在上实施基础架构即产品 (IaP) AWS

AWS 规范性指导

AWS 规范性指导: 在上实施基础架构即产品 (IaP) AWS

Table of Contents

在 AWS 上实现 IaP

Kirsten Kissmeyer，Amazon Web Services（AWS）

2023 年 1 月（文档历史记录）

本指南探讨了管理 AWS 基础设施即产品（IaP）的方法。与基础设施即代码（IaC）相比，IaP 提供了更高级别的抽象和控制，但使用 IaC 方法来实现其目标。该指南还探讨了 AWS 服务和用于管理 IaP 的工具，并重点介绍了每种工具如何支持您管理基础设施的目标。本指南中的信息基于从一家大型金融行业公司的 AWS Service Catalog 赋能计划中吸取的经验。

本指南适用于想要开发功能性 AWS Cloud基础设施服务的用户，这些服务可以根据需要轻松分配和授权给不同的组织用户、业务部门和第三方。

为什么要将基础设施作为产品进行管理？

将基础设施资源作为产品进行管理的优势在于，您可以将消费者功能打包为一组具有标准化定义和配置的资源。产品为组织提供了一种便捷的方式来管理和控制 AWS 功能的分配和使用方式。产品可能仅限于指定的组织单位（OU）或需要这些功能能力的个人使用。也可以将产品限制为特定 AWS 区域。

产品预调配模型还允许您从中心位置封装和更新产品的定义。然后，您可以一次性或定期分发产品更新，因为其实施会随着时间的推移而发生变化。

目标业务成果

各组织一直在寻找更好的方法来管理和预调配其 AWS 基础设施。您的目标可能包括：

- 实现高度的敏捷性、可靠性、容错能力和集中控制，其中单点配置可以满足不断变化的内部和外部标准的合规性。
- 一种低接触或一键式的机制，用于以集中方式分发基础设施，同时允许特定团队或个人在需要时进行自助访问。
- 能够为内部员工、客户账户和合作伙伴 OU 账户预调配 AWS 基础设施和服务。您可能还要控制哪些 OU 或组织可以访问特定区域中的特定基础设施组件。
- 如果您使用第三方工具（例如 ServiceNow）或自定义工具来管理访问和预调配企业资产和基础设施的请求，则您的 AWS 基础设施与这些工具之间可以轻松集成。
- 能够同时为数十甚至数百个目标账户预调配 AWS 基础设施。

- 支持预调配多个 AWS 资源以提供单一功能。
- 能够在紧张的时间表内使用所需的基础设施创建新账户。
- 访问您已预调配的基础设施清单，并能够更新或删除基础设施组件。
- 使预调配和维护过程更简单、更快速、更安全、更可靠的方法和技术。

AWS Service Catalog 用于管理 IaP

AWS 提供一项名为的服务 AWS Service Catalog，该服务支持将 AWS 基础架构作为产品进行管理和配置。您可以使用 Service Catalog 快速定义需要预调配为一组产品的基础设施，将这些产品的权限授予所需方，并实现单个产品所需的预调配和更新模式。

Service Catalog 由 AWS CloudFormation 提供支持。Service Catalog 产品组合、产品及其配置模板作为 CloudFormation 堆栈进行管理。您可以通过以下四种方式定义这些堆栈：

- 通过使用标准 CloudFormation 模板。
- 通过使用 AWS Cloud Development Kit (AWS CDK)和 Service Catalog 构造库，并结合您偏好的受支持的编程语言。
- 使用第三方工具提供的框架，根据描述 CloudFormation 堆栈的声明性元数据生成堆栈定义。
- 通过使用 Service Catalog API。此 API 提供了除构建产品之外的所有方法。您可以将产品添加到产品组合、从产品组合中删除产品、标记产品和产品组合、定义管理和运营产品服务操作，以及浏览和搜索产品组合和产品定义。

从本质上讲，Service Catalog 产品是一组由一个或多个 AWS 资源组成的集合，这些资源配置为提供集体、可自定义（通过参数化）功能。例如，您可以定义 Service Catalog 产品，在目标账户中预调配私有 Amazon Simple Storage Service（Amazon S3）存储桶。S3 存储桶是一种可能包含输入参数的产品，例如存储桶名称、允许从中访问的互联网地址范围、一组可以访问存储桶的用户、生命周期分层策略或存储桶版本控制规范。您还可以定义一个 AWS Identity and Access Management (IAM) 角色来提供对作为产品一部分的存储桶的访问权限。

您可以将 Service Catalog 产品添加到一个或多个产品组合。Service Catalog 产品组合是将产品组合在一起的集合，通常是因为它们的用途相似（例如，分析、开发、客户访问服务、合作伙伴访问服务等）。

您可以为用户、组或角色提供在产品组合级别预调配产品的权限。对于预调配，产品要么与启动 IAM 角色关联（以便以自助方式向可以代入该角色的任何人启动产品），要么与堆栈集关联，该堆栈集定义了一个或多个可以预调配该产品的账户。要使用堆栈集，您必须在 Service Catalog 中心账户中定义一个 Service Catalog 管理员角色，并在堆栈集的每个目标账户中定义一个 Service Catalog 产品预调配执行角色。

以下各部分将更详细地讨论 Service Catalog IaP 功能。

主题

支持模块化和代码重复使用

您可以从许多不同的 AWS 资源甚至其他产品中组装产品。理想情况下，您可以采用模块化方式定义资源，以便可以在多个产品中重复使用它们。资源级重复使用使您能够在一个地方进行任何未来更改，而不是在使用该资源类型的每个产品中进行更改。

Service Catalog 提供了一项称为链接的功能，以支持产品级别的可重用性。您可以将产品链接到一个或多个其他产品。例如，您可能希望将 S3 日志存储桶产品链接到更高级别的监控产品。虽然链接支持模块化，但它会带来一些操作复杂性，因为您必须管理依赖关系。Service Catalog 不会自动维护链式产品之间的版本控制，因此无法确保对一个产品的更改不会破坏依赖于它的其他产品。谨慎使用链接，并开发自己的机制来确保版本控制和维护依赖关系。

Serv CloudFormation ice Catalog 使用本机方式将产品配置模板部署为 CloudFormation堆栈。但是，Service Catalog 对产品堆栈的 CloudFormation 部署施加了一些限制。特别是，Service Catalog 配置不支持用于插入可重复使用的脚本段或将嵌套 CloudFormation 脚本（或堆栈）引用到多个级别的 CloudFormation `include`宏。这些 Service Catalog 限制限制了根据可重复使用的 CloudFormation 模板或组件定义产品的能力，这是您在本地定义堆栈时的标准最佳做法。 CloudFormation

> **Note**
>
> Service Catalog 允许您使用使用这些 CloudFormation 结构的配置模板成功定义产品。但是，如果您在 Service Catalog CloudFormation 模板中使用`include`宏或嵌套多个级别的脚本，则会遇到配置时间错误。

这些限制可能会使在 Service Catalog 中实现模块化和可重复使用的产品变得困难。如果需要模块化，您可以探索使用 AWS CDK 来实现您的产品及其预调配模板，或者使用 AWS Labs Service Catalog Tools 项目中的预调配工作流程和引擎。本指南的后续部分中描述了这两种替代方案。

用于在 Service Catalog 中定义产品的编程选项

使用 Service Catalog 配置 AWS 基础架构的两个编程选项是 CloudFormation模板或 AWS CDK。当前，没有用于定义 Service Catalog 产品的声明式或无代码机制。

CloudFormation 脚本

CloudFormation 是一种久经考验的 IaC 原生脚本语言，用于配置 AWS 基础架构。您可以使用诸如 Visual Studio Code（ AWS Management Console 或简单的文本编辑器）和 () 之类的开发工具在或中 AWS Command Line Interface 开发 CloudFormation 脚本。AWS CLI

有关详情，请参阅 CloudFormation 文档。有关使用 CloudFormation 模板指定 Service Catalog 产品的更多信息，请参阅 CloudFormation 文档中的AWS::ServiceCatalog::CloudFormationProduct 资源。

采用编程方法 AWS CDK

AWS CDK 提供了一个优雅而强大的面向对象的编程框架，用于使用多种编程语言来定义和维护 AWS 基础架构。您可以使用 AWS CDK 为类框架开发面向对象、细粒度的自定义项和扩展。 AWS AWS CDK 适用于想要针对更复杂的基础架构需求 AWS 服务 进行定制且具有必要编程技能和经验的用户。

要使用实施 Service Catalog 解决方案 AWS CDK，您可以使用内置的 Service Catalog 类来定义您的产品和产品组合。这些类由 AWS CDK aws-cdk-lib.aws_ servicecatalog 模块提供。

您可以通过多种方式使用 AWS CDK 来实现产品。为了避免在中为产品编写配置模板 CloudFormation 并保持可重用性，我们建议您使用该 AWS CDK ProductStack类来表示配置模板。`ProductStack`实例是您以编程方式向其添加资源的 AWS CDK 堆栈。例如，您可以添加 S3 存储桶、IAM 角色或 Amazon CloudWatch 日志。当您通过调用将该`ProductStack``servicecatalog.CloudFormationProduct`实例作为其配置模板添加到已定义的实例时`servicecatalog.CloudFormationTemplate.fromProductStack (<ProductStack instance>)`，会 AWS CDK 自动生成该 CloudFormation模板。

以下是 Amazon S3 产品的 Java `ProductStack` 实现示例。

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

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

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

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

AWS CDK 提供了内置的持续集成和持续部署 (CI/CD) 管道。您可以自定义这些内置管线和软件开发生命周期（SDLC）流程，以满足您自己的流程标准和目标。

自定义 AWS CDK 类可以继承其他类以提供专门的函数，而一个类可以由其他类的实例组成。如果您使用共享 AWS CDK 类框架来实现多个 Service Catalog 产品，请考虑任何版本控制或兼容性影响，尤其是在多个开发团队之间。您必须确保更改向后兼容，或者您具有要遵循的版本控制方案，以便您为一个产品所做的类别更改不会破坏另一个产品。

有关详情，请参阅 AWS CDK 文档。

与外部预调配流程和工作流程集成

您可以使用 AWS SDK APIs 或 Service Catalog 组件进行交互 AWS CLI。您可以使用 AWS SDK Service Catalog API 通过任何可以集成 Service Catalog API 调用的工具管理 Service Catalog 产品。该 API 涵盖了 Service Catalog 创建和管理的所有方面。例如，Terraform 通过在其 Launch Wizard 中调用 AWS SDK Service Catalog API 来支持启动（配置）服务目录产品。有关更多信息，请参阅文档中的使用 Terraform 启动 AWS Service Catalog 产品。 AWS

您也可以调用 S AWS CLI ervice Catalog 命令对服务目录执行操作。有关支持的命令的更多信息，请参阅《命令参考》中的 s ervicecatalog。 AWS CLI

产品预调配规格

Service Catalog 将置备过程作为预 CloudFormation 配模板中指定的资源的 CloudFormation 堆栈集部署来启动。（模板可以直接在构造中创建， CloudFormation 也可以由 AWS CDK `ProductStack`构造生成。）Service Catalog 产品预调配是一个封闭的过程 – 您无法对其进行自定义以添加预备步骤或后处理步骤，也无法对其进行调整。但是，您可以修改配置模板以添加 CloudFormation 资源规格形式的步骤。这些资源可以是 AWS Lambda 或 AWS Step Functions，或者是 Lambda 支持的自定义资源，用于执行初步步骤（例如自定义引导以设置在配置期间使用的堡垒主机）和后期步骤（例如拆除堡垒主机）。这种实现预调配前和预调配后步骤的方法与预调配模板一样，都受到相同的 `include` 和嵌套堆栈限制。

您可以将目标帐户指定为个人帐户，而不是组织单位 (OUs)。您可以编写自定义资源或函数来解决此限制。大多数组织向个人账户提供产品组合 OUs ，而不是向个人账户提供产品组合，因为他们会自动生成账户，并且不想手动维护账户列表。

DevSecOps 生命周期支持

目前，使用 Service Catalog CloudFormation 脚本配置的产品没有对 CI/CD 流程的内置支持。我们建议您在中创建 CI/CD 流程 AWS CodePipeline 或其他 DevOps 工具，以便在开发、测试、阶段和生产等生命周期环境中开发、测试和发布产品。

如本指南前面所述， AWS CDK 确实为产品提供了内置 CI/CD 支持。

自定义重复使用和特定于账户的预调配

产品应可重复使用，以用于尽可能多的不同自定义目的。Service Catalog 通过产品参数支持可重复使用。您可以在预调配时将这些参数作为输入提供给产品。

您也可以在 CloudFormation模板级别将这些参数指定为 P AWS Systems Manager arameter Store 值，以应用账户特定值和 OU 特定值。这是 CloudFormation 配置模板设计的最佳实践。预调配产品时，将应用目标账户内命名的参数值。例如，您可以将子网参数指定为 Parameter Store 值，并在产品预调配时将该子网应用于特定 OU 账户。有关参数存储值作为 CloudFormation 模板参数的更多信息，请参阅 CloudFormation 文档中的使用动态引用指定模板值。

将 Service Catalog 产品资源定义为应用程序并进行管理

AWS Service Catalog AppRegistry 提供集中式应用程序搜索、报告和管理功能。一个 AppRegistry 应用程序可以包括一个或多个预配置的产品堆栈以及独立于 Service Catalog 的 CloudFormation堆栈。您

可以对定义为部署目标的所有应用程序资源集合 AWS 账户 进行分组和查看。这些账户可以是您的开发、测试和生产生命周期账户。

您还可以使用 AppRegistry 将元数据属性与应用程序相关联。您可以分配包含属性集的可重复使用的属性组。然后，您可以使用 AppRegistry 或集成服务搜索具有给定属性的应用程序资源并对其进行操作。这些集成的服务包括：

- App@@ lication Manager AWS Systems Manager，一种功能，用于调查和修复应用程序和集群环境中的 AWS 资源问题
- AWS Resource Access Manager，与 AWS 组织负责人共享应用程序和属性组
- AWS 与之配合使用的服务 AWS Resource Groups
- AWS Resilience Hub，用于产品结构发现和韧性评测
- AWS 服务管理连接器声明和设置与 ServiceNow、JIRA 和其他常用工具的连接

有关的更多信息 AppRegistry，请参阅以下内容：

- AppRegistry 管理员指南。
- Increase application visibility and governance using AWS Service Catalog AppRegistry 博客文章。本文概述了如何在基础设施管理 AppRegistry 中使用，并提供了在中注册基础架构为应用程序的命令行示例。 AppRegistry
- 使用 AppRegistry 应用程序管理器博客文章集中管理您的应用程序。本文概述了 AppRegistry 如何申请在上注册 LAMP Web 应用程序 AWS Management Console 并使用应用程序管理器对其进行管理。

库存管理

Service Catalog 拥有自己的内部库存管理功能，可在通过产品共享和自助服务预调配产品时对其进行注册。但是，我们建议您使用AWS Config或AppRegistry和相关服务来管理您的产品配置资源。这些工具提供了一种更全面、更集成的方法，可以将预配置的 Service Catalog 产品与 AWS 基础架构的其余部分一起管理。 AWS Config 允许您在控制台上或使用 AWS SDK API 对预配置产品进行库存和执行操作。 AppRegistry与 Application Manager 集成，还为 Service Catalog 预配置的产品提供库存管理。

使用 AWS Service Catalog 工具

如果您希望以更具声明性的方式使用自定义预调配工作流来预调配 IaC 产品，则可能需要增强 Service Catalog 的部分功能。AWS 提供了多种工具来支持这些需求。AWS Labs 项目中提供了两个常用的工具：Service Catalog Puppet 和 Service Catalog Factory。

主题

Service Catalog Puppet

Service Catalog Puppet 是使用 AWS Boto3 API 在 Python 中实现的。此工具为配置和预调配 Service Catalog 产品提供了多项强大功能。开发者可以使用用作清单的 YAML 模板来配置 Service Catalog 产品和产品组合预调配信息。Service Catalog Puppet 预调配工作流程支持需要比 Service Catalog 更复杂的部署流程的产品。它们还支持性能优化，以便在紧迫的时段内大规模预调配产品。

Service Catalog Puppet 在部署时访问 Service Catalog CloudFormation 模板以进行产品预调配。它直接调用 CloudFormation 来部署产品的预调配模板堆栈，并绕过 Service Catalog 自身堆栈集预调配过程施加的限制。如果预调配模板使用宏来包含其他 CloudFormation 脚本或使用嵌套的 CloudFormation 脚本，您必须在预调配工作流程的引导部分提供对目标账户中这些脚本的访问权限。

有关更多信息：

- 请参阅 Service Catalog Puppet 文档和 GitHub 存储库。
- 如果您想使用 Service Catalog Puppet SDK 以编程方式与该工具交互以启动产品和产品组合预调配，请参阅 SDK 文档。
- GitOps 是管理 Service Catalog Puppet 环境的默认机制。

对开发者来说，Service Catalog Puppet 相当容易学习。要实现产品预调配模板，需要熟悉 CloudFormation；要实现清单，需要熟悉 YAML 模板。有不错的讲习会可以让新开发者快速上手，比如自主进度讲习会。

支持预调配工作流程

Service Catalog Puppet 使用 Python Luigi 任务编排引擎来实现引导和预调配工作流程。这些工作流程中的所有步骤都作为 Luigi 工作流程任务来实现。有关 Luigi 的概述以及它与其他热门工作流程工具的比较，请参阅数据收入博客上的 Airflow vs. Luigi vs. Argo vs. MLFlow vs. KubeFlow。

Luigi 允许 Service Catalog Puppet 控制与工作流程任务关联的 Worker 数量，并控制工作流程的其他方面，以实现更好的扩展和性能。Service Catalog Puppet 还提供了一种 depends_on 机制，用于管理产品和步骤依赖关系以及编排产品预调配。此功能可帮助您实施和运营管理精细的产品定义和复杂的依赖关系。

预调配模型

Service Catalog Puppet 支持以下三种执行模式：中心、分支和异步。这三种模式都在 Service Catalog 中已定义的产品组合内预调配产品。它们依赖 Service Catalog 产品共享将产品部署到目标账户，并使用 Service Catalog 管理员和启动角色在这些目标中实现预调配。Service Catalog Puppet 根据 YAML 配置文件中提供的角色配置在同一个组织内执行引导步骤。该工具还支持通过单个中心账户向多个组织进行预调配。在这种情况下，必须在外部组织中手动执行引导，以允许 Service Catalog Puppet 在外部组织的账户中执行所需的预调配操作。

在所有预调配模式下，Service Catalog Puppet 无需调用 Service Catalog 的预调配流程即可直接实现产品预调配。您可以将预调配清单配置为使用现有 Service Catalog 堆栈集约束中的角色和目标账户规范。Service Catalog Puppet 使用此信息通过 Luigi 工作流程进行自身的预调配。

除了直接指定 OU 或账户外，您还可以根据账户标记方法定义产品组合预调配的目标。在基于账户标签的预调配中，产品组合产品将预调配给具有指定清单预调配标签集中所有标签的所有账户。例如，如果您想向美国东部区域的所有机构生产账户发行投资组合产品，则可以指定标签 `type:prod`、`partition:us-east` 和 `scope:institutional-client`。您还可以声明账户和 OU 排除项，以禁止向 OU 或具有您指定标签的账户，或者向属于 OU 指定目标成员的账户进行预调配。有关账户标记的更多信息，请参阅 Service Catalog 工具文档。

中心模式

在中心预调配模式下，分支账户的所有 Luigi 工作流程都通过指定的中央中心账户进行管理。中心账户扮演一个 IAM 角色，允许其在分支账户中执行操作，但任务管理是在中心账户内进行的。中心账户将同步等待，直到所有分支账户预调配任务完成（无论成功还是失败）。然后，它会报告最终状态。中心账户模式是最古老、最成熟的预调配模式。但是，许多用户已转向分支预调配模式，以实现更高的预调配并发性和速度。

分支模式

在分支模式下，Service Catalog 中心账户启动 Luigi 工作流程，使其在所指定引导的分支账户中运行。分支工作流程完成时，中心账户会收到通知。分支账户中的故障会上报到中心账户。中心账户会轮询分支账户，以查看其是否已完成并确定状态。

由于几乎所有服务都在独立的分支账户中运行，因此分支模式最不可能需要增加 AWS 服务配额。与中心模式相比，分支模式还可提供更高的并发性，同时保留集中控制。与中心模式相比，它可以将预调配速度提高 800%。分支模式支持通过产品之间的 DependsOn 关系实现产品链，从而确保所依赖的产品已经预调配。由链式产品组成的产品也可以预调配组件链式产品。您也可以使用专门的 AWS Lambda 函数调用来执行所需的步骤。一个分支中的故障与其他分支相隔离。

拥有超过 980 个账户、遍布多达 7 个区域的企业可以使用分支模式。这些企业通常能够在一小时内向基础设施中的所有区域和账户提供产品。

> **Note**
>
> 这些结果可能会因网络基础设施、工作负载以及 AWS 组织中心和分支账户的配额等因素而异。它们还取决于正在预调配的产品资源、其固有的创建时间及对其他资源的依赖性。

异步模式

异步模式在分支账户中启动预调配工作流程，但它不会等待或接收来自分支的完成响应。

缓存

Service Catalog Puppet 用来优化工作流程速度的另一种机制是缓存代表工作流中步骤的常见任务。在缓存的任务完成时，它会将其输出写入 Amazon Simple Storage Service（Amazon S3）。下次在同一会话中使用相同的参数调用任务时，Service Catalog Puppet 将使用缓存的值而不是重新运行该任务。有关更多信息，请参阅 Service Catalog Puppet 文档中的 Caching。

DevSecOps 生命周期支持

Service Catalog Puppet 包括对管理 DevSecOps 管线的支持。您可以使用 Service Catalog 工具操作（如 Service Catalog Puppet 概述中所示）来自动化测试，并在您的 AWS 生命周期账户（包括推荐的金丝雀账户）中推广产品。有关更多信息，请参阅 Service Catalog Puppet 文档中的 Managing your environments。

为了确保在广泛生产使用之前检测到与产品更改相关的任何问题，Service Catalog Puppet 要求至少有一个金丝雀账户进行初始部署。在您测试新版本并获得对新版本的信心之后，您可以将其推广到非金丝雀生产账户。如果您发现任何问题，则可以回滚版本，并在问题解决时重新引入该版本。使用此方法时，如果将存在问题的金丝雀版本发布到生产账户，则可能会出现生产问题。另一种替代方法是，在将更改发布到生产环境之前，可以对每个产品更改运行完整的回归测试。这会给 CI/CD 流程带来额外的开销，但有助于避免生产问题。DevSecOps 管理员有责任为其开发团队确定最佳的功能发布场景和方法。

Service Catalog Puppet 允许多个团队同时开发和测试 Service Catalog 产品解决方案的预调配。作为最佳实践，一个产品不应由多个开发者同时更改。相反，您可以将产品分解成更精细的组件，以进行单独的同步修改。

Service Catalog Puppet 还通过提供静态和单元测试功能的断言语句来帮助自动进行测试。您还可以在策略模拟器中测试服务控制策略（SCP）和 IAM 策略。从技术上讲，这些是端到端测试，但可以在系统集成测试（SIT）环境中使用。有关更多信息，请参阅 Service Catalog Puppet 文档中的 Using policy simulations 和 Applying service control policies。

成熟度、完整性和支持

虽然 Service Catalog Puppet 不是官方支持的 AWS 服务，但它已被广泛采用。在过去几年中，大型组织一直在使用此工具在所需的预调配时间段内成功地将产品集中预调配到数百个 OU 账户。事实证明，它可以大规模提供容错产品预调配。遇到 Service Catalog Puppet 任何问题的用户都可以在 GitHub 存储库中记录这些问题，以便 AWS Labs 解决方案的贡献者进行解决。

Service Catalog Factory

Service Catalog Factory 是 AWS 实验室提供的另一种工具。它类似于 AWS Control Tower – 它生成账户并调用 Service Catalog（可能通过 Puppet），来在这些账户中预调配 IaP。它使用许多与 Service Catalog Puppet 相同的机制来实现其功能。Service Catalog Factory 可以调用 Service Catalog 或 Service Catalog Puppet 为账户中的产品预调配基础设施。此工具还支持在多个 AWS 区域和组织中生成账户。有关更多信息，请参阅 Service Catalog Factory 文档和 GitHub 存储库。

摘要和后续步骤

Service Catalog 可帮助您快速、可靠地预调配基础设施即产品。您可以根据已定义的产品目录提供自助式基础设施，也可以采用轴辐式模型将产品推送到指定的目标账户。您可以使用 CloudFormation 脚本或使用 AWS CDK 来定义 Service Catalog 产品及其预调配模板。在这两种方法中，Service Catalog 都通过调用 CloudFormation 来部署代表产品预调配模板的堆栈来预调配产品。该堆栈将部署到 CloudFormation 堆栈集中的所有指定的目标账户。

与 CloudFormation 相比，Service Catalog 开发的 AWS CDK 方法支持更高的模块化和重用性，因为您可以使用预定义的 Service Catalog 产品和产品组合类别以及预定义的资源类型，来定义产品及其资源。AWS CDK 实现需要更高级的编程技能。如果您的组织想要建立自己的可重复使用的产品框架，并以标准化的资源配置和行为作为 AWS 基础设施开发的基础，那么这可能是合理的。

您可以使用 Service Catalog Puppet 和 Service Catalog Factory 来增强 Service Catalog 功能，主要用于预调配。Service Catalog Puppet 具有声明性和基于标签的产品预调配规范；内置、可自定义、高性能和专用预调配工作流程；以及内置、可自定义、基于操作的 CI/CD 和 SDLC 管线。通过使用工作流程依赖项管理和内置的测试自动化功能，您可以以更低的运营风险链接 Service Catalog 产品。Service Catalog Puppet 可帮助您在紧迫的时段内可靠地向数百个账户预调配产品。Service Catalog Factory 类似于 AWS Control Tower。它生成账户，并调用 Service Catalog 在这些账户内预调配 IaP。

Service Catalog 和 Service Catalog 工具提供了广泛的功能，可帮助您在 AWS 上管理 IaP。Service Catalog 和这些工具正在不断改进。有关最新功能，请参阅 AWS Service Catalog 功能和 AWS Service Catalog 产品存储库。

资源

参考

- Service Catalog 文档
- Service Catalog API
- AppRegistry
- CloudFormation 文档
- CloudFormation 堆栈集
- AWS::ServiceCatalog::CloudFormationProduct 资源
- 使用 Terraform 启动 AWS Service Catalog 产品
- AWS Cloud Development Kit (AWS CDK)
- Service Catalog 构造库
- AWS CDK ProductStack 类
- AWS Organizations 文档

工具

- Service Catalog Puppet 文档
- Service Catalog Puppet GitHub 存储库
- Service Catalog Factory 文档
- Service Catalog Factory GitHub 存储库

AWS 规范指引模式

- 在多个 AWS 账户和 AWS 区域中管理 AWS Service Catalog 产品
- 在不同的 AWS 账户和 AWS 区域之间复制 AWS Service Catalog 产品
- 使用 AWS CDK 自动部署 AWS Service Catalog 产品组合和产品

文档历史记录

下表介绍了本指南的一些重要更改。如果您希望收到有关未来更新的通知，可以订阅 RSS 源。

变更	说明	日期
初次发布	—	2023 年 1 月 30 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。