 Terraform 预配置引擎中	11
入门	15
入门库	15
先决条件	16
了解更多	16

CloudFormation 产品入门	16
步骤 1：下载模板	17
步骤 2：创建密钥对	21
步骤 3：创建产品组合	21
步骤 4：在产品组合中创建新产品	22
步骤 5：添加模板约束	23
步骤 6：添加启动约束	24
步骤 7：向最终用户授予产品组合的访问权限	26
步骤 8：测试最终用户体验	27
开始使用 Terraform 产品	28
更新为外部产品类型	29
先决条件：配置您的 Terraform 预置引擎	30
步骤 1：下载 Terraform 配置文件	31
步骤 2：创建 Terraform 产品	32
步骤 3：创建产品组合	33
步骤 4：将产品添加至产品组合	33
步骤 5：创建启动角色	34
步骤 6：添加启动约束	37
步骤 7：授予最终用户访问权限	38
步骤 8：与最终用户共享产品组合	39
步骤 9：测试最终用户体验	39
步骤 10：监控 Terraform 预配置操作	40
安全性	42
数据保护	42
利用加密来保护数据	43
身份和访问管理	44
受众	44
Identity-based 的策略示例 AWS Service Catalog	44
AWS 托管策略	50
使用服务关联角色	58
问题排查 AWS Service Catalog 身份和访问权限	63
控制访问权限	65
日志记录和监控	65
合规性验证	65
恢复能力	66
基础设施安全性	66

安全最佳实践	67
管理目录	68
管理产品组合	68
创建、查看和删除产品组合	69
查看产品组合详细信息	69
创建和删除产品组合	69
添加产品	70
添加约束	72
向用户授予访问权限	73
共享产品组合	74
共享和导入产品组合	81
管理产品	84
查看产品页面	84
创建产品	84
将产品添加到产品组合	87
更新产品	87
将产品与外部存储库中的模板文件进行同步	88
删除产品	96
管理版本	103
使用约束	104
启动约束	105
通知约束	110
标签更新约束	111
堆栈集约束	111
模板约束	112
使用服务操作	116
先决条件	116
步骤 1：配置最终用户权限	116
步骤 2：创建服务操作	118
步骤 3：将服务操作与产品版本关联	118
步骤 4：测试最终用户体验	119
步骤 5：使用管理服务操作 AWS CloudFormation	119
步骤 6：问题排查	119
使用 CloudFormation StackSets	121
堆栈集和堆栈实例	121
堆栈集约束	122

管理预算	122
先决条件	122
创建预算	124
关联预算	125
查看预算	125
取消关联预算	126
管理预配置产品	127
以管理员的身份管理预配置产品	127
更改预配置产品的所有者	128
另请参阅	128
更新预配置产品的模板	128
教程：确定用户资源分配	129
管理 Terraform 开源产品状态错误	133
状态错误示例	133
管理 Terraform 开源产品状态文件	134
管理标签	136
AutoTags	136
TagOption 图书馆	137
使用以下方式发布产品 TagOptions	138
管理 TagOptions	142
TagOptions 与 AWS Organizations 标签策略一起使用	143
外部引擎	147
注意事项	148
参数解析	148
预置	150
Updating	153
正在终止	156
标签	157
监控	159
监控工具	159
自动化工具	159
CloudWatch 指标	160
启用 CloudWatch 指标	160
可用指标和维度	160
查看 AWS Service Catalog 指标	161
CloudTrail 日志	162

AWS Service Catalog 信息在 CloudTrail	162
了解 AWS Service Catalog 日志文件条目	163
控制台品牌	165
AWS 区域 支持主机品牌推广	165
API 概述	168
产品发现	169
置备请求	170
预配置产品	170
预配置产品计划	171
产品组合	171
校长协会	172
产品	173
资源调配工件	174
约束	174
服务操作	175
TagOptions	175
AppRegistry	176
工作流程示例	178
文档历史记录	180
早期更新	181
.....	clxxxv

什么是 Service Catalog ？

Service Catalog 使组织能够创建和管理经批准的 IT 服务目录。AWS 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。

- Self-service 发现和发射

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。

- Fine-grain 访问控制

管理员从其目录中整理产品组合，添加要在配置时使用的限制和资源标签，然后通过 AWS Identity and Access Management (IAM) 用户和群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 管理控制台的替代方案，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：简介 AWS Service Catalog

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。

[简介 AWS Service Catalog](#)

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。

Users

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。目录管理员为产品准备 CloudFormation 模板、配置约束并管理 IAM 角色，以提供高级资源管理。
- 最终用户-从其 IT 部门或经理那里获得 AWS 证书，并使用 AWS 管理控制台 来发布他们已被授予访问权限的产品。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。产品由一个或多个 AWS 资源组成，例如 EC2 实例、存储卷、数据库、监控配置、联网组件或打包 AWS Marketplace 产品。产品可以是运行 AWS Linux 的单个计算实例、在自己的环境中运行的完全配置的多层 Web 应用程序，或者介于两者之间的任何东西。

您可以通过导入 AWS CloudFormation 模板来创建产品。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以配置安全组、创建密钥对和执行其他自定义操作时可以插入的参数。

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。您可以访问 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账户共享。AWS 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

AWS CloudFormation 堆栈使您可以将产品实例作为一个单元进行配置、标记、更新和终止，从而更轻松地管理产品的生命周期。AWS CloudFormation 堆栈包括以 JSON 或 YAML 格式编写的 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。

您还可以与其他 AWS 账户共享您的投资组合，并允许这些账户的管理员分配您的产品组合，但有其他限制，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。

Permissions

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 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

约束条件控制您可以为产品部署特定 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。AWS Service Catalog 约束条件有不同的类型：启动限制、通知限制和模板限制。

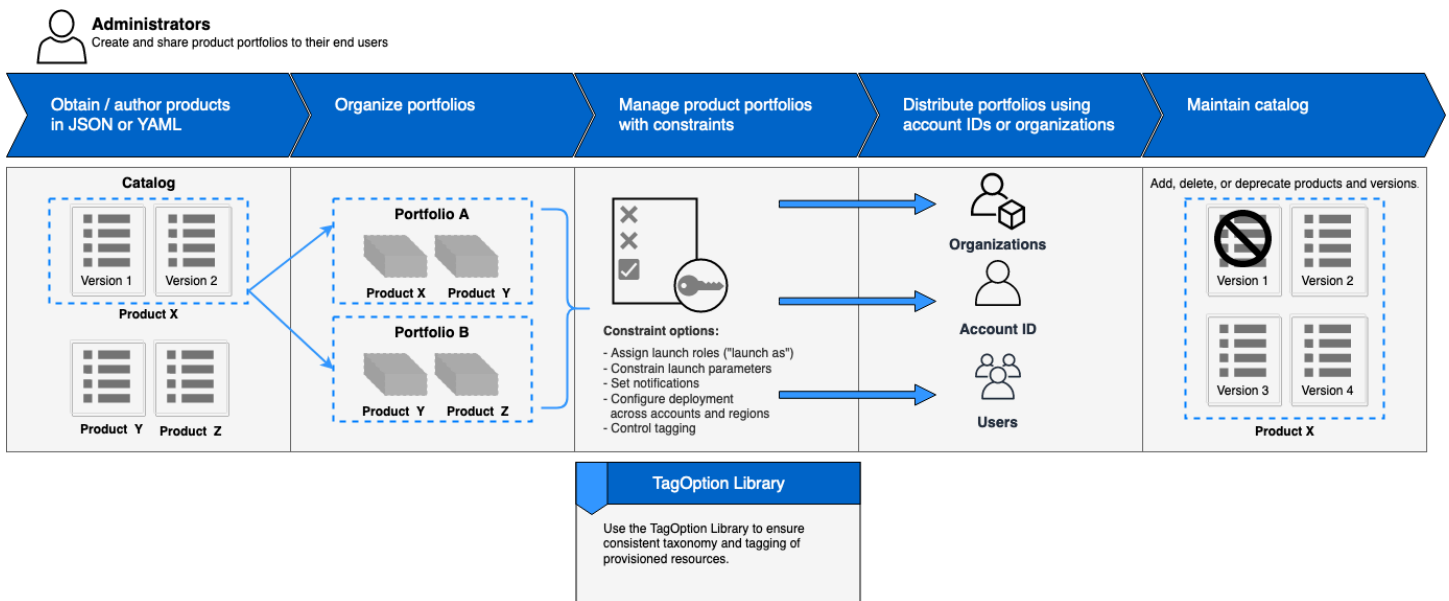
通过启动约束，您可为产品组合中的产品指定一个角色。使用此角色在启动时预配置资源，以便您可以限制用户权限，又不影响用户从目录预配置产品的能力。

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。

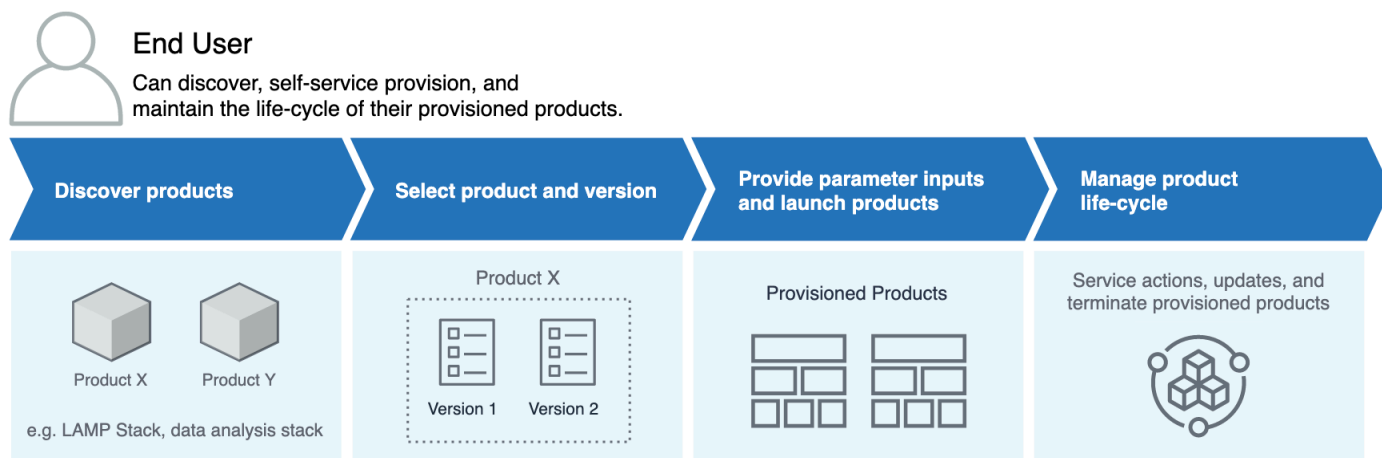
管理员初始工作流程

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 默认服务配额

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。

AWS Organizations

- AWS Service Catalog 每个组织的委派管理员：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

在开始使用之前 AWS Service Catalog，请完成以下任务。

设置步骤

主题

- [注册获取 AWS 账户](#)

注册获取 AWS 账户

要开始使用 AWS，你需要一个 AWS 账户。有关创建的信息 AWS 账户，请参阅《AWS 账户管理 参考指南》AWS 账户中的[入门](#)指南。

向 AWS Service Catalog 管理员授予权限

作为目录管理员，您需要访问 AWS Service Catalog 管理员控制台视图和允许您执行以下任务的 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
- 添加启动限制以定义最终用户启动产品时所 AWS Service Catalog 扮演的 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. 使用 <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 管理控制台 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》中的为 [AWS 资源添加标签](#)。

- 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 Amazon S3 授予访问 AWS Service Catalog 存储在 Amazon S3 中的模板的权限。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

在最终用户可以使用之前 AWS Service Catalog，您必须授予对 AWS Service Catalog 最终用户控制台视图的访问权限。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. 使用 <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 Users (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 预配置引擎

要成功使用 Terraform 产品 AWS Service Catalog，您必须在管理 Terraform 产品的同一个账户中安装和配置 Terraform 配置引擎。首先，您可以使用提供的 Terraform 配置引擎 AWS，该引擎安装和配置 Terraform 配置引擎所需的代码和基础架构。AWS Service Catalog 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

当您调用置备操作时，AWS Service Catalog 准备一条有效负载消息以发送到置备引擎中的相关队列。为了为队列构建 ARN，AWS Service Catalog 会进行以下假设：

- 预配置引擎位于产品所有者的账户中
- 配置引擎位于进行调用的 AWS Service Catalog 同一区域
- 预配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 000000000000 创建的产品 us-east-1 从账户 111111111111 调入，则假定 SQS ARN ProvisionProduct 是正确的。AWS Service Catalog arn:aws:sqs:us-east-1:000000000000:ServiceCatalogTerraform0SProvisionOperationQueue

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 预配置引擎中

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

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

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "servicecatalog.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:111122223333:function:ServiceCatalogTerraform0SParameterParser"
    }
  ]
}
```

这应该是与的集成正常运行所必需 AWS Service Catalog 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。向这些队列 AWS Service Catalog 发送消息时，使用配置账户的 ID AWS Service Catalog 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：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "servicecatalog.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:111122223333: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 提供的引擎创建的配置操作引入 Amazon SQS 队列的访问策略仅向服务主体授予跨账户 `sqs:SendMessage` (和关联的 KMS) 权限：AWS Service Catalog

JSON

```
{
  "Version": "2012-10-17",
  "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-1:111122223333:ServiceCatalogTerraform0SProvision0perationQueue"
    ]
},
{
    "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:ReEncryptFrom",
        "kms:ReEncryptTo",
        "kms:GenerateDataKey"
    ],
    "Resource": "arn:aws:kms:us-east-1:111122223333:key/key_id"
}
]
}

```

这应该是与的集成正常运行所必需 AWS Service Catalog 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。向这些队列 AWS Service Catalog 发送消息时，使用配置账户的 ID AWS Service Catalog 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：

JSON

```

{
    "Version": "2012-10-17",
    "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-1:111122223333: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:ReEncryptFrom",
      "kms:ReEncryptTo",
      "kms:GenerateDataKey"
    ],
    "Resource": "arn:aws:kms:us-east-1:111122223333: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：

- 一种在 Amazon Linux 上运行的云开发环境，其基础是定义产品可以使用的 AWS 资源的 CloudFormation 模板。
- 一种在 Terraform 配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](#)
- [CloudFormation 产品入门](#)
- [开始使用 Terraform 产品](#)

入门库

AWS Service Catalog 提供了一个包含架构良好的产品模板的入门库，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 CloudFormation 模板所需的权限。有关更多信息，请参阅[使用控制访问权限 AWS Identity and Access Management](#)。
- 管理 AWS Service Catalog 所需的权限。有关更多信息，请参阅 [the section called “身份和访问管理”](#)。

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](#)。

CloudFormation 产品入门

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，基于定义产品可以使用的 AWS 资源的 CloudFormation 模板。

Note

开始之前，请确保您已完成 [设置 AWS Service Catalog](#) 中的步骤。

主题

- [第 1 步：下载 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](#)

第 1 步：下载 CloudFormation 模板

您可以使用 CloudFormation 模板来配置和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 CloudFormation 模板格式。您可以使用自己选择的 AWS CloudFormation 编辑器或文本编辑器来创建和保存模板。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 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可在 development <https://awsdocs.s3.amazonaws.com/servicecatalog/-envirodevelopment-environment.template> 上找到。

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](#)。

该`KeyName`参数允许您指定最终用户在使用 AWS Service Catalog 启动您的产品时必须提供的亚马逊弹性计算云 (Amazon EC2) 密钥对名称。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::Interface](#) 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。上面的模板包括区域列表以及与每个 AWS 区域对应的 Amazon 系统映像 (AMI)。AWS Service Catalog 使用此映射根据用户在中选择的 AWS 区域来确定要使用的 AMI AWS 管理控制台。
- 资源 (必需) - 堆栈资源及其属性。您可引用模板的资源 and 输出部分中的资源。在上面的模板中，我们指定了一个运行 Amazon Linux 的 EC2 实例和一个允许 SSH 访问该实例的安全组。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。

CloudFormation 使用当前 AWS 区域从先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《CloudFormation 用户指南》中的[模板参考](#)。

步骤 2：创建密钥对

要使您的最终用户能够启动基于本教程示例模板的产品，您必须创建一个 Amazon EC2 key pair。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，然后查看[亚马逊 EC2 用户指南中的亚马逊 EC2 密钥对](#)。

本教程的 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. 打开亚马逊 EC2 控制台，网址为<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. 打开 Service Catalog 控制台，网址为<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，请打开 <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 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 [T2 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

启动约束指定在最终用户启动产品时 AWS Service Catalog 扮演的 IAM 角色。

在此步骤中，您将向 Linux 桌面产品添加启动限制，以便 AWS Service Catalog 可以使用构成产品 AWS CloudFormation 模板的 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. 在服务拥有的 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，该 `ServiceCatalogEndUserAccess` 策略授予访问 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`— 允许创建、读取、更新、删除、列出和标记 CloudFormation 堆栈的 AWS Service Catalog 完全权限。
 - `ec2`— 允许列出、读取、写入、配置和标记作为产品一部分的亚马逊弹性计算云 (Amazon EC2) 资源的 AWS Service Catalog 全部权限。AWS Service Catalog 根据您要部署的 AWS 资源，此权限可能会发生变化。

- 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— 允许访问拥有的 Amazon S3 存储桶。AWS Service Catalog 要部署产品，AWS Service Catalog 需要访问配置对象。
- servicecatalog— 允许 AWS Service Catalog 代表最终用户列出、读取、写入、标记和启动资源。
- sns— 允许根据启动限制列出、读取、写入和标记 Amazon SNS 主题的 AWS Service Catalog 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。

JSON

```
{
  "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. 在“选择可信实体”中，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. 选择下一步。
 - d. 对于角色名称，键入 **linuxDesktopLaunchRole**。
 - e. 选择创建角色。
8. 在 <https://console.aws.amazon.com/servicecatalog> 上打开控制台。
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

要验证最终用户能否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 AWS 身份登录并执行这些任务。

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，选择您创建 Engineering Tools 投资组合的 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/>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。启动该产品需要几分钟。AWS Service Catalog 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 AWS Service Catalog 将其用作单一工具，在内部大规模组织、管理和分发 Terraform 配置。AWS 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、标记以及与其他账户共享。AWS 在中 AWS Service Catalog，您的最终用户可以看到他们有权访问的产品和版本的简单列表，然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](#)。

您在教程中创建的 Terraform 产品中包含的 AWS 资源是一个简单的 Amazon S3 存储桶。

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](#)
- [第 3 步：创建 AWS Service Catalog 投资组合](#)
- [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](#)
- [步骤 8：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 Terraform 参考引擎 AWS Service Catalog，使其包括对外部和 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

作为在中创建 Terraform 产品的先决条件 AWS Service Catalog，您必须在您的 Service Catalog 管理员帐户（中心帐户）中安装和配置配置引擎。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

您可以使用 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，以便您可以开始使用。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

示例配置文件的文本如下：

```
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

配置文件声明置备产品时 AWS Service Catalog 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）- 用于资源配置的云服务提供商，即AWS。AWS Service Catalog 仅支持AWS作为提供者。因此，Terraform 预配置引擎会覆盖任何其他列出的提供商，预配置为 AWS。
- 资源（必需）- 用于配置 AWS 的基础架构资源。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架构工作流程。

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. 在打开 AWS Service Catalog 控制台<https://console.aws.amazon.com/servicecatalog/>并以管理员用户身份登录。
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。

第 3 步：创建 AWS Service Catalog 投资组合

您可以在 AWS Service Catalog 管理员帐户（中心帐户）中创建产品组合，以便于产品组织和分发给最终用户帐户（分支帐户）。

创建产品组合

1. 在打开 AWS Service Catalog 控制台 <https://console.aws.amazon.com/servicecatalog/> 并以管理员身份登录。
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：

- 访问您的 Terraform 产品的底层 AWS 资源。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。
- 在 AWS Service Catalog 自有的 Amazon S3 存储桶中读取对亚马逊 S3 模板的访问权限
- 对 CreateGroup、ListGroupResources、DeleteGroup 和 Tag 资源组操作的访问权限。这些操作 AWS Service Catalog 允许管理资源组和标签

在 AWS Service Catalog 管理员帐户中创建启动角色

1. 登录 AWS Service Catalog 管理员账户后，按照 IAM 用户指南中 [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— 允许列出、读取、写入、配置和标记 Amazon S3 产品的 AWS Service Catalog 全部权限。
 - s3— 允许访问拥有的 Amazon S3 存储桶。AWS Service Catalog 要部署产品，AWS Service Catalog 需要预配置构件的访问权限。

- `resourcegroups`— AWS Service Catalog 允许创建、列出、删除和标记 AWS Resource Groups。
- `tag`— 允许 AWS Service Catalog 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：

JSON

```
{
  "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：
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 管理员帐户中创建启动角色后，还必须在 AWS Service Catalog 最终用户帐户中创建相同的启动角色。最终用户帐户中的角色必须与管理员帐户中的角色同名并包含相同的策略。

在 AWS Service Catalog 最终用户帐户中创建启动角色

1. 以管理员身份登录最终用户帐户，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. 重复上述在 AWS Service Catalog 管理员帐户中创建启动角色中的步骤 2-10。

Note

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

现在您已经在管理员帐户和最终用户帐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。

在管理员帐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. 在 <https://console.aws.amazon.com/service> 的 AWS Service Catalog catalog 上打开控制台。
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-s 3Product。
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。当您共享投资组合时，请 AWS Service Catalog 验证这些委托人名称是否已经存在。如果确实存在，则 AWS Service Catalog 自动将匹配的 IAM 委托人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。

步骤 8：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，您将使用管理员帐户（中心帐户）与组织共享您的产品组合，该帐户也是组织的管理帐户。

从管理员中心帐户共享产品组合

1. 打开 AWS Service Catalog 控制台，网址为 <https://console.aws.amazon.com/servicecatalog/>。
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，选择最终用户帐户（分支帐户）。

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

要验证最终用户能否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 AWS 身份登录并执行以下任务。

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/> 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。启动该产品需要几分钟。AWS Service Catalog 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 创建名为的新 Amazon S3 存储桶**terraform-s3-product**。

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— 在配置新的 Terraform 产品和更新现有 Terraform 预配置产品时 AWS Service Catalog 调用此状态机。
- TerminateProvisionedProductStateMachine— 在终止现有 Terraform 预配置产品时 AWS Service Catalog 调用此状态机。

要执行监控状态机

1. 打开 AWS 管理控制台，在安装了 Terraform 配置引擎的管理员中心帐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

AWS Service Catalog 在配置产品时创建预配置的产品 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 AWS 是重中之重。作为 AWS 客户，您可以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。

安全是双方共同承担 AWS 的责任。[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 负责保护在 AWS 云中运行 AWS 服务的基础架构。AWS 还为您提供可以安全使用的服务。Third-party 作为[AWS 合规计划](#)的一部分，审计师定期测试和验证我们安全的有效性。

要了解适用于的合规计划 AWS Service Catalog，请参阅[按合规计划划分的范围内的AWS 服务](#)

- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您的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时如何应用分担责任模型 AWS Service Catalog。以下主题向您介绍如何进行配置 AWS Service Catalog 以满足您的安全和合规性目标。还将向您介绍其他可帮助您监控和保护 AWS Service Catalog 资源的 AWS 服务。

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](#)
- [的合规性验证 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

AWS [责任共担模式](#)适用于保护 AWS Service Catalog 中的数据。如本模型所述 AWS，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题解答AWS](#)条款。有关欧洲数据保护的信息，请参阅[通用数据保护条例 \(GDPR\) 中心](#)。

出于数据保护目的，我们建议您保护 AWS 账户凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。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)。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的[使用跟 CloudTrail 踪](#)。
- 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 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 (FIPS) 第 140-3 版》<https://aws.amazon.com/compliance/fips/>。

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）。这包括您使用控制台、API AWS Service Catalog 或 SDK 或以其他 AWS 服务方式使用控制台 AWS CLI、API 或 AWS SDK 的情况。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。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

利用加密来保护数据

静态加密

AWS Service Catalog 使用使用密钥进行静态加密的 Amazon S3 存储桶和亚马逊 DynamoDB 数据库。Amazon-managed 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 AWS 客户端加密。

您可以通过创建 VPC 终端节点从您的亚马逊虚拟私有云（亚马逊 VPC）私有访问 AWS Service Catalog API。使用 VPC 终端节点，VPC 和 AWS Service Catalog VPN 之间的路由由 AWS 网络处理，无需互联网网关、NAT 网关或 VPN 连接。

使用的最新一代 VPC 终端节点 AWS Service Catalog 由一项 AWS 技术提供支持 AWS PrivateLink，该技术使用弹性网络接口实现 AWS 服务之间的私有连接，VPC 中的私有 IP。

Identity and Access Management AWS Service Catalog

访问 AWS Service Catalog 需要凭证。这些证书必须具有访问 AWS 资源（例如产品 AWS Service Catalog 组合或产品）的权限。AWS Service Catalog 与 AWS Identity and Access Management (IAM) 集成，使您能够向 AWS Service Catalog 管理员授予他们创建和管理产品所需的权限，并授予 AWS Service Catalog 最终用户启动产品和管理预配置产品所需的权限。这些策略可以由 AWS 管理员和最终用户创建和管理，也可以由管理员和最终用户单独创建和管理。要控制访问权限，您需要将这些策略附加到用于 AWS Service Catalog 的用户、组和角色。

受众

您通过 AWS Identity and Access Management (IAM) 拥有的权限可能取决于您在 AWS Service Catalog 中所扮演的角色。

您通过 AWS Identity and Access Management (IAM) 拥有的权限可能取决于您在 AWS Service Catalog 中所扮演的角色。

管理员-作为 AWS Service Catalog 管理员，您需要拥有管理员控制台的完全访问权限和 IAM 权限，这样您才能执行诸如创建和管理产品组合和产品、管理限制以及向最终用户授予访问权限之类的任务。

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

IAM 管理员 – 如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 AWS Service Catalog 的访问权限的详细信息。要查看您可以在 IAM 中使用的 AWS Service Catalog 基于身份的策略示例，请参阅 [the section called “AWS 托管策略”](#)

Identity-based 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** 策略授予对 AWS Service Catalog 最终用户控制台视图的访问权限。当拥有其中任一策略的用户 AWS Service Catalog 在中进行选择时 AWS 管理控制台，最终用户控制台视图将显示他们有权启动的产品。

在最终用户成功启动您授予访问权限的产品之前，您必须向他们提供其他 IAM 权限，以允许他们使用产品 AWS CloudFormation 模板中的每个底层 AWS 资源。AWS Service Catalog 例如，如果产品模板包含 Amazon Relational Database Service (Amazon RDS)，则您必须向用户授予用于启动产品的 Amazon RDS 权限。

要了解如何允许最终用户启动产品，同时强制执行对 AWS 资源的最低访问权限，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 IAM 直接向最终用户授予这些权限，但是如果您想限制最终用户对 AWS 资源的访问权限，则应将该策略附加到启动角色。然后，您可以使用 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 启动限制](#)。

Note

如果您向用户授予 AWS Service Catalog 管理员的 IAM 权限，则会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，您必须向他们提供其他 IAM 权限，以允许他们使用产品 CloudFormation 模板中的每个底层 AWS 资源。例如，如果产品模板包含 Amazon Relational Database Service (Amazon RDS)，则您必须向用户授予用于启动产品的 Amazon 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 IAM 中直接向最终用户授予这些权限，但是如果您想限制最终用户对 AWS 资源的访问权限，则应将该策略附加到启动角色。然后，您可以使用 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 启动限制](#)。

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 JSON 中使用 Condition 块作为 `accountLevel`、`roleLevel` 或 `userLevel` 来指定上述权限。

这些示例也适用于 AWS Service Catalog API 写入操作的访问级别：UpdateProvisionedProduct 和 TerminateProvisionedProduct，以及读取操作：DescribeRecordScanProvisionedProducts、和 ListRecordHistory。ScanProvisionedProducts 和 ListRecordHistory API 操作使用 AccessLevelFilterKey 作为输入，该键的值对应于此处讨论的 Condition 块级别（`accountLevel` 等同于“Account”的 AccessLevelFilterKey 值、“Role”的 `roleLevel` 值和“User”的 `userLevel` 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](#)
- [End-user 访问预配置的产品](#)
- [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。

JSON

```
{
  "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：

JSON

```

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

```

不在任何策略中 AWS Service Catalog 为指定Condition区块被视为指定"servicecatalog:accountLevel"访问权限。注意，accountLevel 访问权限包含 roleLevel 和 userLevel 访问权限。

End-user 访问预配置的产品

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。

JSON

```

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

```

```

    {
      "Effect": "Allow",
      "Action": [
        "servicelog:DescribeProduct",
        "servicelog:DescribeProductView",
        "servicelog:DescribeProvisioningParameters",
        "servicelog:DescribeRecord",
        "servicelog:ListLaunchPaths",
        "servicelog:ListRecordHistory",
        "servicelog:ProvisionProduct",
        "servicelog:ScanProvisionedProducts",
        "servicelog:SearchProducts",
        "servicelog:TerminateProvisionedProduct",
        "servicelog:UpdateProvisionedProduct"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "servicelog: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。

JSON

```

{
  "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"
      }
    }
  }
]
}

```

JSON

```

{
  "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**

您可以附加AWSServiceCatalogAdminFullAccess到您的 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。

此策略授予允许完全访问管理员控制台视图的 *administrative* 权限，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**— 允许列出、读取、写入和标记 AWS CloudFormation 堆栈的 AWS Service Catalog 完全权限。
- **config**— 允许通过 AWS Config 访问产品组合、产品和预配置产品的 AWS Service Catalog 有限权限。
- **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**— AWS Service Catalog AWS Systems Manager 允许使用列出和阅读当前 AWS 账户和 AWS 区域中的 Systems Manager 文档。

查看策略：[AWSServiceCatalogAdminFullAccess](#)。

AWS 托管策略：**AWSServiceCatalogAdminReadOnlyAccess**

您可以附加AWSServiceCatalogAdminReadOnlyAccess到您的 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。

此策略授予允许完全访问管理员控制台视图的 *read-only* 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`— 允许列出和读取 AWS CloudFormation 堆栈的 AWS Service Catalog 有限权限。
- `config`— 允许通过 AWS Config访问产品组合、产品和预配置产品的 AWS Service Catalog 有限权限。
- `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`— AWS Service Catalog AWS Systems Manager 允许使用列出和阅读当前 AWS 账户和 AWS 区域中的 Systems Manager 文档。

查看策略：[AWSServiceCatalogAdminReadOnlyAccess](#)。

AWS 托管策略：[AWSServiceCatalogEndUserFullAccess](#)

您可以附加[AWSServiceCatalogEndUserFullAccess](#)到您的 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。

此策略授予允许完全访问最终用户控制台视图的 *contributor* 权限，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`— 允许列出、读取、写入和标记 AWS CloudFormation 堆栈的 AWS Service Catalog 完全权限。
- `config`— 允许 AWS Service Catalog 有限权限通过 AWS Config列出和阅读有关产品组合、产品和预配置产品的详细信息。
- `ssm`— AWS Service Catalog 允许使用读 AWS Systems Manager 取当前 AWS 账户和 AWS 区域中的 Systems Manager 文档。

查看策略：[AWSServiceCatalogEndUserFullAccess](#)。

AWS 托管策略：[AWSServiceCatalogEndUserReadOnlyAccess](#)

您可以附加[AWSServiceCatalogEndUserReadOnlyAccess](#)到您的 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。

此策略授予的 *read-only* 权限允许对最终用户控制台视图进行只读访问。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`— 允许列出和读取 AWS CloudFormation 堆栈的 AWS Service Catalog 有限权限。
- `config`— 允许 AWS Service Catalog 有限权限通过 AWS Config 列出和阅读有关产品组合、产品和预配置产品的详细信息。
- `ssm`— AWS Service Catalog 允许使用读 AWS Systems Manager 取当前 AWS 账户和 AWS 区域中的 Systems Manager 文档。

查看策略：[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对 AWS Service Catalog 操作（例如，API 调用）和其他 AWS Service Catalog 依赖的 AWS 服务操作进行有限的访问。

权限详细信息

该策略包含以下权限。

- `servicecatalog`— 允许 AWS Service Catalog 工件同步角色有限地访问 AWS Service Catalog 公共 API。
- `codeconnections`— 允许 AWS Service Catalog 工件同步角色有限地访问 CodeConnections 公共 API。
- `cloudformation`— 允许 AWS Service Catalog 工件同步角色有限地访问 AWS CloudFormation 公共 API。

查看策略：[AWSServiceCatalogSyncServiceRolePolicy](#)。

Service-linked 角色详情

AWS Service Catalog 将上述权限详细信息用于在用户创建或更新使用的 AWS Service Catalog CodeConnections 产品时创建的 AWSServiceRoleForServiceCatalogSync 服务相关角色。您可以使用 AWS CLI、AWS API 或通过 AWS Service Catalog 控制台修改此策略。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 [为 AWS Service Catalog 使用服务相关角色 \(SLR\)](#)。

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

- `servicecatalog:ListProvisioningArtifacts`— 允许 AWS Service Catalog 工件同步角色列出同步到存储库中模板文件的给定 AWS Service Catalog 产品的配置工件。
- `servicecatalog:DescribeProductAsAdmin`— 允许 AWS Service Catalog 工件同步角色使用 `DescribeProductAsAdmin` API 获取 AWS Service Catalog 产品及其关联的预配置工件的详细信息，这些工件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工件的服务限额限制。
- `servicecatalog>DeleteProvisioningArtifact`— 允许 AWS Service Catalog 工件同步角色删除已置备的对象。
- `servicecatalog:ListServiceActionsForProvisioningArtifact`— 允许 AWS Service Catalog 对象同步角色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 工件同步角色从 `DescribeProvisioningArtifact` API 中检索详细信息，包括 `SourceRevisionInfo` 输出中提供的提交 ID。
- `servicecatalog>CreateProvisioningArtifact`— 如果检测到外部存储库中的源模板文件发生了更改（例如，已提交 `git-push`），则允许 AWS Service Catalog 工件同步角色创建新的预配置工件。
- `servicecatalog:UpdateProvisioningArtifact`— 允许 AWS Service Catalog 工件同步角色更新已连接或已同步产品的已配置对象。
- `codeconnections:UseConnection`— 允许 AWS Service Catalog 工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 对象同步角色的有限访问权限 AWS CloudFormation 来验证外部存储库中使用的模板的模板格式，并验证是否 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)，AWS Service Catalog 允许与同步。AWS Organizations

此策略授予的权限允许对 AWS Service Catalog 操作（例如，API 调用）和其他 AWS Service Catalog 依赖的 AWS 服务操作进行有限的访问。

权限详细信息

该策略包含以下权限。

- `organizations`— 允许 AWS Service Catalog 数据同步角色有限地访问 AWS Organizations 公共 API。

查看策略：[AWSServiceCatalogOrgsDataSyncServiceRolePolicy](#)。

Service-linked 角色详情

AWS Service Catalog 将上述权限详细信息用于在用户启用 AWS Organizations 共享产品组合访问权限或创建投资组合共享时创建的 `AWSServiceRoleForServiceCatalogOrgsDataSync` 服务相关角色。您可以使用 AWS CLI、AWS API 或通过 AWS Service Catalog 控制台修改此策略。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[为 AWS Service Catalog 使用服务相关角色 \(SLR\)](#)。

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

- `organizations:DescribeAccount`— 允许 Org AWS Service Catalog anizations Data Sync 角色检索有关指定账户的 AWS Organizations 相关信息。
- `organizations:DescribeOrganization`— 允许 Organizations Data Sync 角色检索有关用户帐户所属组织的信息。AWS Service Catalog
- `organizations:ListAccounts`— 允许 Organizations Data Sync 角色列出用户组织中的帐户。AWS Service Catalog
- `organizations:ListChildren`— 允许 Or AWS Service Catalog ganizations Data Sync 角色列出包含在指定父组织单位或根目录中的所有组织单位 (UO) 或帐户。
- `organizations:ListParents`— 允许 Organizations Data Sync 角色列出作为指定子 AWS Service Catalog 组织单位或账户直系父级的一个或多个根组织单位。

- `organizations:ListAWSServiceAccessForOrganization`— 允许 AWS Service Catalog Organizations Data Sync 角色检索用户允许与其组织集成的 AWS 服务列表。

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策略，请参阅《AWS Identity and Access Management 用户指南》中的 [添加和删除 IAM 身份权限](#)。

AppRegistry 更新到 AWS 托管策略

查看 AppRegistry 自该服务开始跟踪这些更改以来 AWS 托管策略更新的详细信息。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 RSS feed。

更改	描述	日期
AWSServiceCatalogSyncServiceRolePolicy — 更新托管策略	AWS Service Catalog 已将 <code>AWSServiceCatalogSyncServiceRolePolicy</code> 政策更新为 <code>codeconnections</code> 为 <code>codeconnections</code> 。	2024 年 5 月 7 日
AWSServiceCatalogAdminFullAccess — 更新托管策略	AWS Service Catalog 更新了 <code>AWSServiceCatalogAdminFullAccess</code> 策略，增加了 AWS Service Catalog 管理员在其账户中创建 <code>AWSServiceRoleForS</code>	2023 年 4 月 14 日

更改	描述	日期
AWSServiceCatalogOrgsDataSyncServiceRolePolicy : 新托管策略	<p>serviceCatalogOrgsDataSync 服务相关角色 (SLR) 所需的权限。</p> <p>AWS Service Catalog 添加了AWSServiceCatalogOrgsDataSyncServiceRolePolicy 附加到AWSServiceRoleForServiceCatalogOrgsDataSync 服务相关角色 (SLR) 的，AWS Service Catalog 允许与同步。AWS Organizations此策略允许对所 AWS Service Catalog 依赖的 AWS Service Catalog 操作 (例如 API 调用) 和其他 AWS 服务操作进行有限的访问。</p>	2023 年 4 月 14 日
AWSServiceCatalogAdminFullAccess — 更新托管策略	<p>AWS Service Catalog 更新了AWSServiceCatalogAdminFullAccess 策略，使其包含 AWS Service Catalog 管理员的所有权限，并创建了与的兼容性 AppRegistry。</p>	2023 年 1 月 12 日

更改	描述	日期
AWSServiceCatalogSyncServiceRolePolicy : 新托管策略	AWS Service Catalog 添加了附加到AWSServiceRoleForServiceCatalogSync 服务相关角色 (SLR) 的AWSServiceCatalogSyncServiceRolePolicy 策略。此策略 AWS Service Catalog 允许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)。此角色是使用、创建、更新 CodeConnections 和描述产品的 AWS Service Catalog 预配对象所必需的。AWS Service Catalog	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，使其包含 AWS Service Catalog 管理员所需的所有权限。该策略确定了管理员可以对所有 AWS Service Catalog 资源采取的具体操作，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基于属性的访问控制 (ABAC) 功能。AWS Service Catalog ABAC 允许您将 AWSServiceCatalogAdminFullAccess 策略用作模板，根据标签允许或拒绝对 AWS Service Catalog 资源执行操作。有关更多信息，请参阅 AWS Identity and Access Management 中的 什么是适用于 AWS 的 ABAC 。	2022 年 9 月 30 日
AppRegistry 已开始跟踪更改	AppRegistry 开始跟踪其 AWS 托管策略的更改。	2022 年 9 月 15 日

将服务相关角色用于 AWS Service Catalog

AWS Service Catalog 使用 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 IAM 角色的独特类型。AWS Service Catalog Service-linked 角色由预定义 AWS Service Catalog ，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色使设置变得 AWS Service Catalog 更加容易，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 AWS Service Catalog 只能担任其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其他 IAM 实体的权限策略。

只有在首先删除相关资源后，您才能删除服务关联角色。这样可以保护您的 AWS Service Catalog 资源，因为您不会无意中删除访问资源的权限。

有关支持服务相关角色的其他服务的信息，请参阅与 [IAM 配合使用的AWS 服务](#)，并在Service-linked 角色列中查找标有“是”的服务。请选择是与查看该服务的[服务关联角色文档](#)的链接。

Service-linked 的角色权限 **AWSServiceRoleForServiceCatalogSync**

AWS Service Catalog 可以使用名为的服务相关角色

AWSServiceRoleForServiceCatalogSync— 必须使用此服务相关角色 AWS Service Catalog 才能使用 CodeConnections 、创建、更新和描述产品的 AWS Service Catalog 置备工件。

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

- `sync.servicecatalog.amazonaws.com`

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

- 操作：CodeConnections 上的 Connection
- 操作：Create, Update, and DescribeProvisioningArtifact为 AWS Service Catalog 产品开启

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。有关更多信息，请参阅 IAM 用户指南中的[Service-linked 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、或 AWS API CodeConnections 中建立服务相关角色时 AWS 管理控制台，会自动为您创建服务相关角色。AWS CLI

Important

如果您在其他使用此角色支持的的功能的服务中完成某个操作，此服务关联角色可以出现在您的账户中。另外，如果您在 2022 年 11 月 18 日 AWS Service Catalog 服务开始支持服务相关角色之前使用该服务，则在您的账户中 AWS Service Catalog 创建了该AWSServiceRoleForServiceCatalogSync角色。要了解更多信息，请参阅[我的 IAM 账户中出现新角色](#)。

如果您删除该服务关联角色，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。当您建立服务时 CodeConnections，AWS Service Catalog 会再次为您创建服务相关角色。

您还可以使用 IAM 控制台创建具有同步 AWS Service Catalog 产品用例的服务相关角色。在 AWS CLI 或 AWS API 中，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。

Service-linked 的角色权限 **AWSServiceRoleForServiceCatalogOrgsDataSync**

AWS Service Catalog 可以使用名为的服务相关角色

AWSServiceRoleForServiceCatalogOrgsDataSync— AWS Service Catalog 组织需要此服务相关角色才能与之保持同步。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：

JSON

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

名为的角色权限策略AWSServiceCatalogOrgsDataSyncServiceRolePolicy AWS Service Catalog 允许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

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。有关更多信息，请参阅 IAM 用户指南中的[Service-linked 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 将您的行为视为允许[与共享 AWS Organizations](#)或[共享产品组合](#)允许您代表您在后台创建 SLR。AWS Service Catalog

AWS Service Catalog 在您请求时或在 AWS 管理控制台、EnableAWSOrganizationsAccess或 AWS API CreatePortfolioShare 中自动为您创建服务相关角色。AWS CLI

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，此服务关联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 账户中出现新角色](#)。

如果您删除该服务关联角色，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。当您请求 EnableAWSOrganizationsAccess 或 CreatePortfolioShare 时，AWS Service Catalog 将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 AWSServiceRoleForServiceCatalogOrgsDataSync服务相关的角色。创建服务关联角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务关联角色](#)。

删除的服务相关角色 AWS Service Catalog

您可以使用 IAM 控制台、AWS CLI 或 AWS API 手动删除 `AWSServiceRoleForServiceCatalogSync` 或

`AWSServiceRoleForServiceCatalogOrgsDataSync` SLR。为此，必须先手动删除所有使用服务相关角色的资源（例如，任何同步到外部存储库的 AWS Service Catalog 产品），然后才能手动删除服务相关角色。

支持的区域 AWS Service Catalog 服务关联角色

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 区域和端点](#)。

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（北加利福尼亚）	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
亚太地区（大阪）	ap-northeast-3	是
亚太地区（首尔）	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
亚太地区（东京）	ap-northeast-1	是

区域名称	区域标识	Support in AWS Service Catalog
加拿大 (中部)	ca-central-1	是
欧洲地区 (法兰克福)	eu-central-1	是
欧洲地区 (爱尔兰)	eu-west-1	是
欧洲地区 (伦敦)	eu-west-2	是
欧洲地区 (米兰)	eu-south-1	是
欧洲地区 (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 (US-East)	us-gov-east-1	否
AWS GovCloud (US-West)	us-gov-west-1	否

问题排查 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 AWS Service Catalog 和 IAM 时可能遇到的常见问题。

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 管理控制台 告诉您您无权执行某项操作，则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有关虚构 my-example-widget 资源的详细信息，但不拥有虚构 aws:GetWidget 权限时，会发生以下示例错误。

```
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。请求该人员更新您的策略，以便允许您将角色传递给 AWS Service Catalog。

某些 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服务相关角色。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 AWS Service Catalog 中执行操作时，会发生以下示例错误。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，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACL) 的服务，您可以使用这些策略向人员授予对您的资源的访问权。

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 Service Catalog 管理员指南》AWS Identity and Access Management [AWS Service Catalog](#) 中的。
- 要了解如何通过您拥有的 AWS 账户提供对资源的访问权限，请参阅 [IAM 用户指南中的向您拥有的另一个 AWS 账户中的 IAM 用户提供访问权限](#)。
- 要了解如何向第三方 AWS 账户提供对您的资源的访问权限，请参阅 [IAM 用户指南中的向第三方 AWS 账户提供访问权限](#)。
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的 [为经过外部身份验证的用户 \(联合身份验证\) 提供访问权限](#)。

- 要了解使用角色和基于资源的策略进行跨账户存取之间的差别，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。

控制访问权限

AWS Service Catalog 产品组合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，启动约束将与最终用户从该产品组合中启动产品时使用的角色 ARN 关联。使用此模式，您可以控制对 AWS 资源创建的访问权限。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 与一项服务集成 AWS CloudTrail，该服务可捕获所有 AWS Service Catalog API 调用并将日志文件传送到您指定的 Amazon S3 存储桶。有关更多信息，请参阅使用 [记录 AWS Service Catalog API 调用 CloudTrail](#)。

您还可以使用通知约束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”](#)。

的合规性验证 AWS Service Catalog

Third-party AWS Service Catalog 作为多个合规计划的一部分，审计师评估的安全性和 AWS 合规性，包括：

- 系统和组织控制 (SOC)
- 支付卡行业数据安全标准 (PCI DSS)
- 联邦风险与授权管理项目 (FedRAMP)
- 健康保险流通与责任法案 (HIPAA)

有关特定合规计划范围内的 AWS 服务列表，请参阅[按合规计划划分的范围内的 AWS 服务](#)。有关一般信息，请参阅[AWS 合规计划AWS](#)。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅[下载 AWS Artifact 中的报告](#)。

您在使用时的合规责任 AWS Service Catalog 取决于数据的敏感性、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来创建应用程序。AWS HIPAA-compliant
- [AWS 合规性资源](#) – 此业务手册和指南集合可能适用于您的行业和位置。
- [AWS Config](#)— 该 AWS 服务评估您的资源配置在多大程度上符合内部实践、行业指导方针和法规。
- [AWS Security Hub CSPM](#)— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。

韧性在 AWS Service Catalog

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。AWS 区域提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。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失效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 区域和可用区的更多信息，请参阅[AWS 全球基础设施](#)。

除了 AWS 全球基础架构外，还 AWS Service Catalog 提供 AWS Service Catalog 自助服务。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。利用自助服务操作，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，例如，备份和还原、排查问题、运行批准的命令或请求权限。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 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅[AWS 云安全](#)。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的[基础设施保护](#)。

您可以使用 AWS 已发布的 API 调用 AWS Service Catalog 通过网络进行访问。客户端必须支持以下内容：

- 传输层安全性协议 (TLS)。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE (短暂的) 或 ECDHE (椭圆曲线短暂的 Diffie-Hellman)。Diffie-Hellman 大多数现代系统 (如 Java 7 及更高版本) 都支持这些模式。

使用 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 CopyProduct API 将产品复制到其他区域。

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。这些值称为模板约束，因为它们约束部署产品的 CloudFormation 模板的方式。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，请考虑下面一种情况：产品允许启动所有 Amazon EC2 实例且组合具有两项约束：一项允许启动所有非 GPU 类型的 EC2 实例，另一项仅允许启动 t1.micro 和 m1.small EC2 实例。在本示例中，AWS Service Catalog 应用第二个限制性更强的约束 (t1.micro 和 m1.small)。

当您 IAM 策略附加到启动角色时，您可以限制最终用户对 AWS 资源的访问权限。然后，您可以使用 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 托管策略 AWS Service Catalog](#)。

管理目录

AWS Service Catalog 提供了通过管理员控制台管理产品组合、产品和限制的界面。

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](#)
- [使用 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. 打开 Service Catalog 控制台，网址为<https://console.aws.amazon.com/servicecatalog/>。
2. 根据需要选择不同的区域。
3. 如果您是新手 AWS Service Catalog，则会看到起 AWS Service Catalog 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。

使用时 AWS Service Catalog，您可以随时返回“投资组合”页面；在导航栏中选择 Service Catalog，然后选择 Portfolios。

查看产品组合详细信息

在 AWS Service Catalog 管理员控制台中，投资组合详细信息页面列出了投资组合的设置。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. 打开 Service Catalog 控制台，网址为<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 创建投资组合并显示投资组合的详细信息。

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 CloudFormation 模板或 Terraform 配置文件。该 CloudFormation 模板存储在亚马逊简单存储服务 (Amazon S3) 存储桶中，存储桶名称以 “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。通过此页面创建产品时，将其 AWS Service Catalog 添加到当前选定的产品组合中。

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 CloudFormation 产品，请选择您自己的模板文件、本地驱动器中的 CloudFormation 模板或指向 Amazon S3 中存储的模板的 URL、现有 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. 打开 Service Catalog 控制台，网址为<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 启动限制](#)。

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 通知限制](#)。

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。规则已添加到产品使用的 CloudFormation 模板中。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 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. 登录 AWS 管理控制台 并打开 AWS Service Catalog 管理员控制台，网址为 <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](#)。

除了访问产品组合外，用户还必须有权访问 AWS Service Catalog 最终用户控制台。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. 打开 Service Catalog 控制台，网址为<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 定向到投资组合详情页面。
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。

Account-to-account 共享

要完成这些步骤，您必须获取目标账户的 AWS 账户 ID。您可以在目标账户的“我的账户”页面上找到该 ID。AWS 管理控制台

与 AWS 账户共享投资组合

1. 打开 Service Catalog 控制台，网址为 <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 发送给目标账户的 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 管理员与您共享投资组合，请将该投资组合导入您的账户，以便您可以将其产品分发给最终用户。

如果投资组合是通过共享的，则无需导入该投资组合 AWS Organizations。

要导入产品组合，您必须从管理员处获取产品 ID。

要查看所有导入的投资组合，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。

与共享 AWS Organizations

您可以使用共享 AWS Service Catalog 投资组合 AWS Organizations。

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《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投资组合

1. 打开 AWS Service Catalog 控制台，网址为<https://console.aws.amazon.com/servicecatalog/>。
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、上级组织单位 (OU)、子组织单位或组织内的 AWS 帐户共享产品组合。

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 帐户”以查看组织中所有 AWS 帐户的列表。

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，网址为<https://console.aws.amazon.com/servicecatalog/>。
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 控制台中找到该节点，网址为<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。

委派的管理员可以像管理账号一样共享其组织中的 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

如果您在组织内转移账户，则与该账户共享 AWS Service Catalog 的产品组合可能会发生变化。

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，您需要消费者的 AWS 账户 ID。在控制台的 AWS “我的账户” 中找到账户 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。当您共享投资组合时，请 AWS Service Catalog 验证这些委托人名称是否已经存在。如果确实存在，则 AWS Service Catalog 自动将匹配的 IAM 委托人与共享产品组合关联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。对于收款人账户中不是 AWS Service Catalog 管理员但仍有能力创建委托人（用户/角色）的用户，该用户可以创建与投资组合的委托人名称关联相匹配的 IAM 委托人。尽管该用户可能不知道通过哪些主体名称相关联 AWS Service Catalog，但他们可能能够猜出该用户。如果担心这种潜在的升级路径，则 AWS Service Catalog 建议使用 a `PrincipalType s` 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委托人姓名时，AWS Service Catalog 会自动将这些更改应用到收款人账户中。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

主人姓名共享仅适用于 AWS Organizations。

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”以启动委托人共享。AWS Service Catalog 然后在收款人账户中共享主体姓名。

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 委托人（用户、群组或角色）名称授予投资组合访问权限。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

要向不在您的 AWS 账户用户中的用户（例如属于其他组织或组织 AWS 账户中其他组织的用户）提供您的 AWS Service Catalog 产品，您可以与他们共享您的产品组合。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，不能更改产品或限制，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](#)。

收件人管理员可以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。
- 通过将导入的投资组合中的产品添加到本地产品组合，即收款人管理员创建的属于其 AWS 账户的单独产品组合。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，但无法删除原本从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。

启动角色是一个 IAM 角色，AWS Service Catalog 用于在最终用户启动产品时预配置 AWS 资源（例如 Amazon EC2 实例或 Amazon RDS 数据库）。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 启动限制](#)。拥有启动角色的 AWS 账户会配置 AWS 资源，该账户将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跨账户共享投资组合 AWS Service Catalog。

在@@ [各个账户中共享 \(https://www.youtube.com/embed/BVSohYOppjk% 22% 3EShare\)](https://www.youtube.com/embed/BVSohYOppjk%22%3EShare) 投资组合。AWS Service Catalog

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	<p>已继承。</p> <p>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。</p> <p>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。</p>	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	<p>已继承。</p> <p>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。</p> <p>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。</p>	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

您可以在 AWS Service Catalog 管理员控制台的“产品”列表页面上管理产品。

要查看产品列表页面

1. 打开 Service Catalog 控制台，网址为<https://console.aws.amazon.com/servicecatalog/>。
2. 选择产品列表。

创建产品

您可以从 AWS Service Catalog 管理员控制台的“产品”页面创建产品。

Note

创建 Terraform 产品需要额外配置，包括 Terraform 预配置引擎和启动角色。有关更多信息，请查看 [开始使用 Terraform 产品](#)。

创建新 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. 产品详情-允许您选择要创建的产品类型。AWS Service Catalog 支持 CloudFormation、Terraform Cloud 和 External (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-从本地驱动器上传 CloudFormation 模板或 Terraform tar.gz 配置文件。
 - 使用 Amazon S3 URL - 指定 URL，该 URL 指向 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 CloudFormation 或 Terraform 模板中定义参数属性，并在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、允许值以及该值的正则表达式。AWS Service Catalog 如果提供的值不符合参数属性，则在置备期间发出警告。要了解有关参数属性的更多信息，请参阅《CloudFormation 用户指南》中的[参数](#)。

问题排查

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。
- 以 “cf-templates-*” 开头的存储桶，用于 AWS Service Catalog 存储配置项目模板。
- 以 “sc-*” 开头的内部存储桶，用于 AWS Service Catalog 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许多混合使用 Terraform 产品类型的 CloudFormation 产品。

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（CloudFormation 或 Terraform）。例如，如果您更新 CloudFormation 产品，则无法用 Terraform tar.gz 配置文件替换现有 CloudFormation 模板。必须使用新的 CloudFormation 模板文件更新现有 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 建议您在 Terraform 引擎中 CloudFormation 或在 Terraform 引擎中测试产品更新，以确保它们能正常运行。

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-从本地驱动器上传 CloudFormation 模板或 Terraform tar.gz 配置文件。

使用 Amazon S3 URL - 指定 URL，该 URL 指向 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 Git-synced 商品称为产品。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-synced 产品连接](#)
- [更新 Git-synced 产品连接](#)
- [删除 Git-synced 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 对 Git-synced 产品的支持](#)

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](#)。

JSON

```
{
  "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，您必须创建并授权一次性的账户到账户连接。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[的 Service-linked 角色。AWS Service Catalog](#)

创建新 Git-synced 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、Enterprise GitHub 或 Bitbucket 的模板文件](#)。

- a. （创建新 AWS Service Catalog 产品资源时可选）在 Support Details 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-synced 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，然后根据需要进行添加产品。
5. 添加所有所需产品后，选择批量创建产品。

连接现有的 AWS Service Catalog 产品到外部存储库

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-synced 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 管理控制台

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-synced 产品连接

您可以使用 AWS Service Catalog 控制台、AWS Service Catalog API 或，更新现有的账户关联和 Git-synced 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 Git-synced 产品连接](#)。

将现有产品更新为 Git-synced 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。

要更新连接和存储库的详细信息，请使用 AWS Service Catalog 控制台

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-synced 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物。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](#)。

要断开 Git-synced 产品的连接，请使用 AWS Service Catalog 控制台

1. 在中 AWS 管理控制台，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-synced 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、Enterprise GitHub 或 Bitbucket 的模板文件

使用 Terraform 配置文件创建 Git-synced 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 区域对 Git-synced 产品的支持

AWS Service Catalog 支持中的 Git-synced 产品，AWS 区域如下表所示。

AWS 区域 名字	AWS 区域 身份	Support 对 Git-synced 产品的支持
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 (北加利福尼亚)	us-west-1	是
美国西部 (俄勒冈州)	us-west-2	是
非洲 (开普敦)	af-south-1	否
亚太地区 (香港)	ap-east-1	否
亚太地区 (雅加达)	ap-southeast-3	否
亚太地区 (新西兰)	ap-southeast-6	否
亚太地区 (孟买)	ap-south-1	是
亚太地区 (大阪)	ap-northeast-3	否
亚太地区 (首尔)	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
亚太地区 (东京)	ap-northeast-1	是
加拿大 (中部)	ca-central-1	是
加拿大西部 (卡尔加里)	ca-west-1	否
欧洲地区 (法兰克福)	eu-central-1	是
欧洲地区 (爱尔兰)	eu-west-1	是
欧洲地区 (伦敦)	eu-west-2	是
欧洲地区 (米兰)	eu-south-1	否

AWS 区域 名字	AWS 区域 身份	Support 对 Git-synced 产品的支持
欧洲地区 (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	否
南美洲 (圣保罗)	sa-east-1	是
AWS GovCloud (US-East)	us-gov-east-1	否
AWS GovCloud (US-West)	us-gov-west-1	否

删除产品

删除产品时，AWS Service Catalog 会从包含该产品的每个产品组合中删除所有产品版本。

AWS Service Catalog 允许您使用 AWS Service Catalog 控制台删除产品或 AWS CLI。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。

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”表格，该表显示了该产品的所有关联资源。AWS Service Catalog 当您删除产品时，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。

AWS Service Catalog 如果无法解除所有产品资源的关联，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

- [使用删除商品 AWS CLI](#)
- [解决删除产品时资源取消关联失败的问题](#)

使用删除商品 AWS CLI

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 是一款开源工具，可让您使用命令行外壳中的命令与 AWS 服务进行交互。AWS Service Catalog 强制删除功能需要 [AWS CLI 别名](#)，这是您可以在中创建的快捷方式，AWS CLI 以缩短经常使用的命令或脚本。

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](#)。请使用最低 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](#)。
- 创建 AWS CLI 别名以批处理 Disassociation API 调用，使您能够在单个命令中删除产品。

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该 `DeleteProduct` 调用的资源。AWS CLI

在您的 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 配置文件夹并打开该 `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` 别名添加到您的 `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 使用别名和删除 AWS Service Catalog 产品的更多信息，请查看以下资源：

- 在 AWS Command Line Interface (CLI) 用户指南中@@@ [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](#)。
- [AWS re: 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。

版本包含 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。
- 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 启动限制](#)
- [AWS Service Catalog 通知限制](#)
- [AWS Service Catalog 标签更新限制](#)

- [AWS Service Catalog 堆栈集约束](#)
- [AWS Service Catalog 模板约束](#)

AWS Service Catalog 启动限制

启动约束指定最终用户启动、更新或终止产品时所扮演的 AWS Identity and Access Management (IAM) 角色。AWS Service Catalog IAM 角色是用户或 AWS 服务可以临时承担的使用 AWS 服务的权限集合。有关介绍性示例，请参阅：

- 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，他们必须拥有产品使用的 AWS 服务的权限，以及 AWS Service Catalog。CloudFormation 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 CloudFormation 托管策略
- 产品 AWS CloudFormation 模板中的服务

- 在服务拥有的 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 预配置引擎在执行配置操作时担任)
- `resource-groups:CreateGroup`用于资源组标记 (假设 AWS Service Catalog 用于创建资源组和分配标签)

IAM 角色的信任策略必须 AWS Service Catalog 允许代入该角色。在以下步骤中，当您选择角色类型时 AWS Service Catalog，将自动设置信任策略。如果您不使用控制台，请参阅[如何在 IAM 角色中使用信任策略中的为代入角色的 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 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. 使用 <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。此操作指示 AWS Service Catalog 在最终用户启动产品时代入该角色。

将角色分配给产品

1. 打开 Service Catalog 控制台，网址为 <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 支持使用代入角色请求 APIs 的 [混乱副手](#) 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 APIs 是为了保护混乱的副手：

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

AWS Service Catalog 仅对 sourceArn 保护支持模板化 ARNs，例如 “arn:<aws-partition>:servicecatalog:<region>:<accountId>:” 它不支持特定的资源 ARNs。

验证启动约束

要验证 AWS Service Catalog 使用角色启动产品并成功配置产品，请从 AWS Service Catalog 控制台启动产品。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 通知限制

Note

AWS Service Catalog 不支持 Terraform 开源或 Terraform Cloud 产品的通知限制。

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. 在 [v3/home](https://console.aws.amazon.com/sns/) 上打开亚马逊 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. 打开 Service Catalog 控制台，网址为 <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 标签更新限制

Note

AWS Service Catalog 不支持 Terraform 开源产品的标签更新限制。

通过标签更新限制，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. 打开 Service Catalog 控制台，网址为<https://console.aws.amazon.com/servicecatalog/>。
2. 选择包含要更新的产品产品组合。
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 Terraform 开源产品的堆栈集约束。
- AutoTags 目前不支持 CloudFormation StackSets。

堆栈集约束允许您使用配置产品部署选项 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. 打开 Service Catalog 控制台，网址为<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 Cloud 产品的模板限制。

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。您可以将模板约束应用于产品 AWS Service Catalog 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。

模板约束由一个或多个规则组成，这些规则缩小了在产品基础 CloudFormation 模板中定义的参数的允许值。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，参数可以定义用户在启动包含实例的堆栈时可以选择的各种 EC2 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 实例类型对于仅应使用小实例类型（例如 `t2.micro` 或 `t2.small`）的用户来说太大，则可以添加模板约束来限制最终用户可以选择的实例类型。有关 CloudFormation 模板参数的更多信息，请参阅《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。

您可以使用规则编辑器或在 AWS Service Catalog 管理员控制台中将规则写成 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. 打开 Service Catalog 控制台，网址为<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何时可以使用该模板，以及他们可以为用于创建他们尝试使用的产品的 CloudFormation 模板中声明的参数指定哪些值。规则可用于防止最终用户无意中指定错误的值。例如，您可以添加一条规则来验证最终用户是否在给定 VPC 中指定了有效的子网，或者是否在测试环境中使用了 `m1.small` 实例类型。CloudFormation 在为产品创建资源之前，使用规则来验证参数值。

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，会在创建或更新堆栈之前 CloudFormation 评估断言以检查子网参数值是否在该 VPC 中。如果参数值无效，则 CloudFormation 立即无法创建或更新堆栈。如果用户未指定 VPC，则 CloudFormation 不检查子网参数值。

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` 的规则的部分 `Rule02`。 `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 Cloud 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，同时遵守合规性和安全措施。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。这些 [AWS Systems Manager 文档](#) 允许访问实施 AWS 最佳实践的预定义操作，例如 Amazon EC2 停止和重启，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

本教程假设您拥有完全的 AWS 管理员权限，您已经熟悉了 AWS Service Catalog，并且您已经拥有一组基本的产品、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：使用管理服务操作 AWS CloudFormation](#)
- [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。在此示例中，最终用户需要获得访问 AWS Service Catalog 服务操作功能和执行 Amazon EC2 重启的权限。

更新权限

1. 打开 AWS Identity and Access Management (IAM) 控制台，网址为<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。在此示例中，我们通过选择为群组的“置备”和“终止”权限创建的自定义策略，AWS Service Catalog 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：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1536341175150",
      "Action": [
        "servicelog:ListServiceActionsForProvisioningArtifact",
        "servicelog: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，选择一个 AWS Systems Manager 文档来定义服务操作。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-重启EC2实例”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-重新启动 EC2instance，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-重新启动 EC2 instance 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：使用管理服务操作 AWS CloudFormation

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

- [AWS::ServiceCatalog::CloudFormationProduct ProvisioningArtifactProperties](#)
- [AWS::ServiceCatalog::ServiceAction](#) 协会

Note

如果您管理服务操作与 CloudFormation 资源的关联，请不要通过或添加或删除服务操作 AWS 管理控制台。AWS Command Line Interface 当您执行堆栈更新时，在 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 无法担任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 SSM 文档中指定的目标类型相匹配的资源，例如:: EC2 AWS:: Instance。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 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 担任指定角色时遇到未知错误。

在中找不到参数的值 **InvalidValue** "" {ValidValue1}, {ValidValue2}

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)

传递给的参数 AWS Service Catalog 未在服务操作定义中定义。您只能使用在服务操作定义中定义参数。

步骤处于 executing/canceling 操作状态时失败。 ***Error message***. 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。

使用 CloudFormation StackSets

Note

AutoTags 目前不支持 CloudFormation StackSets。

您可以使用 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 CloudFormation StackSets](#)。

堆栈集和堆栈实例

堆栈集允许您使用单个 CloudFormation 模板在跨 AWS 区域的 AWS 账户中创建堆栈。

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

在中 AWS Service Catalog，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 预算来跟踪您的服务成本和使用情况 AWS Service Catalog。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。

AWS 预算使您能够设置自定义预算，以便在您的成本或使用量超过（或预计超过）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 账单与成本管理 控制台中激活成本分配标签。有关更多信息，请参阅《AWS 账单与成本管理 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。

您还需要允许将要使用预算功能的任何用户或群组访问 AWS 账单与成本管理 控制台。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. 使用 <https://console.aws.amazon.com/iam/> 打开 IAM 控制台。
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。

JSON

```
{
  "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-1:123456789012:*"
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

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. 打开 Service Catalog 控制台，网址为<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“在中创建”AWS Budgets。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

如果您从中删除预算 AWS Budgets，则与 AWS Service Catalog 产品和产品组合的现有关联仍然存在。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. 打开 Service Catalog 控制台，网址为 <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

如果您从预算中删除 AWS 预算，则与 AWS Service Catalog 产品和产品组合的现有关联仍然存在。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. 打开 Service Catalog 控制台，网址为 <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 提供了用于管理预配置产品的接口。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，网址为 <https://console.aws.amazon.com/servicecatalog/>。

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，您也可以在此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。

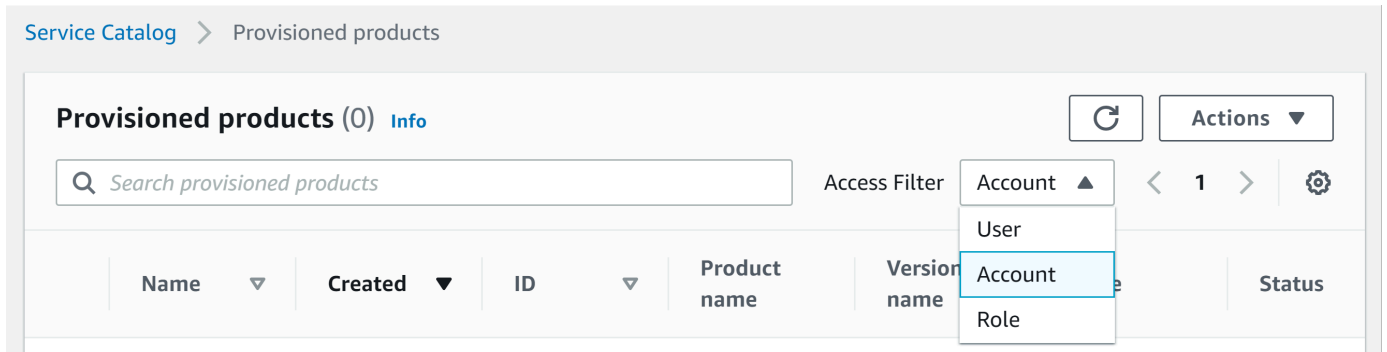
教程：确定用户资源分配

您可以使用 AWS Service Catalog 控制台识别配置了产品和与该产品关联的资源的用户。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。

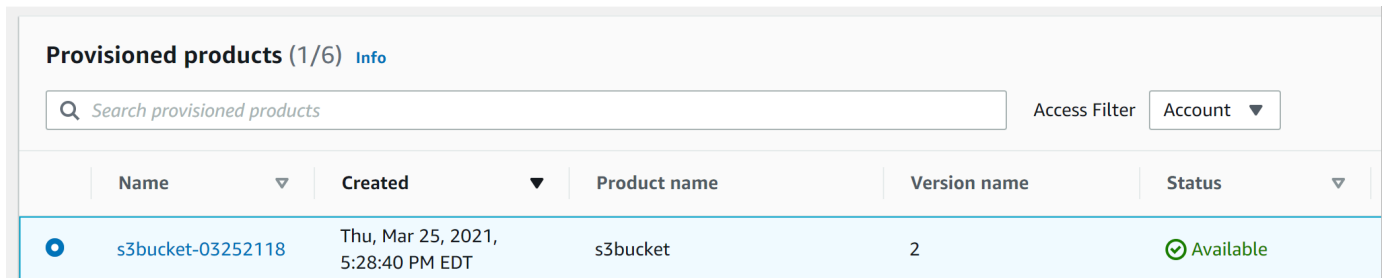
要管理账户的所有预配置产品，您需要具备 `AWSServiceCatalogAdminFullAccess` 或对预配置产品的同等写入访问权限。有关更多信息，请参阅《AWS Service Catalog 用户指南》中的[身份和访问权限管理](#)。

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。

Provisioned product details

Product description
-

Provisioned product ID pp-4asmmz2dkcows	User name SCAdminAllow	Status Available
Product name shsen-test	User ARN arn:aws:iam::776643078058:user/SCAdminAllow	Version name -
Created Thu, Jul 15, 2021, 9:49:54 AM PDT		

▼ More details

Product ID prod-y7bsnu3kn7eso	Type CFN_STACK	Support email contact -
Version ID pa-2d5inxhjryyrg4	Product owner 55440542	Support link -

Support description
-

5. 向下滚动到事件部分。请注意 Provisioned product ID 和 CloudformationStackARN 的值。

Events (4) [Info](#)

Search events

Sort by Newest

◀ 1 ▶ ⚙

▼ UPDATE_PROVISIONED_PRODUCT

Date created Thu, May 27, 2021, 5:06:38 PM EDT	CloudFormationStackARN Copy to clipboard	Status Succeeded
Record ID rec-444444444444	Product name ssmlimport	Product version 1
Provisioning artifact ID pa-444444444444		

Output key	Output value	Output description
CloudformationStackARN	arn:aws:cloudformation:us-east-1:776643078058:stack/SC-444444444444-11eb-b851-0a8a0480d74d	The ARN of the launched Cloudformation Stack

6. 使用预配置的产品 ID 来识别与本次发布相对应的 AWS CloudTrail 记录并识别请求用户（通常，您在联合身份验证期间输入电子邮件地址）。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-userstest/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. 使用该CloudFormationStackARN值来识别 CloudFormation 事件，以查找有关已创建资源的信息。您也可以使用 CloudFormation API 来获取此信息。有关更多信息，请参阅 [AWS CloudFormation API 参考](#)。

您可以使用 AWS Service Catalog API 或执行步骤 1 到 4 AWS CLI。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 仅在此期间应用标签ProvisionProduct。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：在期间 AWS Service Catalog 不创建资源组 ProvisionProduct

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 预配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，并将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 预配置引擎会做出表示“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：在此期间 AWS Service Catalog 创建新资源 UpdateProvisionedProduct

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 预配置引擎会做出表示“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 预配置引擎会做出表示“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 确保为所有设置为 from 的预配置产品创建资源组。TAINTED ProvisionProduct 如果 Terraform 配置引擎未返回 ResourceIdentifier，或者创建资源组 AWS Service Catalog 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 预配置引擎可以读取或写入文件内容。

在中管理标签 AWS Service Catalog

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，这些标签反映了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，可以在用于配置的下游服务（例如 Amazon 和 Amazon EC2 S3）的“标签”部分中查看。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 ARN。
- `aws:servicecatalog:provisioningArtifactIdentifier`-原始配置工件（产品版本）的 ID。

AWS Service Catalog TagOption 图书馆

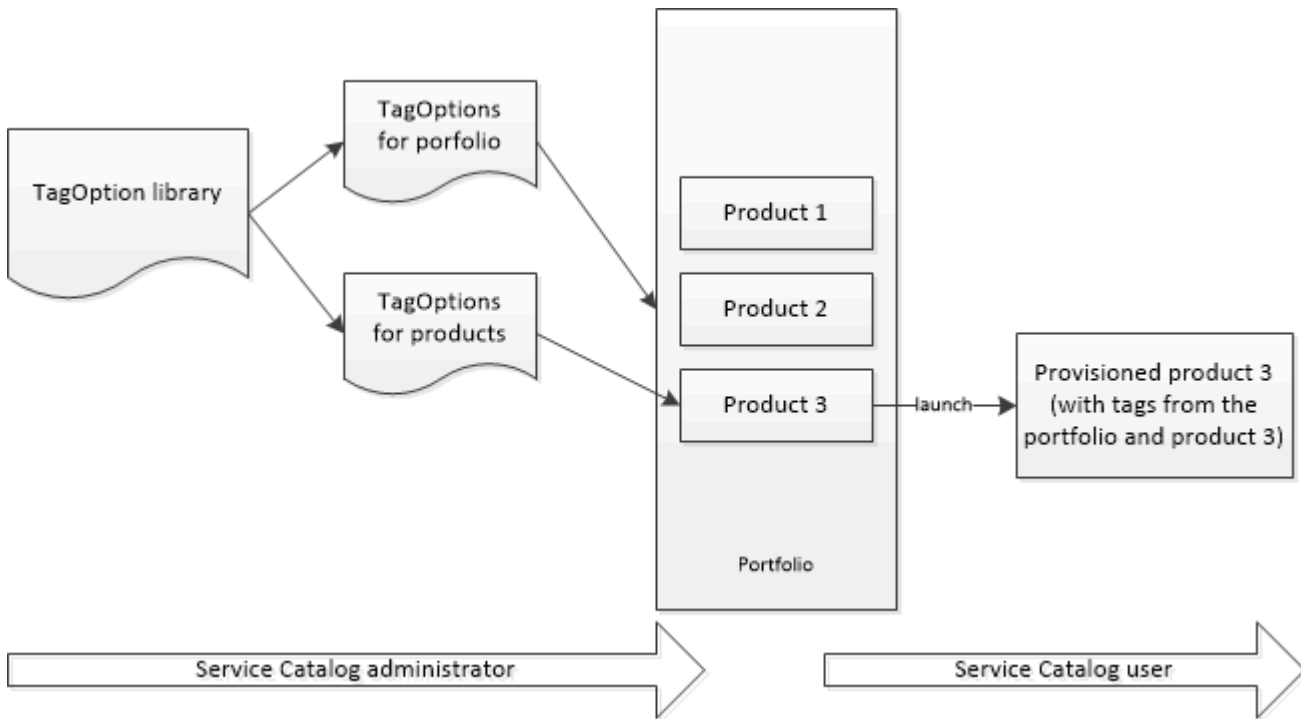
为了让管理员能够轻松管理已配置产品的标签，AWS Service Catalog 提供了一个 TagOption 库。A TagOption 是在中管理的键值对。AWS Service Catalog 它不是 AWS 标签，而是用作根据创建 AWS 标签的模板 TagOption。

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
- 正确标记资源 AWS Service Catalog
- 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

- [使用以下方式发布产品 TagOptions](#)
- [管理 TagOptions](#)
- [TagOptions 与 AWS Organizations 标签策略一起使用](#)

使用以下方式发布产品 TagOptions

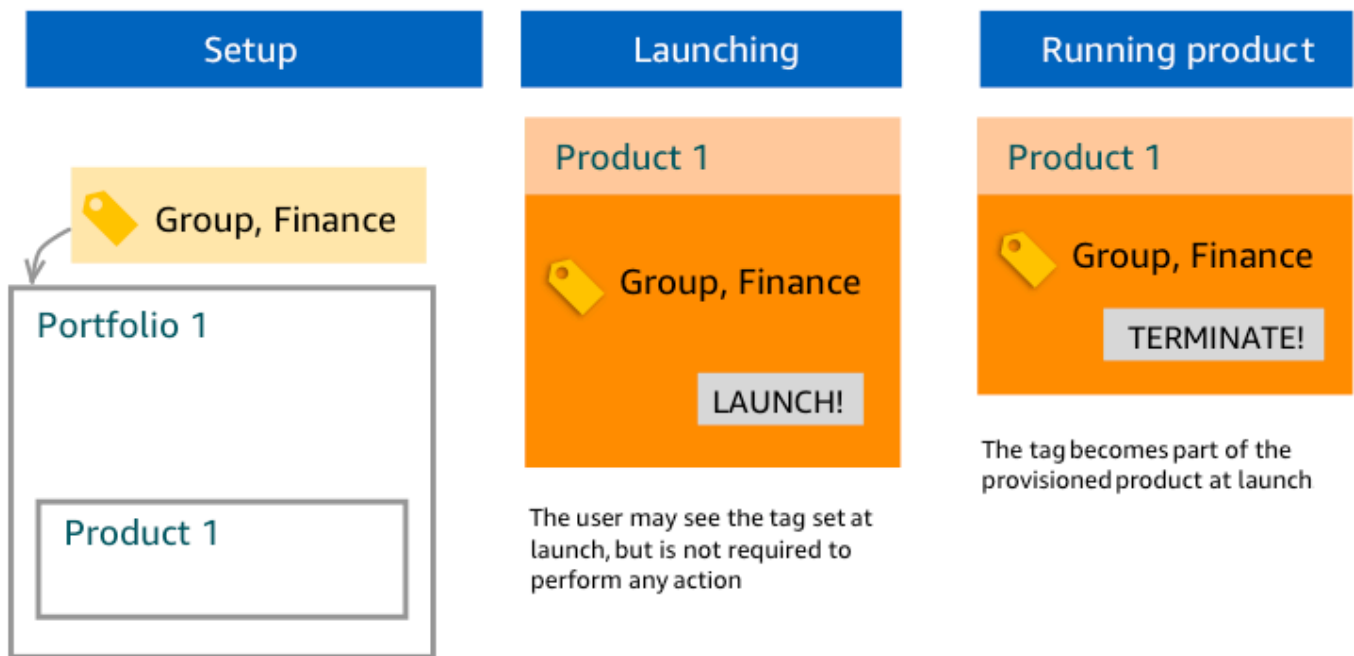
当用户启动的产品时 TagOptions，AWS Service Catalog 会代表您执行以下操作：

- 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。

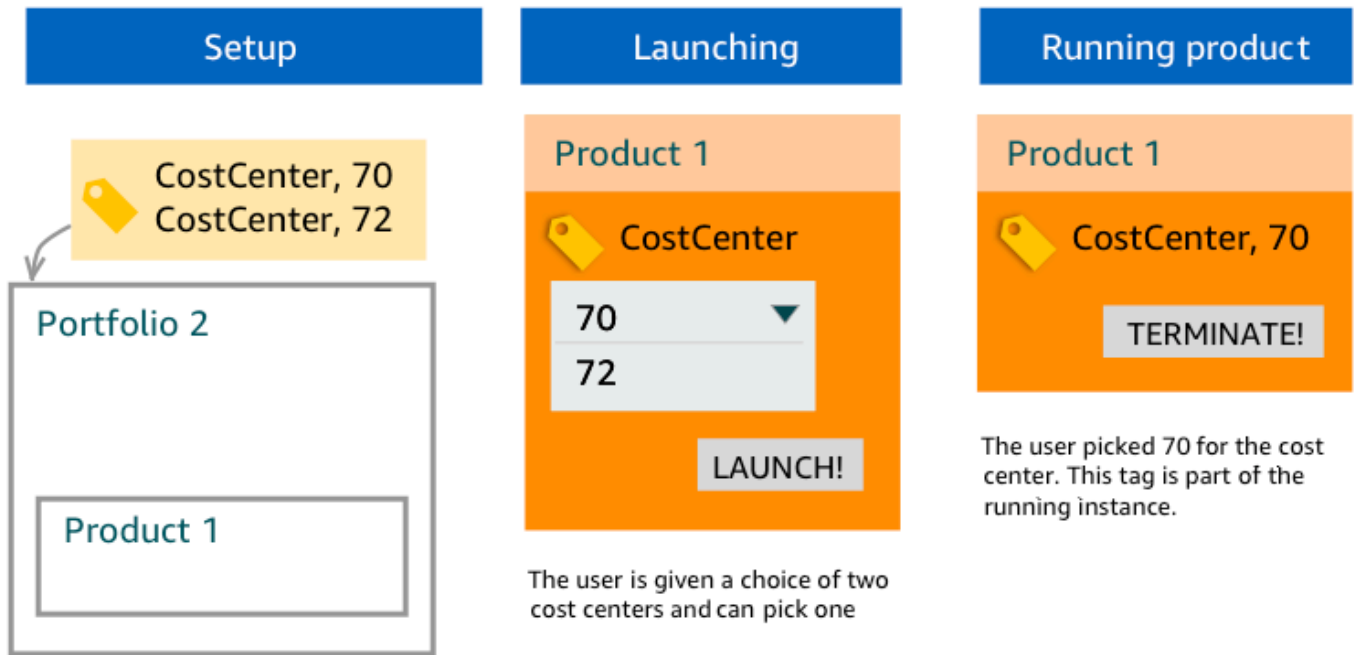
示例 1：唯一 TagOption 密钥

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：



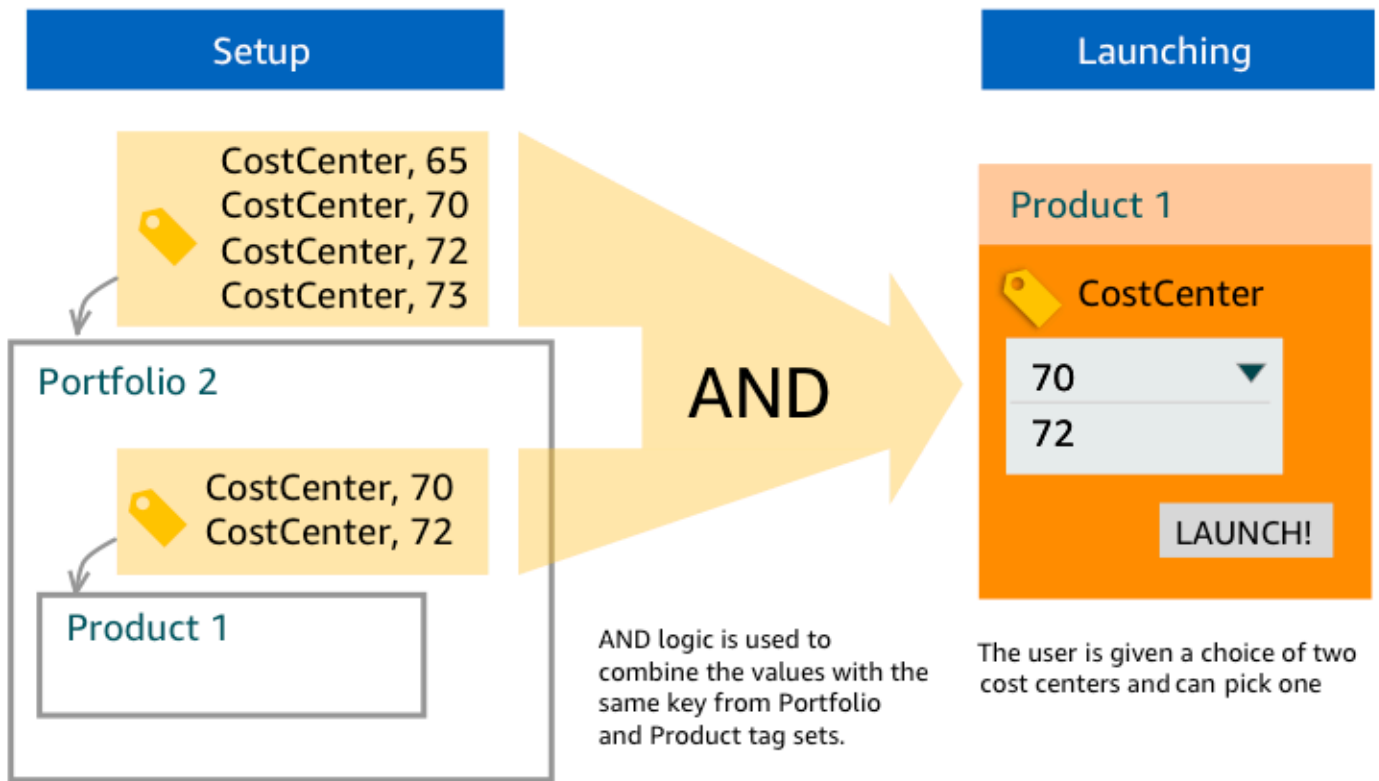
示例 2：投资组合上 TagOptions 具有相同密钥的集合

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。



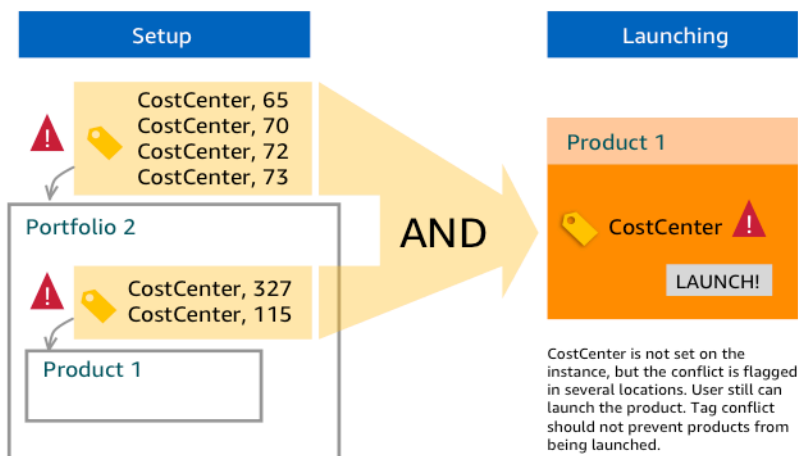
示例 3：一组在产品组合和该产品组合中的产品上都 TagOptions 具有相同密钥的

管理员在产品组合中放置了多个 TagOptions 具有相同密钥的同一个密钥，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。如果聚合找不到密钥的值，则 AWS Service Catalog 会创建一个具有相同键值的标签 `sc-tagconflict-portfolioid-productid`，其值为，产品组合 *portfolioid* 和产品的 *productid* 位置。ARNs 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. 打开 Service Catalog 控制台，网址为<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. 打开 Service Catalog 控制台，网址为<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. 打开 Service Catalog 控制台，网址为<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。

在控制台将一个或多个产品组合 TagOptions 与投资组关联或取消关联

1. 打开 Service Catalog 控制台，网址为<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，网址为：<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. 打开 Service Catalog 控制台，网址为<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更改不合规的标签，但在控制台中将其报告为不合规。AWS Organizations

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，我们建议您采取以下措施：

- 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。
- 与可能在中发布产品 AWS Service Catalog 并在产品发布中附加可选的最终用户标签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 **Atlanta** **San 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。	<p>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。</p> <p>添加了不符合您标签策略的可选自定义标签。</p>	<p>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。</p> <p>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）。</p>
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。	<p>正确配置标签策略。如果您未指定标签键大小写合规性，则默认的标志键大小写均为小写。</p> <p>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 标签密钥全部 AWS Service</p>

场景	Reason	解决方案
		<p>Catalog 为小写以遵守执法规则。</p> <p>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。</p>
由于标签值不兼容，产品无法启动。	为产品发布选择不在于 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 Catalog [DescribeProvisioningArtifact](#) 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 会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 [NotifyProvisionProductEngineWorkflowResult](#) API 来报告成功或失败。

对 [UpdateProvisionedProduct](#) 和的呼叫处理方式类似，每个呼叫 [TerminateProvisionedProduct](#) 都有不同的队列和 Notify APIs：

- [NotifyProvisionProductEngineWorkflowResult](#)
- [NotifyUpdateProvisionedProductEngineWorkflowResult](#)
- [NotifyTerminateProvisionedProductEngineWorkflowResult](#).

主题

- [注意事项](#)
- [参数解析](#)
- [预置](#)
- [Updating](#)
- [正在终止](#)

- [标签](#)

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、更新或终止产品时所扮演的 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	Type	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	Type	必填	描述
参数	列表	是	Service Catalog 要求最终用户在置备产品

字段	Type	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。
键	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。
类型	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。
描述	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	Type	必填	描述
token	字符串	是	标识此操作的标记。 必须将令牌返回到

字段	Type	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。
provisionedProduct Name	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。
身份	字符串	否	该字段目前未被使用。

字段	Type	必填	描述
参数	列表	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	列表	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	Type	必填	描述
token	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。
provisionedProduct Name	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	Type	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。
身份	字符串	否	该字段目前未被使用。
参数	列表	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	列表	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 [NotifyUpdateProvisionedProductEngineWorkflowResult](#) API 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	Type	必填	描述
token	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 <code>TERMINATE_PROVISIO</code>

字段	Type	必填	描述
			N_PRODUCT 用于此操作。
provisionedProductId	字符串	是	预配置产品的 ID。
provisionedProduct Name	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)。
身份	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。

标签

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。

监视中 AWS Service Catalog

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 为可读指标。这些统计数据会记录两周，这样您就可以访问历史信息并更好地了解您的服务绩效。AWS Service Catalog 指标数据会 CloudWatch 在 1 分钟内自动发送到。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](#)。

监控是维护 AWS 解决方案的可靠性、可用性和性能的重要组成部分。AWS Service Catalog 您应该从 AWS 解决方案的所有部分收集监控数据，以便在出现多点故障时可以更轻松地进行调试。在开始监控之前 AWS Service Catalog，您应该创建一个包含以下问题的答案的监控计划：

- 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 提供了各种可用于监控的工具 AWS Service Catalog。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 A EC2 uto Scaling 政策的通知。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。这些统计数据会记录两周，这样您就可以访问历史信息并更好地了解您的服务绩效。AWS Service Catalog 指标数据会 CloudWatch 在 1 分钟内自动发送到。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	说明
ProvisionedProductLaunch	<p>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。这些维度作为单独的记录发布在 CloudWatch 日志中。</p> <p>单位：Count</p> <p>有效统计数据：Minimum、Maximum、Sum、Average</p> <p>维度：State、PPState、ProductId、ProvisioningArtifactId</p>
ProductProvisioningOperation	<p>对产品编号执行的操作次数provisioningArtifactId。维度在 CloudWatch 日志中作为一条记录发布。</p>

指标	说明
	单位 : Count 有效统计数据 : Minimum、Maximum、Sum、Average 维度 : State、PPState、ProductId 、 ProvisioningArtifactId

AWS Service Catalog 指标的维度

AWS Service Catalog 向 Amazon 发送以下尺寸 CloudWatch。

维度	说明
PPState	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。 有效状态 : 可用、已污染、错误
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。
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

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 Amazon 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。

使用记录 AWS Service Catalog API 调用 AWS CloudTrail

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。捕获的调用包括来自 AWS Service Catalog 控制台的调用和对 AWS Service Catalog API 操作的代码调用。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。您可以在自己的 AWS 账户中查看、搜索和下载最近发生的事件。有关更多信息，请参阅[使用事件历史记录查看 CloudTrail 事件](#)。

要持续记录您 AWS 账户中的事件，包括的事件 AWS Service Catalog，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，在控制台中创建跟踪记录时，此跟踪记录应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。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：

- [创建跟踪记录概述](#)
- [AWS CloudTrail 支持的服务和集成](#)
- [为 AWS CloudTrail配置 Amazon SNS 通知](#)
- [接收来自多个地区的 AWS CloudTrail日志文件](#)和[接收来自多个账户的 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](#)

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根证书还是 AWS Identity and Access Management (IAM) 用户凭证发出。
- 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 boto3/1.17.63",
"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偏好。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 支持主机品牌的区域列表，请查看[对主机品牌的AWS 区域 支持](#)。

AWS 区域 支持主机品牌偏好

AWS Service Catalog 支持下表所 AWS 区域 列的主机品牌偏好。

AWS 区域 名字	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 名字	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 名字	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	

AWS Service Catalog API 概述

使用 Service Catalog API 的好处

该 AWS Service Catalog API 提供对所有最终用户操作的编程控制，以此作为使用的一种替代方案 AWS 管理控制台。使用 API 时，您可以执行以下操作：

- 编写自己的自定义界面和应用程序
- 获得对最终用户产品配置操作的精细控制
- 将资源配置集成到您的编排管道中
- 访问托管应用程序及其资源的中心位置

访问 Service Catalog

要使用特定语言的 API 构建应用程序，请使用面向软件开发人员的库、示例代码、教程和其他资源。这些库文件提供可自动执行任务的基本功能，例如以加密方式对请求签名、重试请求和处理错误响应，因此您可以更轻松地上手。要开始使用，请打开适用于 [Amazon Web Services 的工具](#)，然后在 SDK 下找到您选择的软件开发工具包。

如果倾向于使用命令行界面，您可使用以下选项：

AWS 命令行界面 (CLI)

要开始使用，请参阅 [《AWS Command Line Interface 用户指南》](#)。有关 Service Catalog 命令的更多信息，请参阅《命令参考》AWS CLI 中的 [servicecatalog](#)。

AWS 适用于 Windows 的工具 PowerShell

要开始使用，请参阅 [《AWS Tools for PowerShell 用户指南》](#)。有关 Service Catalog 的 cmdlet 的更多信息，请打开 [AWS Tools for PowerShell Cmdlet 参考](#) 并展开。AWS Service Catalog

从逻辑上讲，AWS Service Catalog API 可以分为以下几类。

主题

- [产品发现](#)
- [置备请求](#)
- [预配置产品](#)

- [预配置产品计划](#)
- [产品组合](#)
- [校长协会](#)
- [产品](#)
- [资源调配工件](#)
- [约束](#)
- [服务操作](#)
- [TagOptions](#)
- [AppRegistry](#)
- [工作流程示例](#)

产品发现

使用这些操作来发现或获取有关产品及其发布要求的信息。这些操作不会创建或修改资源。

[SearchProducts](#)

列出来电者有权访问的所有产品。

[DescribeProduct](#)

获取有关产品的详细信息。

[DescribeProductView](#)

功能上与相同DescribeProduct，不同之处在于它使用的是产品视图的 ID 而不是产品的 ID。

[ListLaunchPaths](#)

列出用户访问指定产品的所有方式，称为产品路径。用户必须选择路径才能配置产品。

[DescribeProvisioningParameters](#)

获取预配置指定产品所需的参数，并提供有关配置产品时会发生什么情况的其他元数据。

每个ProvisioningArtifactParameter都是用户为成功配置产品而必须指定的内容（例如，EC2 实例的大小）。这些ConstraintSummary对象包含允许值的列表以及有关这些ProvisioningArtifactParameter对象的其他元数据。

置备请求

使用这些操作来请求、更新或终止产品的配置。

[ProvisionProduct](#)

请求为产品预配。配置产品就是启动使该产品上线以供实际使用所需的资源。例如，配置由 CloudFormation 模板支持的产品意味着启动 CloudFormation 堆栈及其所有底层资源。

[UpdateProvisionedProduct](#)

更新预配置产品的配置。例如，由支持的产品 CloudFormation 会更新其底层 CloudFormation 堆栈。请求者必须拥有对指定 ProvisionedProduct 内容的足够访问权限。

[TerminateProvisionedProduct](#)

请求终止预配置的产品。例如，对于由支持的产品 CloudFormation，这会删除底层 CloudFormation 堆栈。申请者必须拥有对指定预配置产品的足够访问权限。

预配置产品

使用这些操作来获取有关预配置产品的信息。这些操作不会创建或修改资源。

[ListRecordHistory](#)

列出所有已执行的请求，即使是已终止的预配置产品也是如此。

[DescribeRecord](#)

获取有关请求的信息。在请求操作之后使用此操作来获取当前 RecordDetail 信息。

[SearchProvisionedProducts](#)

获取有关符合指定条件的预配置产品的信息。

[ScanProvisionedProducts](#)

列出未终止的预配置产品。

[DescribeProvisionedProduct](#)

获取有关预配置产品的信息。

[ImportAsProvisionedProduct](#)

请求将资源作为与 Service Catalog 产品和配置对象关联的 Service Catalog 预配置产品导入。导入后，预配置的产品将支持所有支持的 Service Catalog 管理操作。

[UpdateProvisionedProductProperties](#)

请求更新指定预配置产品的属性。

预配置产品计划

使用这些操作来管理您的预配置产品计划。计划包括执行计划时要创建或修改的资源列表。

[CreateProvisionedProductPlan](#)

创建计划。

[DescribeProvisionedProductPlan](#)

获取有关计划资源变化的信息。

[ExecuteProvisionedProductPlan](#)

根据计划配置或修改产品。

[ListProvisionedProductPlans](#)

列出预配置产品的计划。

[DeleteProvisionedProductPlan](#)

删除计划。

产品组合

目录管理员使用这些操作作为产品组合管理提供所有必要的操作。

[CreatePortfolio](#)

创建产品组合。

[DeletePortfolio](#)

删除投资组合。

[DescribePortfolio](#)

获取有关投资组合的详细信息。

[DescribePortfolioShares](#)

返回为指定投资组合创建的每份投资组合份额的摘要。

[ListPortfolios](#)

列出目录中的所有投资组合。

[ListPortfoliosForProduct](#)

列出与产品关联的所有产品组合。

[UpdatePortfolio](#)

更新投资组合。

[UpdatePortfolioShare](#)

更新投资组合份额。

[CreatePortfolioShare](#)

与 AWS 账户共享投资组合。

[DeletePortfolioShare](#)

停止共享投资组合。

[AcceptPortfolioShare](#)

接受共享投资组合的提议。

[RejectPortfolioShare](#)

拒绝共享投资组合的提议。

[ListAcceptedPortfolioShares](#)

列出该账户接受共享的所有投资组合的详细信息。

[ListPortfolioAccess](#)

列出有权访问投资组合的账户 ID。

校长协会

目录管理员使用这些操作为主体关联提供所有必要的操作。

[AssociatePrincipalWithPortfolio](#)

将委托人 ARN 与投资组合相关联。

[DisassociatePrincipalFromPortfolio](#)

取消本金 ARN 与投资组合的关联。

[ListPrincipalsForPortfolio](#)

列出与投资组合关联的所有委托人 ARN。

产品

目录管理员使用这些操作作为产品管理提供所有必要的操作。

[SearchProductsAsAdmin](#)

获取产品的摘要和状态信息。

[DescribeProductAsAdmin](#)

获取有关产品的信息。

[CreateProduct](#)

创建产品。

[CopyProduct](#)

复制产品。

[DescribeCopyProductStatus](#)

获取复制产品操作的状态。

[UpdateProduct](#)

更新产品。

[DeleteProduct](#)

删除产品。

[AssociateProductWithPortfolio](#)

将产品与产品组合相关联。

[DisassociateProductFromPortfolio](#)

取消产品与产品组合的关联。

资源调配工件

目录管理员使用这些操作来管理配置对象（也称为产品版本）。

[DescribeProvisioningArtifact](#)

获取有关置备构件的信息。

[CreateProvisioningArtifact](#)

为产品创建配置对象。

[DeleteProvisioningArtifact](#)

删除置备对象。

[ListProvisioningArtifacts](#)

列出与产品关联的所有配置对象。

[UpdateProvisioningArtifact](#)

更新置备工件。

约束

目录管理员使用这些操作来管理约束。

[CreateConstraint](#)

创建约束。

[DeleteConstraint](#)

删除约束条件。

[DescribeConstraint](#)

获取有关约束条件的信息。

[UpdateConstraint](#)

更新约束条件。

[ListConstraintsForPortfolio](#)

获取产品组合和产品的约束信息。

服务操作

目录管理员使用这些操作来管理服务操作。

[AssociateServiceActionWithProvisioningArtifact](#)

将自助操作与置备对象关联。

[CreateServiceAction](#)

创建自助服务操作。

[DeleteServiceAction](#)

删除自助服务操作。

[DescribeServiceAction](#)

描述自助服务操作。

[DescribeServiceActionExecutionParameters](#)

查找针对特定预配置产品的特定自助操作的默认参数，并将结果映射返回给用户。

[ExecuteProvisionedProductServiceAction](#)

对已配置的产品执行自助操作。

[UpdateServiceAction](#)

更新自助服务操作。

TagOptions

目录管理员使用这些操作进行管理 TagOptions。

[CreateTagOption](#)

创建一个 TagOption.

[ListTagOptions](#)

列出你的 TagOptions.

[DescribeTagOption](#)

描述一个 TagOption.

[UpdateTagOption](#)

更新 a TagOption.

[AssociateTagOptionWithResource](#)

将 TagOption 与资源关联。

[DisassociateTagOptionFromResource](#)

取消与资源的关联。 TagOption

[ListResourcesForTagOption](#)

列出 a 的资源 TagOption。

[DeleteTagOption](#)

删除 a TagOption.

AppRegistry

用作您的应用程序、其资源以及您在企业中使用的应用程序元数据的存储库。

[AssociateAttributeGroup](#)

将属性组与应用程序关联，以使应用程序的元数据与该组的属性相结合。

[AssociateResource](#)

将资源与应用程序关联。

[CreateApplication](#)

创建一个新应用程序，该应用程序是相关云资源抽象层次结构中的顶级节点。

[CreateAttributeGroup](#)

创建一个新的属性组作为用户定义属性的容器。

[DeleteApplication](#)

删除由其应用程序 ID 或名称指定的应用程序。

[DeleteAttributeGroup](#)

删除由其属性组 ID 或名称指定的属性组。

[DisassociateAttributeGroup](#)

取消属性组与应用程序的关联，以从应用程序的元数据中移除该属性组中包含的额外属性。

[DisassociateResource](#)

取消资源与应用程序的关联。

[GetApplication](#)

检索有关您的一个应用程序的元数据信息。

[GetAssociatedResource](#)

获取与应用程序关联的资源。

[GetAttributeGroup](#)

按名称或 ID 检索属性组。

[ListApplications](#)

列出与指定应用程序关联的所有属性组。

[ListAssociatedAttributeGroups](#)

列出与指定应用程序关联的所有属性组。

[ListAssociatedResources](#)

列出与指定应用程序关联的所有资源。

[ListAttributeGroups](#)

列出您有权访问的所有属性组。

[ListAttributeGroupsForApplication](#)

列出与特定应用程序关联的所有属性组的详细信息。

[ListTagsForResource](#)

列出资源上的所有标签。

[TagResource](#)

为指定资源分配一个或多个标签（键值对）。

[SyncResource](#)

将资源与当前录制的内容同步。AppRegistry

[UntagResource](#)

删除资源的标签。

[UpdateApplication](#)

使用新属性更新现有应用程序。

[UpdateAttributeGroup](#)

使用新的详细信息更新现有属性组。

工作流程示例

在这种情况下，管理员使用 AWS Service Catalog 创建资源，最终用户查找可用产品并配置产品。这是一个示例工作流程；这不是使用 AWS Service Catalog API 的唯一方法。

管理员任务

- 创建产品组合、产品视图、产品、产品版本和约束条件。
- 将 IAM 用户分配给产品，这样他们就可以访问了。

最终用户任务

1. 用户调用[SearchProducts](#)时不带任何参数。这将返回用户有权访问的产品列表，以及可用于确定结果范围的 SearchDomain ""。
2. 用户继续使用其他搜索过滤器[SearchProducts](#)进行呼叫，直到找到所需的产品。
3. 用户打电话[DescribeProductView](#)以查找此产品的配置对象（也称为版本）列表。这决定了用户实际提供的内容。
4. 用户调用[ListLaunchPaths](#)以查找此产品的路径列表以及每条路径的限制。这决定了对预配置产品应用哪组约束。

5. 选择配置对象和路径后，用户调用[DescribeProvisioningParameters](#)。这将返回用户在使用配置工件和路径配置产品之前必须提供的参数列表，以及管理员决定提供的任何其他使用说明。
6. 用户调用 [ProvisionProduct](#)，指定产品、配置对象、路径和输入参数。输入参数是键值对的列表，其中密钥是使用获取的 [DescribeProvisioningParameters](#)，值由用户提供（例如）。`{ParameterKey:"dbpassword", ParameterValue:"mycoolpassword"}`这将启动创建指定 AWS 资源的工作流程。它还会创建用于跟踪置备请求的记录详细信息以及代表底层 AWS 资源的预配置产品对象。
7. 用户进行轮询[DescribeRecord](#)以查看记录详细信息的状态何时从IN_PROGRESS状态变为已完成状态（SUCCEEDED或者ERROR）。
8. 当请求的记录详细信息处于已完成状态时，用户会[DescribeRecord](#)再次调用。输出标识了创建的资源。
9. 用户调用[UpdateProvisionedProduct](#)以更新现有的底层资源。根据请求的特定更新，此操作可以不间断地更新，但会有一些中断，也可以完全替换预配置的产品。
10. 最后，用户调用终[TerminateProvisionedProduct](#)止已配置的产品。

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 Catalog Amazon 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。