aws

开发人员指南

AWS 基础架构编辑器

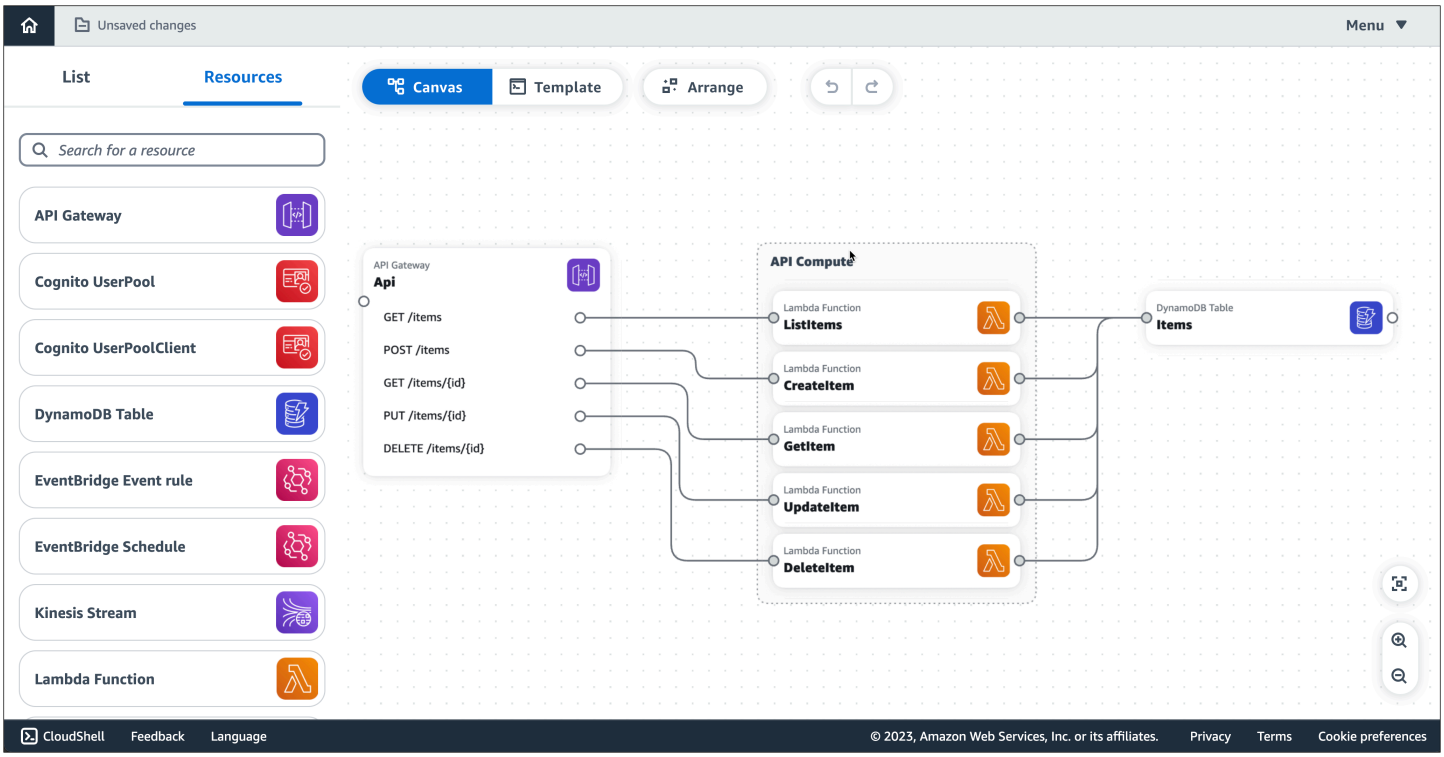
AWS 基础架构编辑器: 开发人员指南

Table of Contents

什么是 AWS 基础架构编辑器？

AWS 基础架构编辑器 允许您在上直观地撰写现代应用程序。 AWS更具体地说，您可以使用 Infrastructure Composer 对所有支持的 AWS 服务进行可视化、构建和部署现代应用程序，而 AWS CloudFormation 无需成为这方面的专家 CloudFormation。

在您构建 AWS CloudFormation 基础架构时，Infrastructure Composer 会通过令人愉悦的 drag-and-drop界面创建您的基础设施即代码 (IaC) 模板，同时遵循 AWS 最佳实践。下图显示了在 Infrastructure Composer 的可视化画布上拖放、配置和连接资源是多么容易。



基础设施编排器可以在基础设施编排控制台和控制 CloudFormation 台模式下使用。 AWS Toolkit for Visual Studio Code

主题

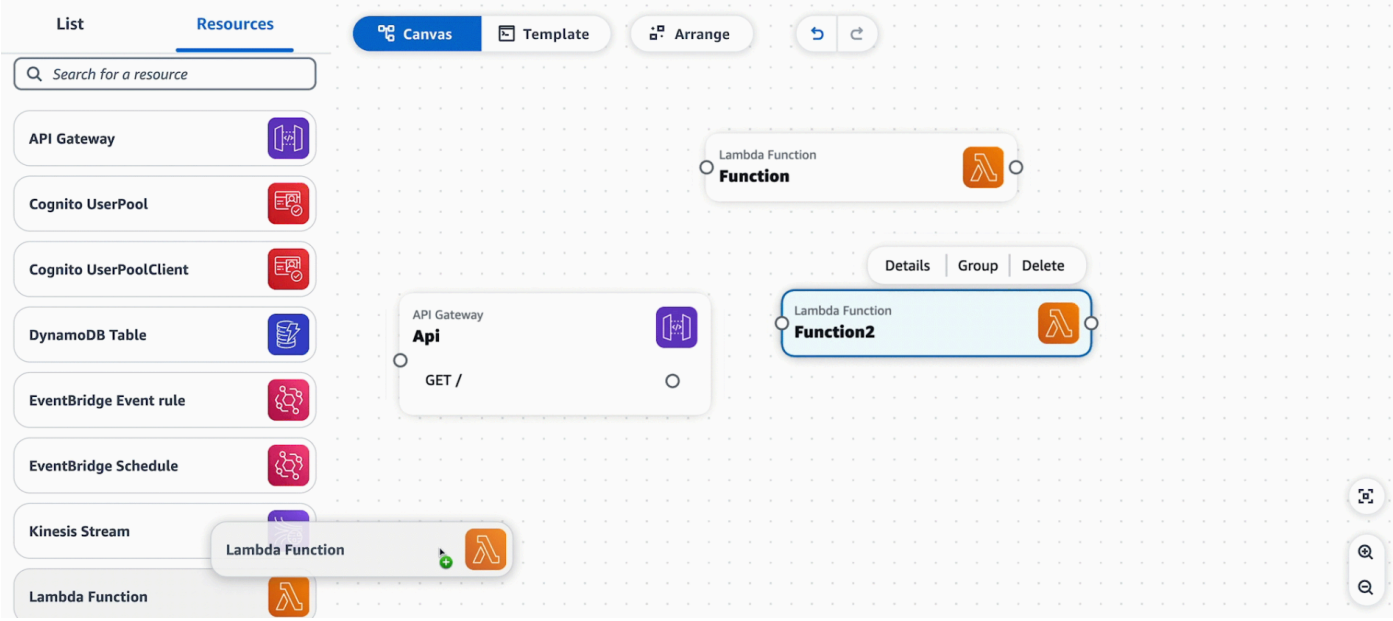
- 构建您的应用程序架构
- 定义您的基础设施即代码 (IaC) 模板
- 与现有工作流程集成
- 访问基础架构编辑器的方法
- 了解详情

- 后续步骤
- 的无服务器概念 AWS 基础架构编辑器

构建您的应用程序架构

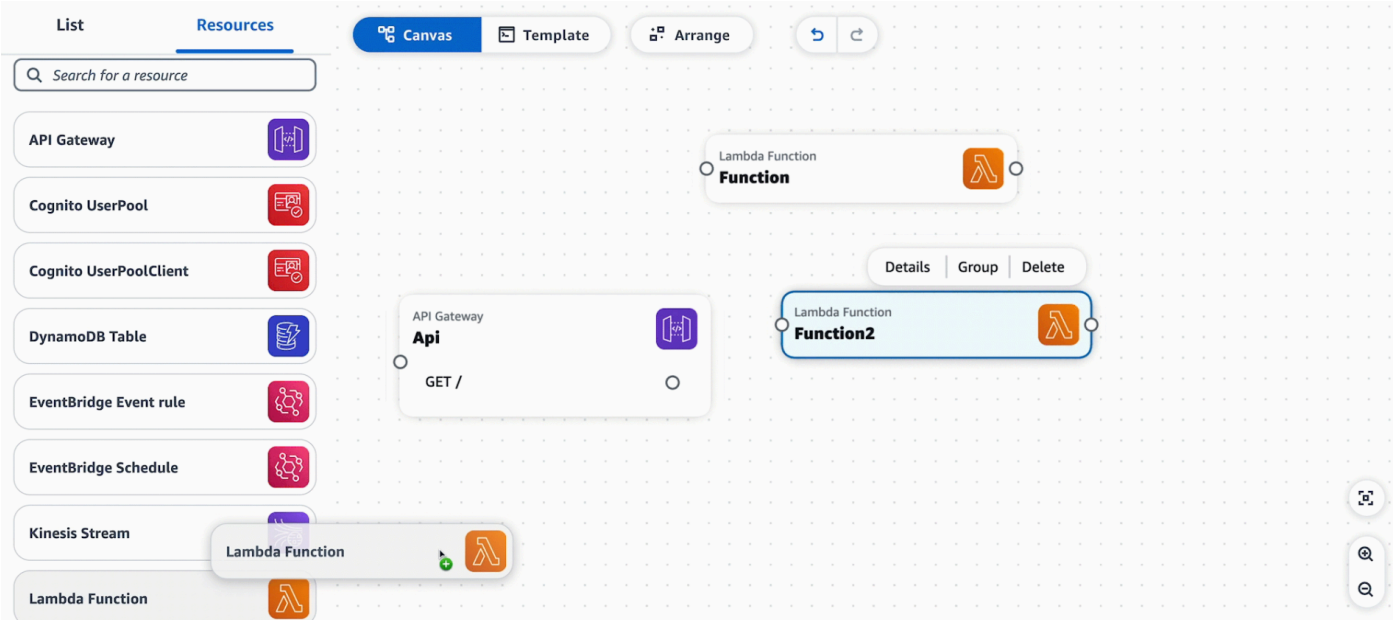
用卡片建造

将卡片放在基础设施编排器画布上，以可视化和构建您的应用程序架构。



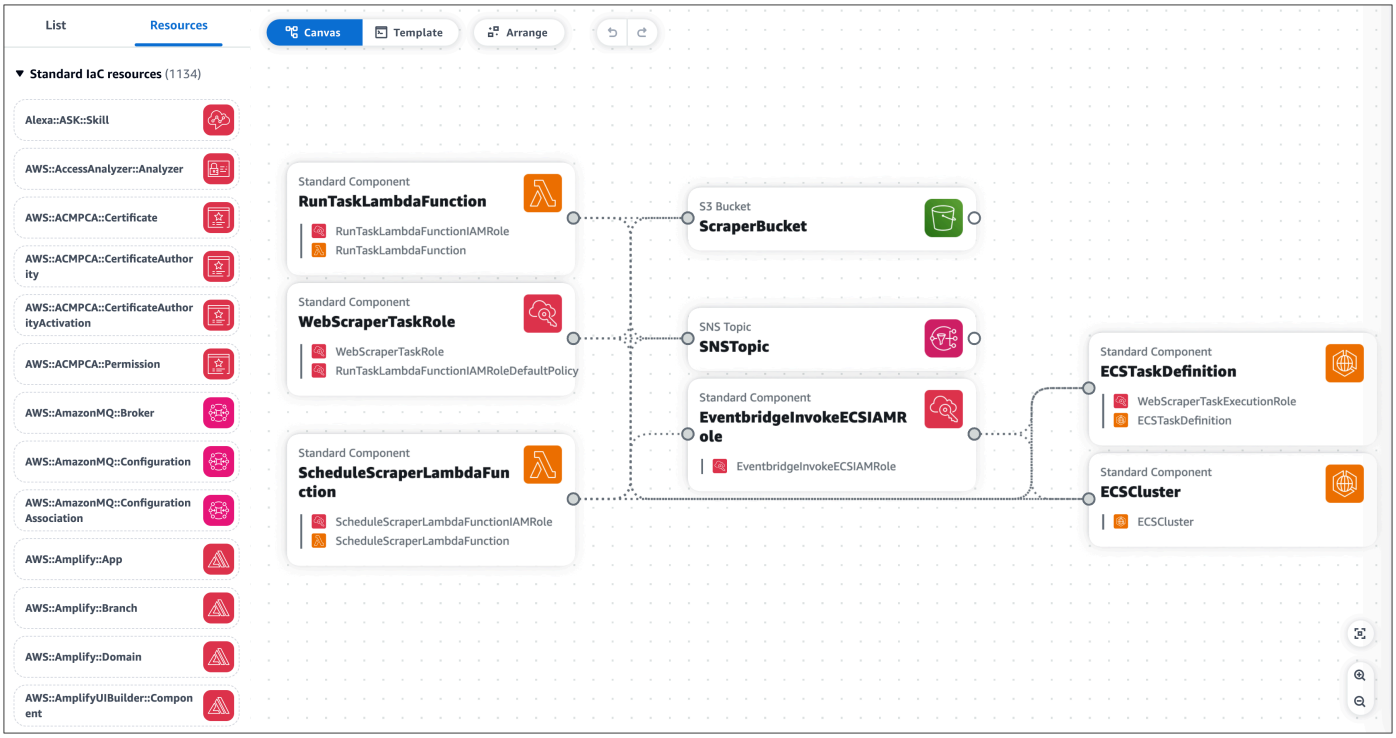
将卡片连接在一起

通过直观地将资源连接在一起，配置它们之间的交互方式。通过精选的属性面板进一步指定其属性。



使用任何 AWS CloudFormation 资源

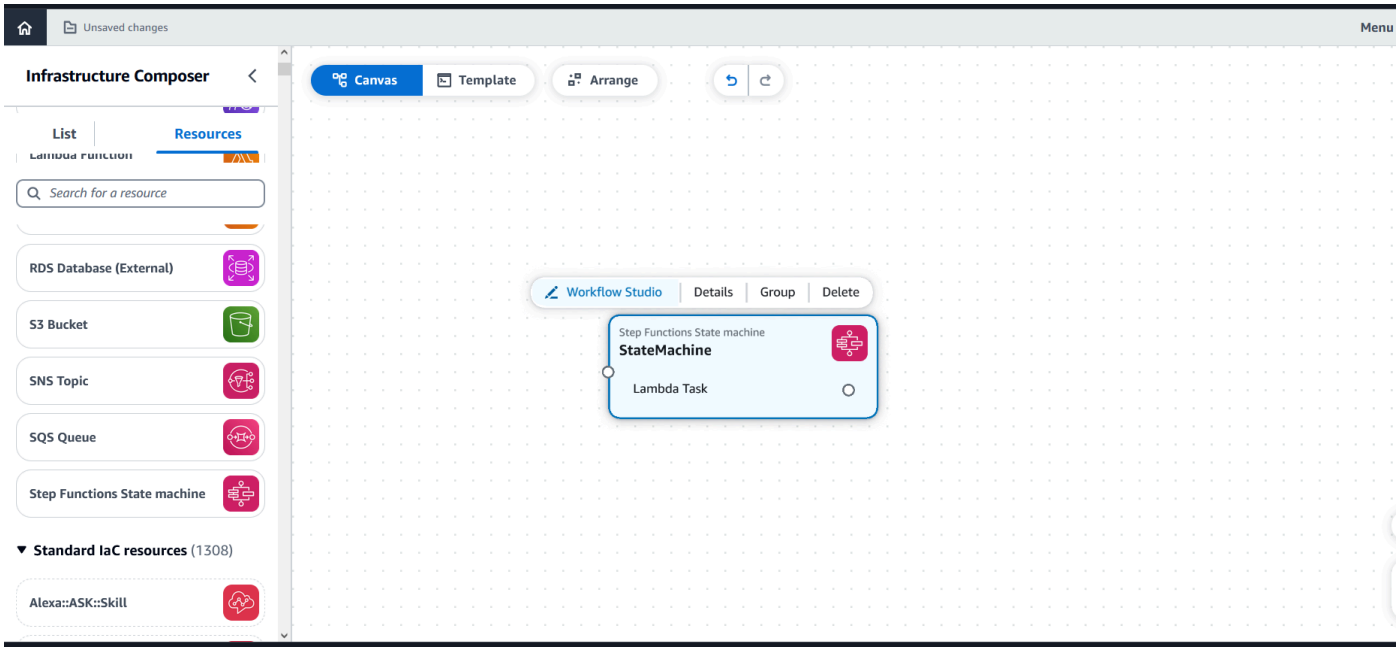
将任何 CloudFormation 资源拖到画布上以构成您的应用程序架构。Infrastructure Composer 提供了一个起始 IaC 模板，您可以使用该模板来指定资源的属性。要了解更多信息，请参阅[在基础设施编排器中配置和修改卡片]。



使用精选功能访问其他功能 AWS 服务

构建应用程序时通常使用或一起配置的基础设施编排器功能 AWS 服务 。要了解更多信息，请参阅与 Amazon VPC 集成。

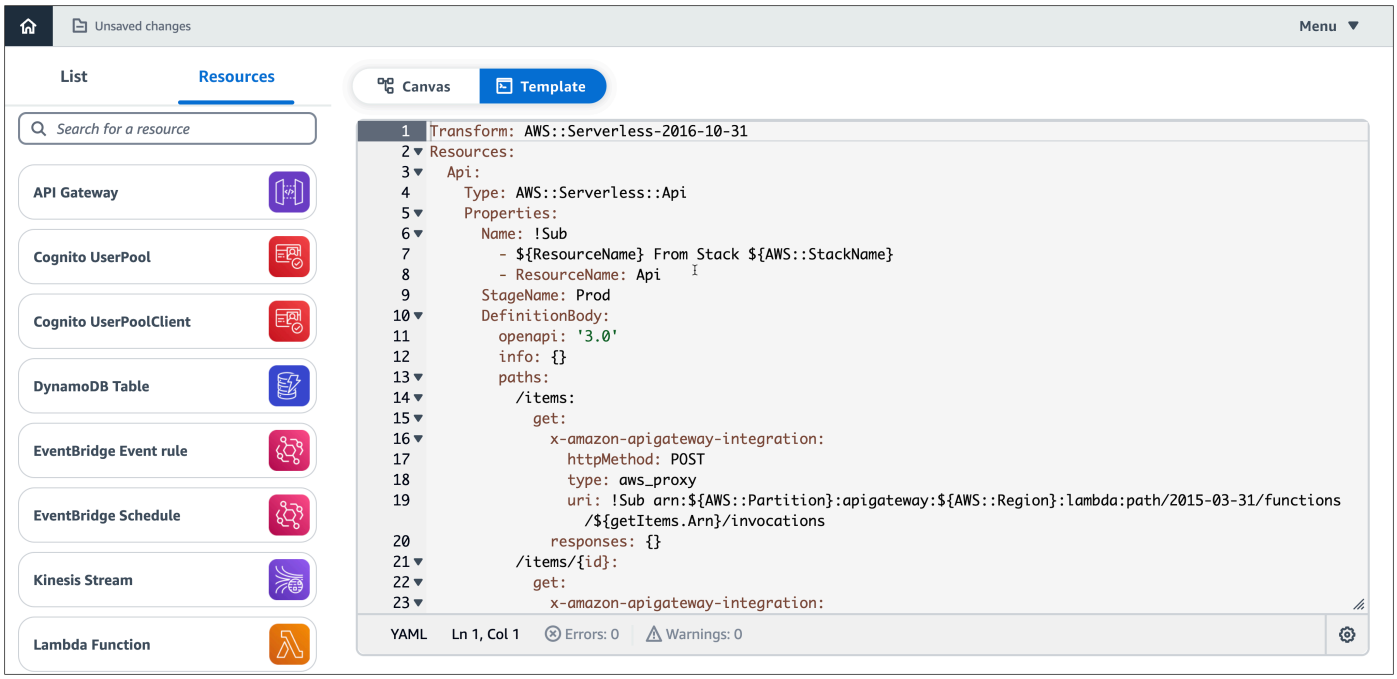
以下是该功能的示例，该 AWS Step Functions 功能提供了Workflow Studio直接在基础架构编辑器画布中启动 Step Functions 的集成。



定义您的基础设施即代码 (IaC) 模板

基础架构编排器创建您的基础架构代码

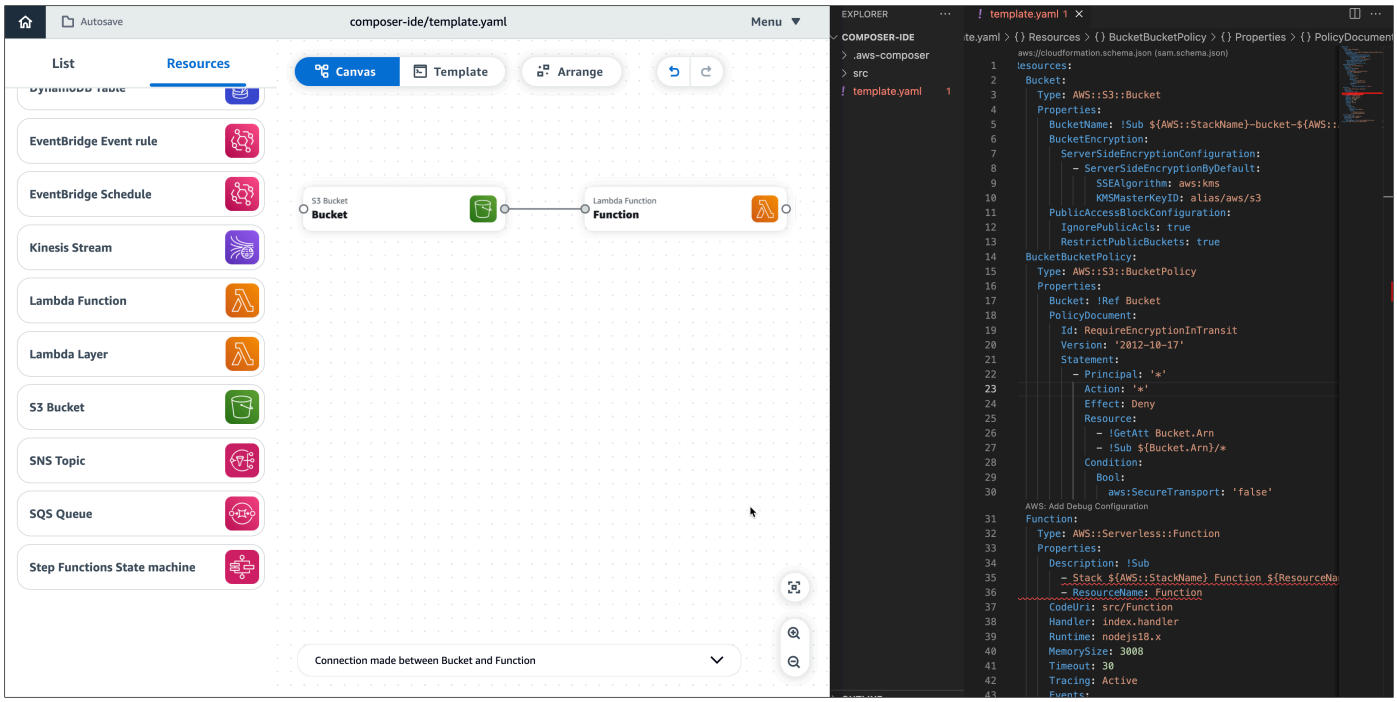
在你撰写时，基础设施编排器会按照 AWS 最佳实践自动创建你的 AWS CloudFormation 和 AWS Serverless Application Model (AWS SAM) 模板。您可以直接在基础架构编排器中查看和修改您的模板。基础架构编排器会自动同步可视化画布和您的模板代码之间的更改。



与现有工作流程集成

导入现有模板和项目

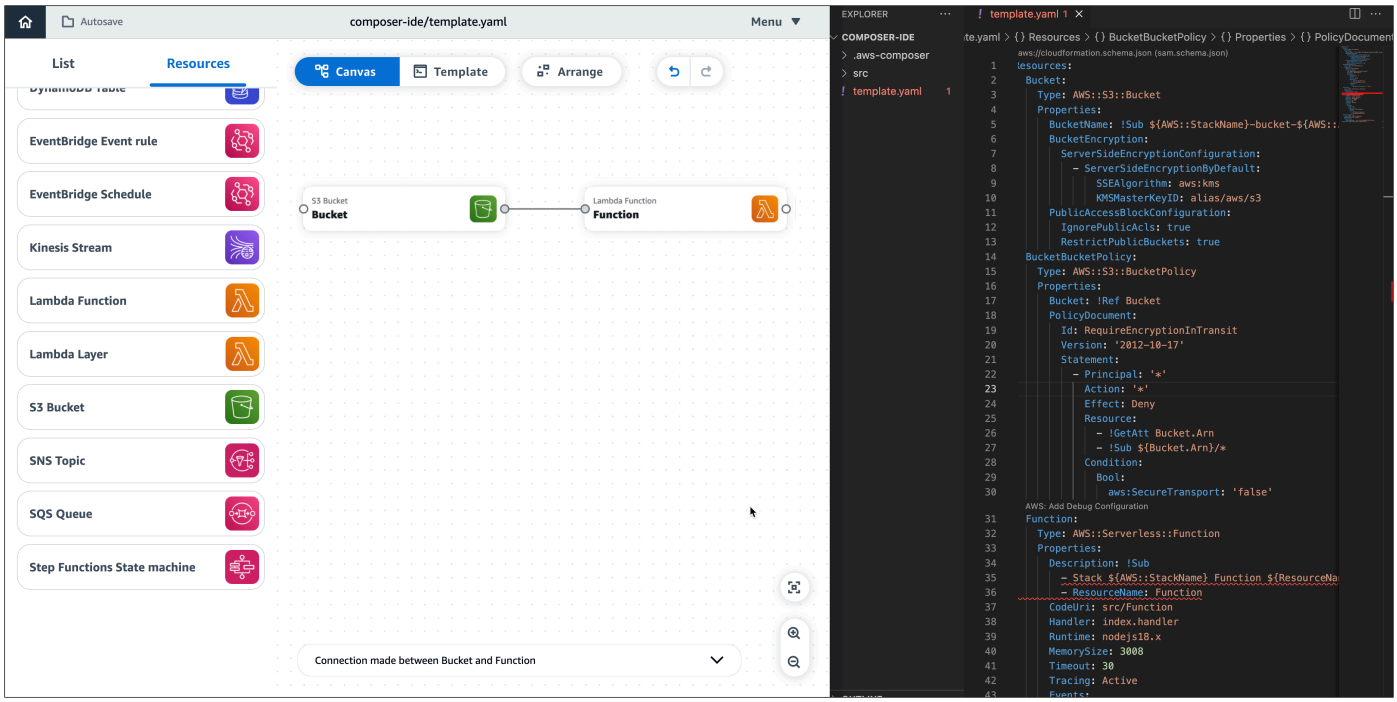
导入现有 AWS SAM 模板 CloudFormation 和模板以对其进行可视化，以便更好地理解和修改其设计。导出您在 Infrastructure Composer 中创建的模板，并将其集成到现有的工作流程中，以便进行部署。



访问基础架构编辑器的方法

从基础设施编排控制台中

通过基础设施编排控制台访问基础架构编排器以快速入门。此外，您还可以使用本地同步模式自动将基础设施编排器与本地计算机同步和保存。



从控制 CloudFormation 台

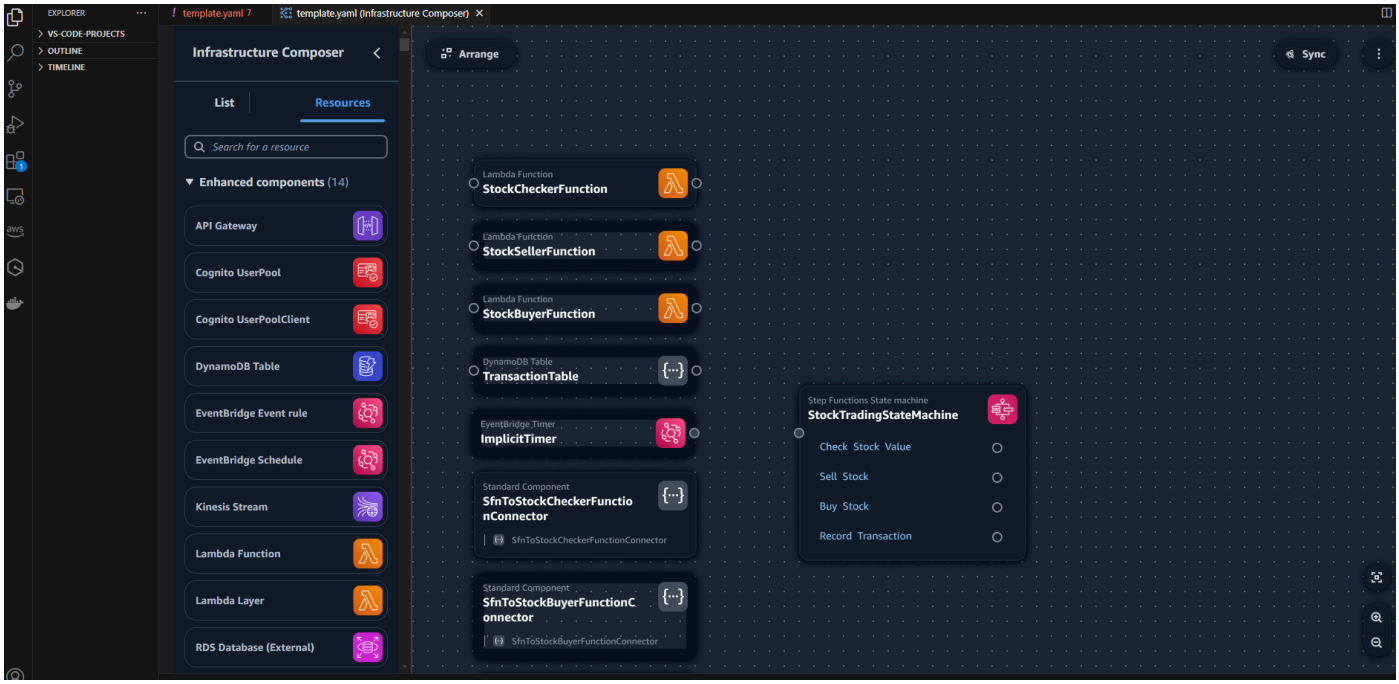
Infrastructure Composer CloudFormation 控制台还支持控制台模式，这是与 CloudFormation 堆栈工作流程集成的 CloudFormation Designer 的一项改进。现在，这个新工具是可视化 CloudFormation 模板的推荐工具。

从 Lambda 控制台

借助基础设施编排器，您还可以从 Lambda 控制台导入 Lambda 函数。要了解更多信息，请参阅从 Lambda 控制台将函数导入基础设施编排器。

来自 AWS Toolkit for Visual Studio Code

通过 Toolkit for VS Code 扩展访问基础架构编排器，将基础设施编排器引入本地开发环境。



了解详情

要继续学习基础架构 Composer，请参阅以下资源：

- 基础架构编辑器卡片
- 以可视化方式撰写和创建无服务器应用程序 | 无服务器办公时间 — Infrastr ucture Composer 概述和演示。

后续步骤

要设置基础架构编排器，请参阅基础架构编排控制台入门。

的无服务器概念 AWS 基础架构编辑器

在使用 AWS 基础架构编辑器之前，请先了解基本的无服务器概念。

无服务器概念

事件驱动型架构

无服务器应用程序由单独的服务组成，例如 AWS Lambda 用于计算的 AWS 服务和用于数据库管理的 Amazon DynamoDB，每个服务都扮演一个专门的角色。然后，这些服务通过事件驱动型架构相互松散地集成。要了解有关事件驱动型架构的更多信息，请参阅什么是事件驱动型架构？。

基础设施即代码（IaC）

基础设施即代码（IaC）是一种以开发人员对待代码的方式对待基础设施的方法，将与应用程序代码开发相同的严密性应用于基础设施配置。您可以在模板文件中定义基础架构，将其部署到模板文件中 AWS，然后为您 AWS 创建资源。使用 IAC，您可以在代码中定义 AWS 要配置的内容。有关更多信息，请参阅 AWS AWS 白皮书简介中的基础架构即代码。 DevOps

无服务器技术

借助 AWS 无服务器技术，您无需管理自己的服务器即可构建和运行应用程序。所有服务器管理都是通过完成的 AWS，它提供了许多好处，例如自动扩展和内置的高可用性，使您可以将自己的想法快速付诸实践。使用无服务器技术，您可以专注于产品的核心，而不必担心服务器的管理和操作。要了解有关无服务器的更多信息，请参阅上的 Ser verless。 AWS

有关核心 AWS 无服务器服务的基本介绍，请参阅 Serv erless 101：了解无服务器领域的无服务器服务。

基础架构编辑器卡片

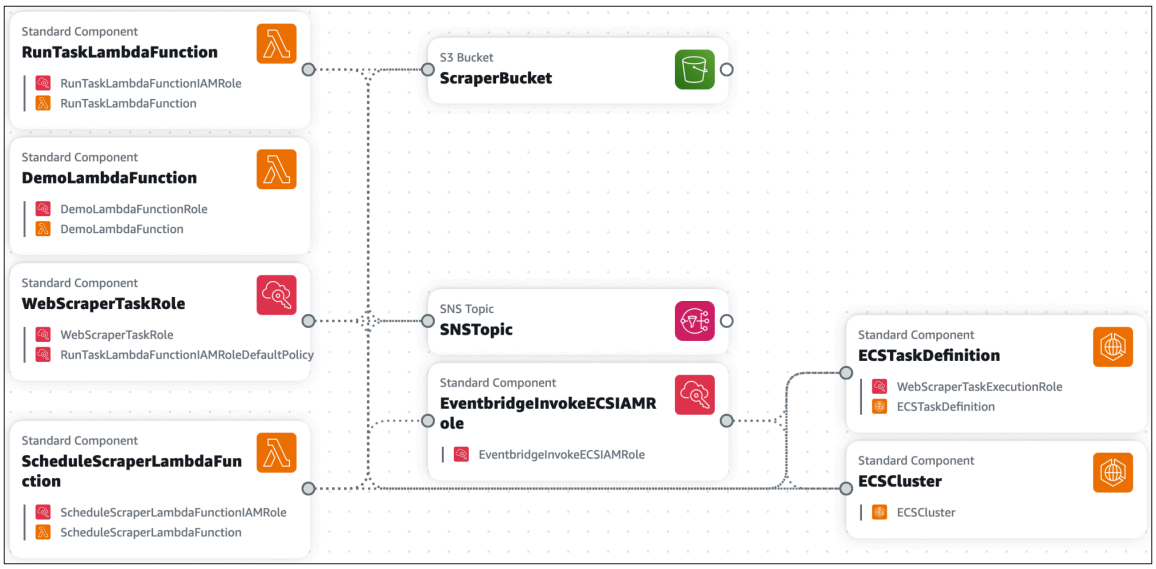
基础设施编排器简化了为 CloudFormation 资源编写基础架构即代码 (IaC) 的过程。要有效地使用基础架构 Composer，首先需要了解两个基本概念：基础设施编排卡和卡片连接。

在 Infrastructure Composer 中，卡片代表 CloudFormation 资源。卡片一般分为两类：

- 增强型组件卡 — 一组 CloudFormation 资源已组合成单个精选卡片，可增强易用性和功能性，专为各种用例而设计。增强型组件卡是基础架构编排器资源选项板中列出的第一张卡片。
- 标准 IaC 资源卡-单一 AWS CloudFormation 资源。每张标准 IaC 资源卡一旦拖到画布上，就会被标记为标准组件，可以组合成多个资源。

> **Note**
>
> 根据卡的不同，标准 IaC 资源卡在被拖到视觉画布上后可能会被标记为标准组件卡。这只是意味着该卡是一张或多张标准 IaC 资源卡的集合。

虽然资源面板中提供了某些类型的卡片，但当您将现有 CloudFormation 或 AWS Serverless Application Model (AWS SAM) 模板导入基础架构编辑器时，卡片也可能出现在画布上。下图是包含各种卡片类型的导入应用程序的示例：



主题

- 基础架构编排器中的增强组件卡

- 基础架构编辑器中的标准组件卡
- 基础架构编辑器中的卡片连接

基础架构编排器中的增强组件卡

增强的组件卡由基础架构编排器创建和管理。每张卡片都包含在上面构建应用程序时通常一起使用的 CloudFormation 资源 AWS。他们的基础架构代码是由基础设施编排器按照 AWS 最佳实践创建的。增强型组件卡是开始设计应用程序的好方法。

增强组件卡可在资源面板的 “增强组件” 部分下找到。

增强型组件卡可以在基础设施编排器中进行全面配置和使用，以设计和构建您的无服务器应用程序。我们建议您在不使用现有代码的情况下设计应用程序时使用增强型组件卡。

下表显示了我们的增强组件，其中包含指向该卡片特色资源的 AWS CloudFormation 或 AWS Serverless Application Model (AWS SAM) 模板规范的链接：

卡片	参考
Amazon API Gateway	AWS::Serverless:: API
亚马逊 Cognito UserPool	AWS::Cognito::UserPool
亚马逊 Cognito UserPoolClient	AWS::Cognito::UserPool客户端
亚马逊 DynamoDB 表	AWS::DynamoDB::Table
亚马逊 EventBridge 活动规则	AWS::Events::Rule
EventBridge 日程安排	AWS::Scheduler::Schedule
亚马逊 Kinesis Stream	AWS::Kinesis::Stream
AWS Lambda 函数	AWS::Serverless::Function
Lambda 层	AWS::Serverless::LayerVersion
亚马逊简单存储服务 (Amazon S3) Semple Service Bucket	AWS::S3::Bucket

卡片	参考
亚马逊简单通知服务 (Amazon SNS) Semple Notification	AWS::SNS::Topic
亚马逊简单队列服务 (Amazon SQS) Simple Queue	AWS::SQS::Queue
AWS Step Functions 状态机	AWS::Serverless::StateMachine

示例

以下是 S 3 存储桶增强型组件的示例：

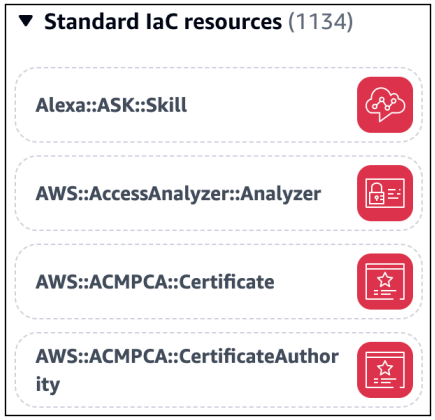
当您将一个 S3 Bucket 组件卡片拖到画布上并查看您的模板时，您将看到以下两个 CloudFormation 资源已添加到您的模板中：

- `AWS::S3::Bucket`
- `AWS::S3::BucketPolicy`

S3 Bucket 增强版组件卡代表了两个 CloudFormation 资源，这两个资源都是亚马逊简单存储服务 (Amazon S3) 存储段与应用程序中的其他服务交互所必需的。

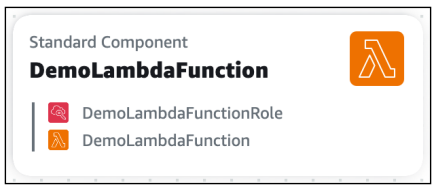
基础架构编辑器中的标准组件卡

在将标准组件卡放在基础设施编排器的可视画布上之前，它会作为标准 (IaC) 资源卡列在基础设施编排器的资源选项板上。标准 (IaC) 资源卡代表单一 CloudFormation 资源。每张标准 IaC 资源卡片一旦放置在视觉画布上，就会变成一张标有 “标准组件” 的卡片，并且可以组合起来表示多个 CloudFormation 资源。

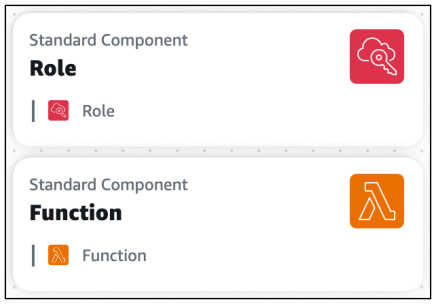


每个标准 IaC 资源卡均可通过其 CloudFormation 资源类型进行识别。以下是代表资源类型的标准 IaC 资源卡的示例：`AWS::ECS::Cluster` CloudFormation

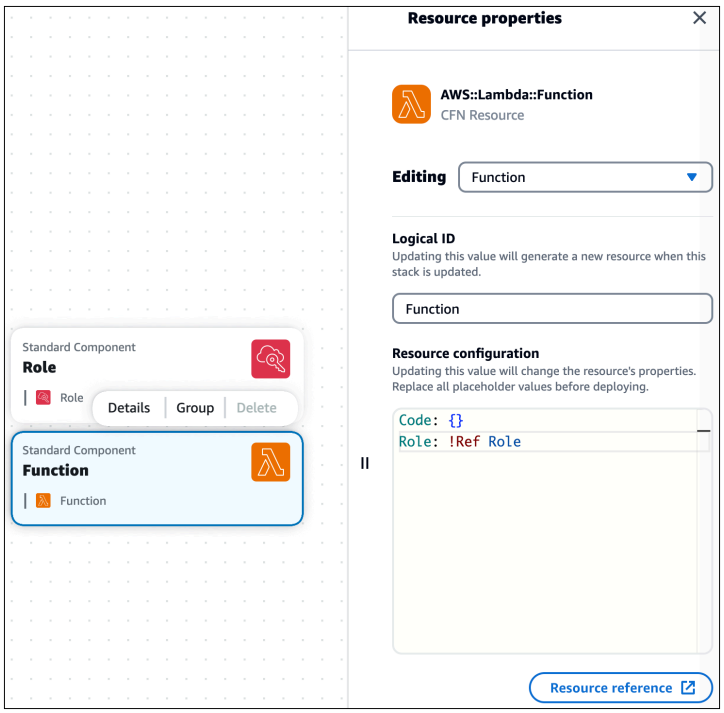
每个标准组件卡片都可视化其包含的 CloudFormation 资源。以下是包含两个标准 IaC 资源的标准组件卡示例：



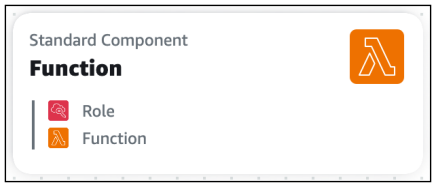
在配置标准组件卡的属性时，Infrastructure Composer 可能会将相关的卡片组合在一起。例如，以下是两张标准组件卡：



在代表资源的标准组件卡片的**`AWS::Lambda::Function`**资源属性面板中，我们通过其逻辑 ID 来引用 AWS Identity and Access Management (IAM) 角色：



保存模板后，两个标准组件卡片合并为一个标准组件卡。



基础架构编辑器中的卡片连接

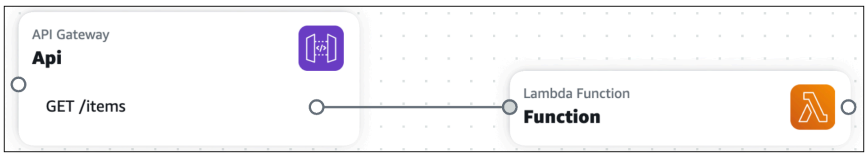
在中 AWS 基础架构编辑器，两张卡片之间的连接用一条线直观地显示。这些行表示应用程序中事件驱动的关系。

主题

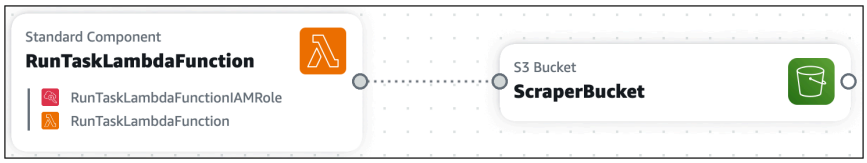
- 卡片之间的连接
- 增强型组件卡之间的连接
- 与标准 IaC 资源卡的连接和连接

卡片之间的连接

将卡片连接在一起的方式因卡片类型而异。每张增强型卡都至少有一个连接器端口。要连接它们，您只需选择一个连接器端口并将其拖到另一张卡的端口，Infrastructure Composer 就会连接这两个资源或显示一条消息，说明不支持此配置。



如上所示，增强型组件卡之间的线条是实线的。相反，标准 IaC 资源卡（也称为标准组件卡）没有连接器端口。对于这些卡片，您必须在应用程序的模板中指定这些事件驱动的关系，Infrastructure Composer 将自动检测它们的连接，并在卡片之间用虚线将它们可视化。

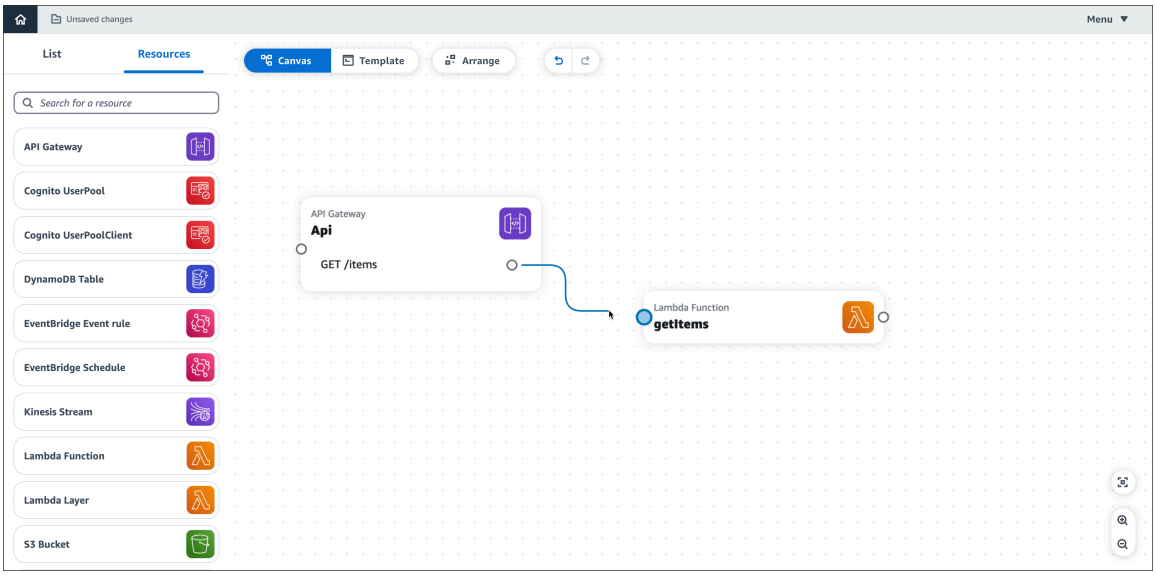


要了解更多信息，请参阅以下部分。

增强型组件卡之间的连接

在 Infrastructure Composer 中，两张增强型组件卡之间的连接以实线直观显示。这些行表示应用程序中事件驱动的关系。

要连接两张卡，请单击一张卡上的端口，然后将其拖到另一张卡上的端口上。



> **Note**
>
> 标准 IaC 资源卡没有连接器端口。对于这些卡片，您必须在应用程序的模板中指定它们的事件驱动关系，Infrastructure Composer 将自动检测它们的连接，并在卡片之间用虚线将它们可视化。

有关更多信息，请参阅 在基础设施编辑器的可视化画布上连接卡片。

哪些增强的组件卡规定

两张卡之间的连接（用一条直线直观表示），必要时可提供以下内容：

- AWS Identity and Access Management (IAM) 策略
- 环境变量
- Events

IAM 策略

当资源需要调用其他资源的权限时，Infrastructure Composer 会使用 AWS Serverless Application Model (AWS SAM) 策略模板来配置基于资源的策略。

- 要了解有关 IAM 权限和策略的更多信息，请参阅 I AM 用户指南中的访问管理概览：权限和策略。
- 要了解有关 AWS SAM 策略模板的更多信息，请参阅《AWS Serverless Application Model 开发人员指南》中的AWS SAM 策略模板。

环境变量

环境变量是临时值，可以对其进行更改以影响资源的行为。必要时，Infrastructure Composer 会定义基础设施代码，以便在资源之间使用环境变量。

Events

资源可以通过不同类型的事件调用其他资源。必要时，Infrastructure Composer 会定义资源通过事件类型进行交互所需的基础设施代码。

与标准 IaC 资源卡的连接和连接

所有 CloudFormation 资源都可用作 “资源” 选项板中的标准 IaC 资源卡。当您将标准 IaC 资源卡拖到画布上时，标准 IaC 资源卡就会变成标准组件卡，这会提示基础架构编排器在应用程序中为您的资源创建起始模板。

有关更多信息，请参阅 基础架构编辑器中的标准卡。

基础架构编排控制台入门

使用本节中的主题来设置 AWS 基础架构编辑器 和学习如何使用其视觉画布设计应用程序。本节中的教程和教程显示在 Infrastructure Composer 控制台中，这是默认的用户体验。本节中的主题向您展示如何完成使用基础设施编排器的先决条件、如何使用基础设施编排控制台、加载和修改项目以及如何构建第一个应用程序。

基础架构编排器也可在控制台模式 AWS Toolkit for Visual Studio Code 和 CloudFormation 控制台模式下使用。工具之间的体验通常是相同的，但每种工具之间有一些差异。有关在这些工具中使用基础设施编排器的详细信息，请参阅你可以在哪里使用基础设施编排器。

主题

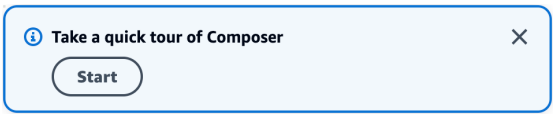
- 浏览基础架构编排控制台
- 加载和修改基础设施编排器演示项目
- 使用基础架构编排器构建您的第一个应用程序

浏览基础架构编排控制台

要大致了解 AWS 基础架构编辑器 工作原理，请浏览基础架构 Composer 控制台中内置的内容。有关基础设施编排控制台的概述，请参阅浏览基础架构编排控制台。有关使用基础架构编排器的深入指导，请参阅如何作曲 AWS 基础架构编辑器。

参观《基础设施编译器》

1. 登录基础设施编排控制台。
2. 在主页上，选择打开演示。
3. 在右上角的 “快速浏览 Composer” 窗口中，选择 “开始”。



4. 在 Composer 浏览窗口中，执行以下操作：
   - 要进入下一步，请选择 “下一步”。
   - 要返回上一个步骤，请选择 “上一步”。
   - 在最后一步中，要完成游览，请选择 “结束”。

本教程简要概述了 Infrastructure Composer 的基本功能，例如使用、配置和连接卡。有关更多信息，请参阅如何作曲 AWS 基础架构编辑器。

后续步骤

要在基础设施编排器中加载和修改项目，请参阅加载和修改基础设施编排器演示项目。

加载和修改基础设施编排器演示项目

使用本教程熟悉基础设施编排器的用户界面，学习如何加载、修改和保存基础设施编排演示项目。

本教程是在基础设施编排控制台中完成的。完成后，你就可以开始了使用基础架构编排器构建您的第一个应用程序。

主题

第 1 步：打开演示

通过创建演示项目开始使用基础架构编排器。

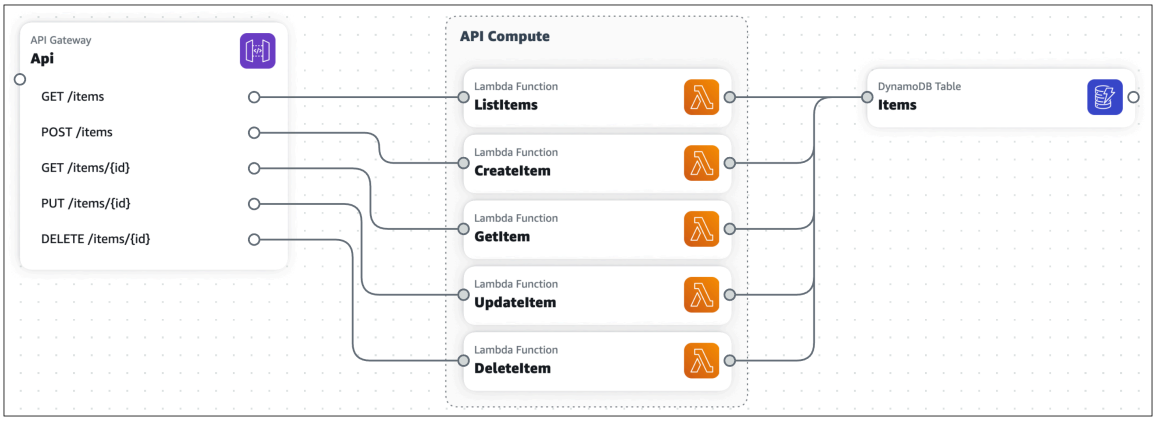
创建演示项目

1. 登录基础设施编排控制台。
2. 在主页上，选择打开演示。

演示应用程序是一个基本的创建、读取、删除和更新 (CRUD) 无服务器应用程序，包括：

- 包含五条路线的 Amazon API Gateway 资源。
- 五个 AWS Lambda 功能。
- Amazon DynamoDB 表。

下图为演示：

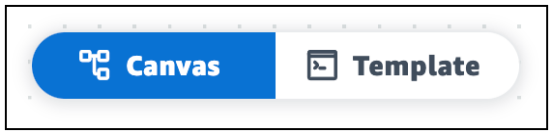


第 2 步：探索基础设施编排器的可视化画布

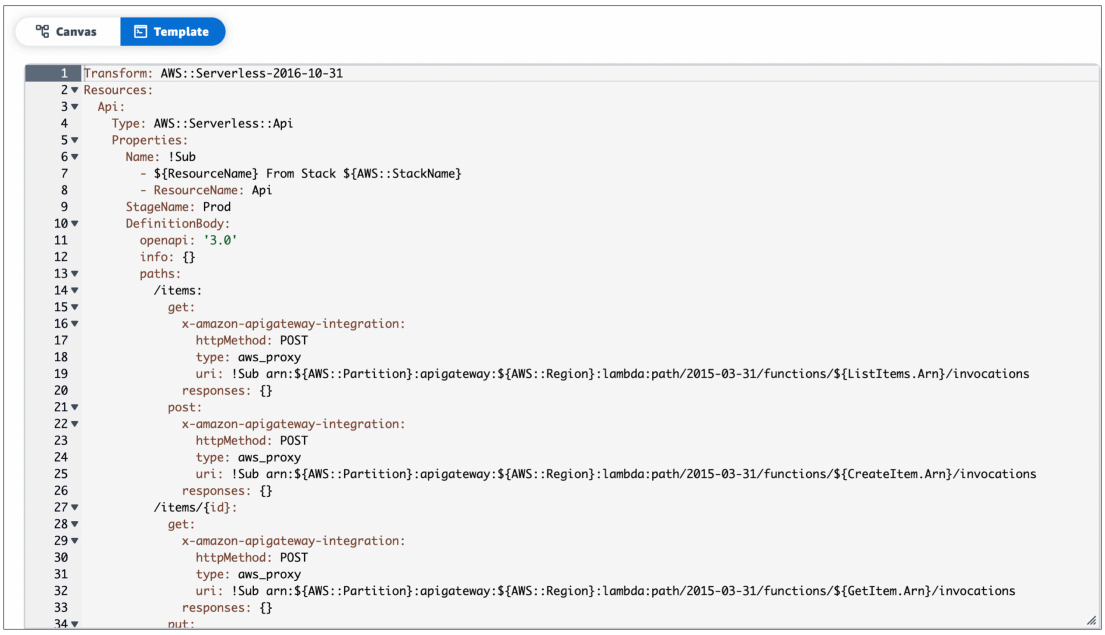
了解可视化画布的功能，构建您的基础架构编排演示项目。有关视觉画布布局的概述，请参阅视觉概述。

探索视觉画布的特点

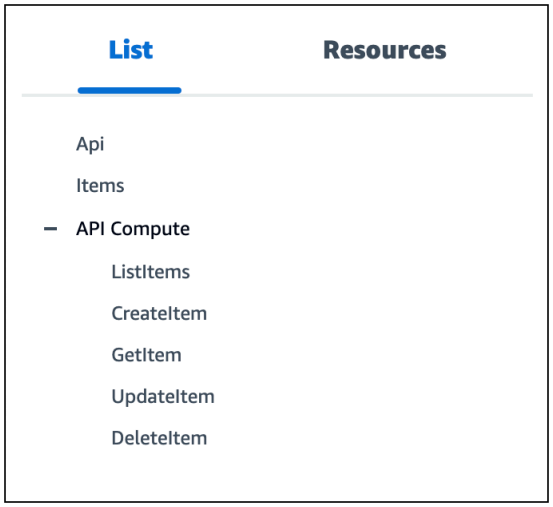
1. 当您打开新的或现有的应用程序项目时，Infrastructure Composer 会加载画布视图，如主视图区域上方所示。

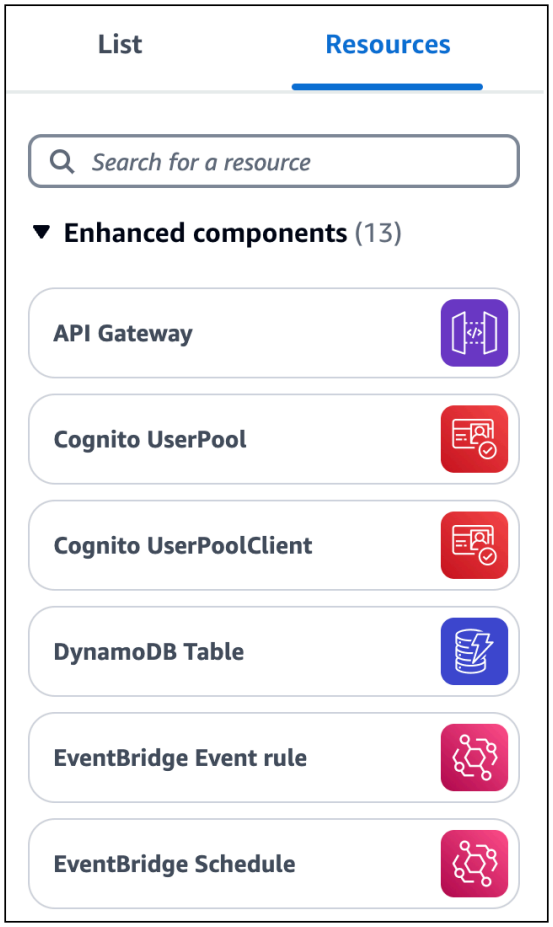
   要在主视图区域显示应用程序的基础架构代码，请选择模板。例如，这是基础设施编排器演示项目的 AWS Serverless Application Model (AWS SAM) 模板视图。



2. 要再次显示应用程序的画布视图，请选择 “画布”。
3. 要以树状视图显示应用程序的资源，请选择列表。



4. 要显示资源选项板，请选择资源。此调色板包含可用于扩展应用程序架构的卡片。您可以搜索卡片或滚动浏览列表。



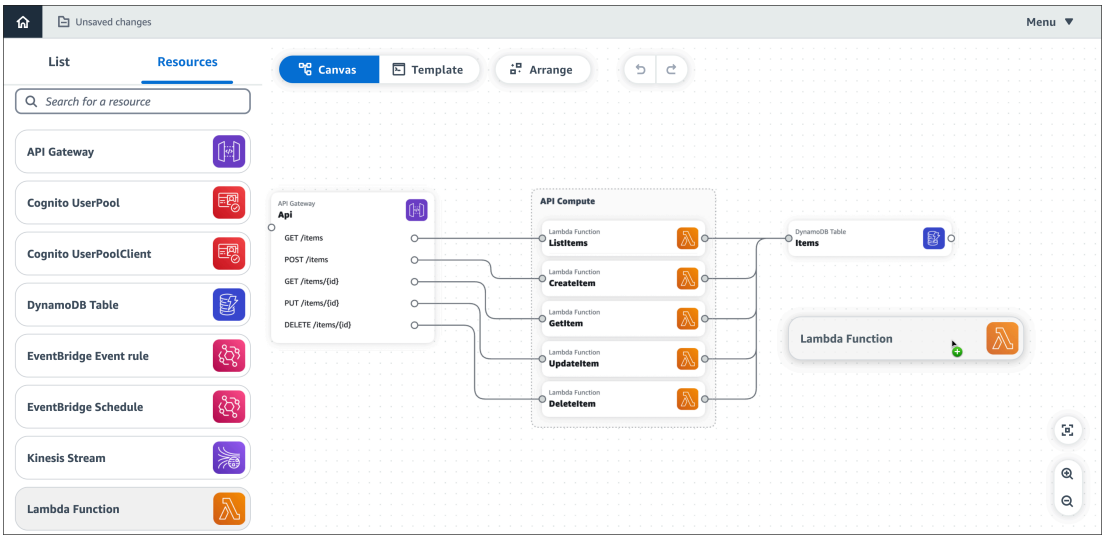
5. 要在视觉画布上移动，请使用基本手势。有关更多信息，请参阅 将卡片放在画布上。

第 3 步：扩展您的应用程序架构

在本步骤中，您将通过向 DynamoDB 表中添加 Lambda 函数来扩展您的应用程序架构。

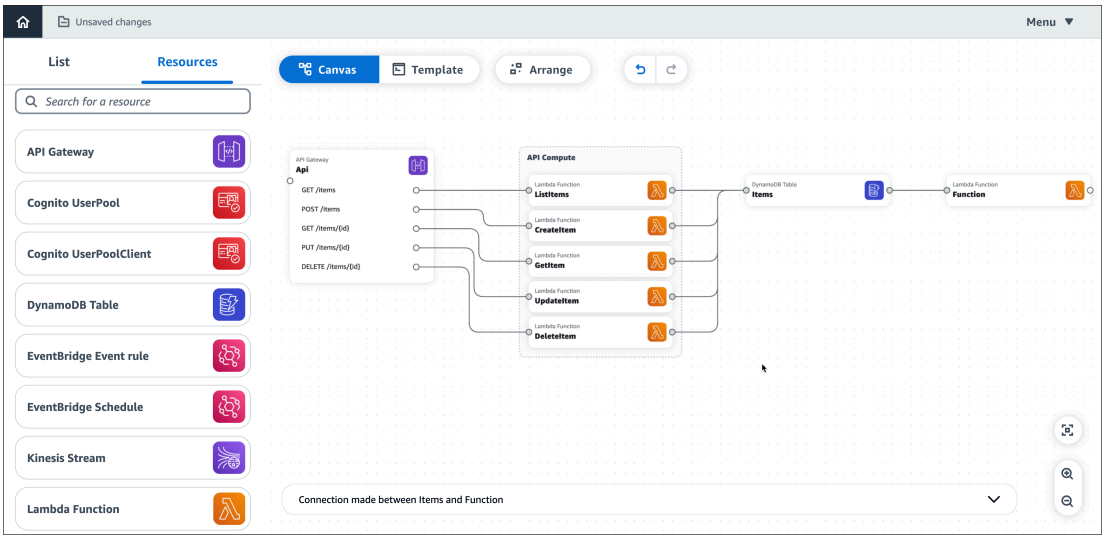
向 DynamoDB 表中添加 Lambda 函数

1. 从资源选项板（资源）中，将 Lambda 函数增强型组件卡片拖到 D ynamoDB 表格卡片右侧的画布上。



2. 将 DynamoDB 表连接到 Lambda 函数。要连接它们，请单击 D ynamoDB 表格卡的右侧端口，然后将其拖到 Lambda 功能卡的左侧端口上。

3. 选择 “排列” 以在画布视图中整理卡片。



4. 配置您的 Lambda 函数。要对其进行配置，请执行以下任一操作：

   - 在画布视图中，在资源属性面板上修改函数的属性。要打开面板，请双击 Lambda 函数卡片。或者，选择卡片，然后选择 “详细信息”。有关资源属性面板中列出的可配置 Lambda 函数属性的更多信息，请参阅AWS Lambda 开发人员指南。
   - 在模板视图中，修改您的函数 (`AWS::Serverless::Function`) 的代码。基础架构编辑器会自动将您的更改同步到画布。有关 AWS SAM 模板中函数资源的更多信息，请参阅AWS SAM 资源和属性参考AWS::Serverless::Function中的。

第 4 步：保存您的应用程序

通过手动将应用程序模板保存到本地计算机或激活本地同步来保存应用程序。

手动保存您的应用程序模板

1. 从菜单中选择 “保存” > “保存模板文件”。
2. 为您的模板提供一个名称，然后在本地计算机上选择一个位置来保存您的模板。按保存。

有关激活本地同步的说明，请参阅在基础设施编排控制台中本地同步并保存您的项目。

后续步骤

要开始构建您的第一个应用程序，请参阅使用基础架构编排器构建您的第一个应用程序。

使用基础架构编排器构建您的第一个应用程序

在本教程中，您将使用 AWS 基础架构编辑器 创建用于管理数据库中用户的创建、读取、更新和删除 (CRUD) 无服务器应用程序。

在本教程中，我们在中使用基础设施编排器 AWS 管理控制台。我们建议您使用Google Chrome或 Microsoft Edge，并使用全屏浏览器窗口。

> 您是无服务器技术的新手吗？
>
> 我们建议您初步了解以下主题：
>
> - 事件驱动型架构
> - 基础设施即代码（IaC）
> - 无服务器技术
>
> 要了解更多信息，请参阅的无服务器概念 AWS 基础架构编辑器。

主题

- 资源属性参考
- 第 1 步：创建您的项目

资源属性参考

在构建应用程序时，请使用此表作为参考，配置您的 Amazon API Gateway 和 AWS Lambda 资源的属性。

方法	路径	函数名称
GET	/items	GetItems
GET	/items/ {id}	getItem
PUT	/items/ {id}	更新项目
POST	/item	AddIte
DELETE	/items/ {id}	deleteItem

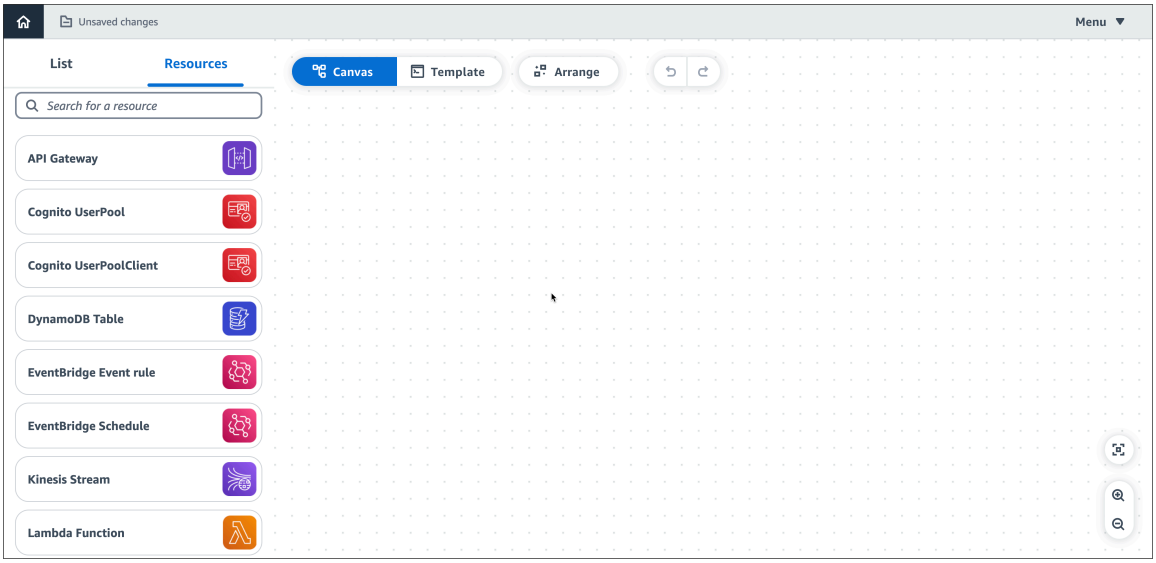
第 1 步：创建您的项目

要开始使用 CRUD 无服务器应用程序，请在 Infrastructure Composer 中创建一个新项目并激活本地同步。

创建新的空白项目

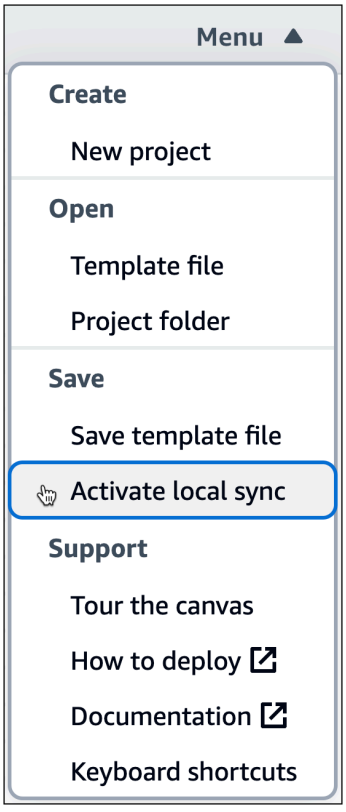
1. 登录基础设施编排控制台。
2. 在主页上，选择创建项目。

如下图所示，Infrastructure Composer 打开可视化画布并加载起始（空白）应用程序模板。



激活本地同步

1. 从 “基础设施编排” 菜单中，选择 “保存” > “激活本地同步”。



2. 在 “项目位置” 中，按选择文件夹，然后选择一个目录。在您设计时，基础设施编排器将在这里保存和同步您的模板文件和文件夹。

项目位置不得包含现有的应用程序模板。

> **Note**
>
> 本地同步需要支持文件系统访问 API 的浏览器。有关更多信息，请参阅 数据基础架构编排器可以访问。

3. 当系统提示允许访问时，选择 “查看文件”。
4. 按 “激活” 开启本地同步。当系统提示保存更改时，选择保存更改。

   激活后，自动保存指示器将显示在画布的左上角区域。

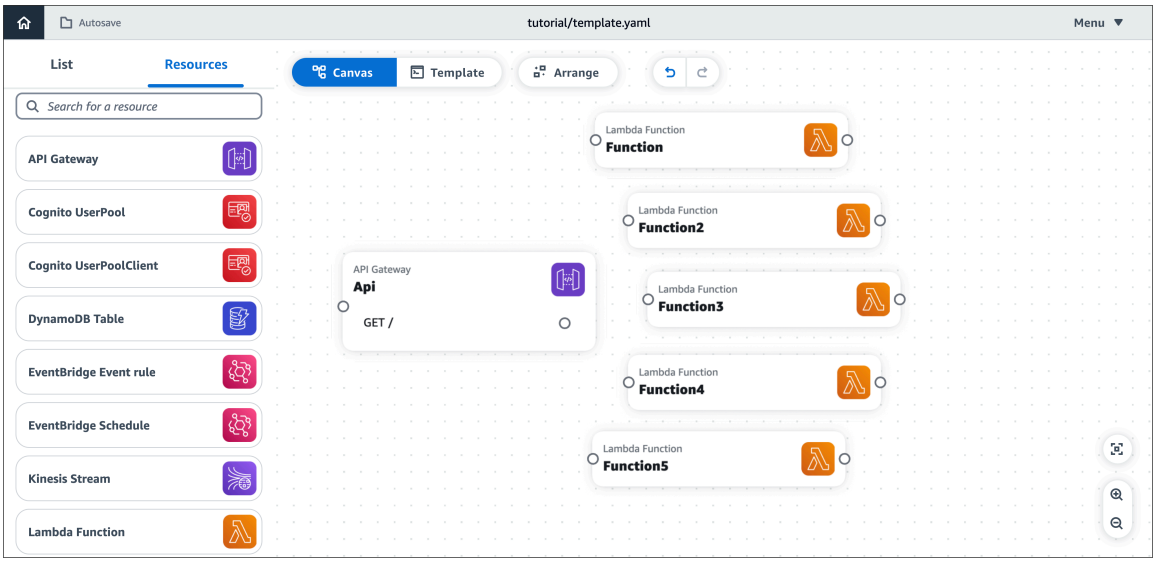
第 2 步：向画布添加卡片

开始使用增强的组件卡设计您的应用程序架构，首先是一个 API Gateway REST API 和五个 Lambda 函数。

将 API Gateway 和 Lambda 卡片添加到画布中

在 “资源” 选项板的 “增强组件” 部分下，执行以下操作：

1. 将 API Gateway 卡片拖到画布上。
2. 将 Lambda 函数卡片拖到画布上。重复此操作，直到在画布上添加了五张 Lambda 函数卡。



第 3 步：配置你的 API Gateway REST API

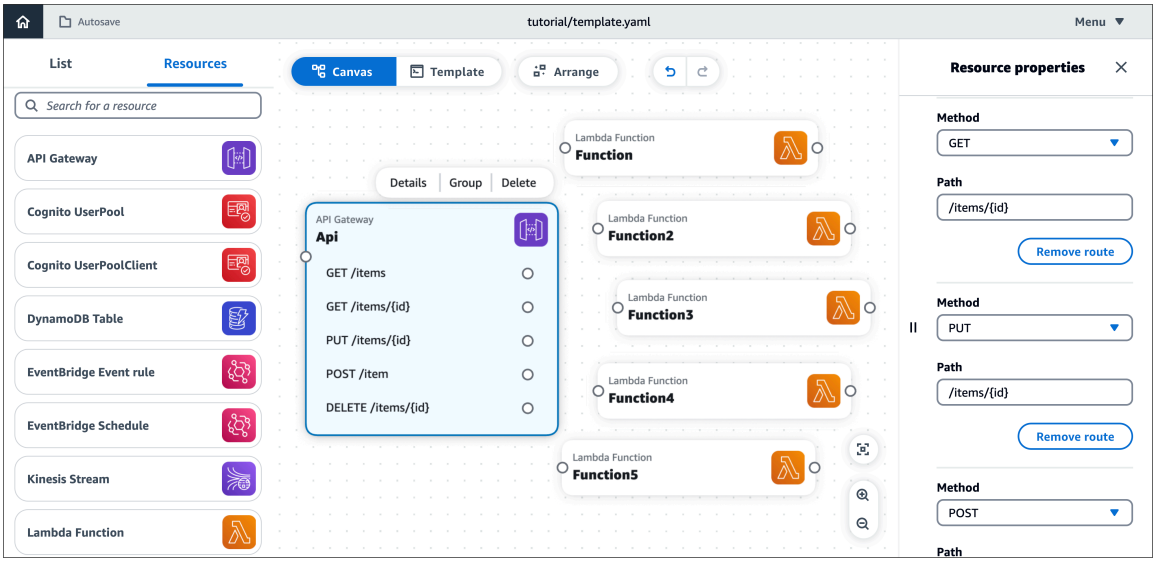
接下来，在您的 API Gateway 卡片中添加五条路由。

向 API Gateway 卡片添加路由

1. 打开 API Gateway 卡片的资源属性面板。要打开面板，请双击卡片。或者，选择卡片，然后选择“详细信息”。
2. 在“资源属性”面板的“路由”下，执行以下操作：

> **Note**
>
> 对于以下每条路由，请使用资源属性参考表中指定的 HTTP 方法和路径值。

   a. 在“方法”中，选择指定的 HTTP 方法。例如，GET。
   b. 在路径中，输入指定的路径。例如 **`/items`**。
   c. 选择 Add route（添加路由）。
   d. 重复前面的步骤，直到添加完所有五条指定路线。
3. 选择保存。

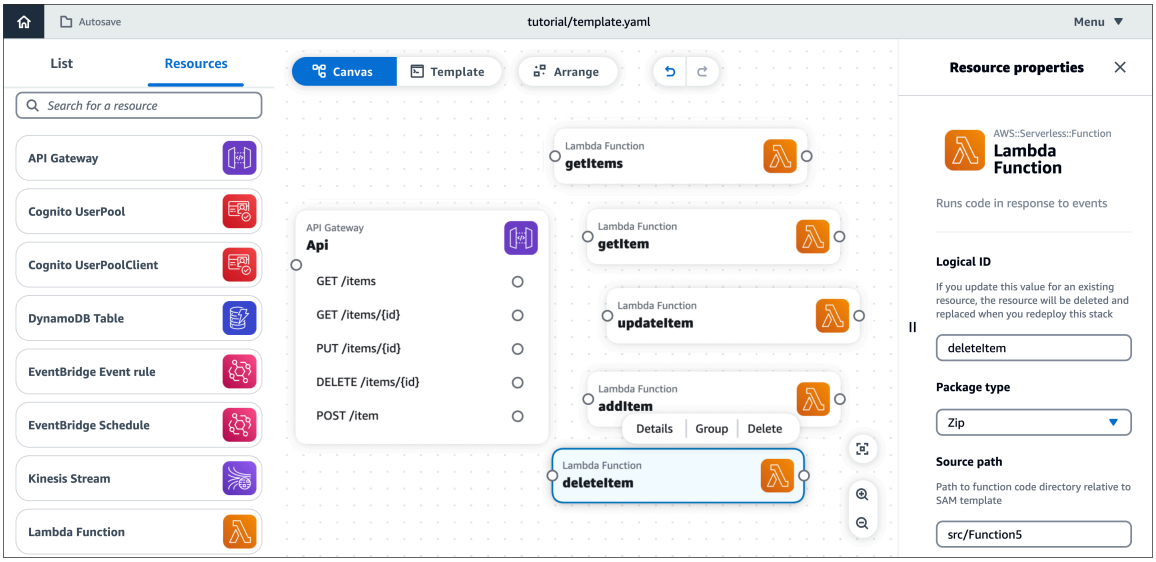


第 4 步：配置您的 Lambda 函数

按照资源属性参考表中的规定，分别命名五个 Lambda 函数。

命名 Lambda 函数

1. 打开 Lambda 函数卡片的资源属性面板。要打开面板，请双击卡片。或者，选择卡片，然后选择“详细信息”。
2. 在资源属性面板中，为逻辑 ID 输入指定的函数名称。例如 **`getItems`**。
3. 选择保存。
4. 重复前面的步骤，直到命名完所有五个函数。

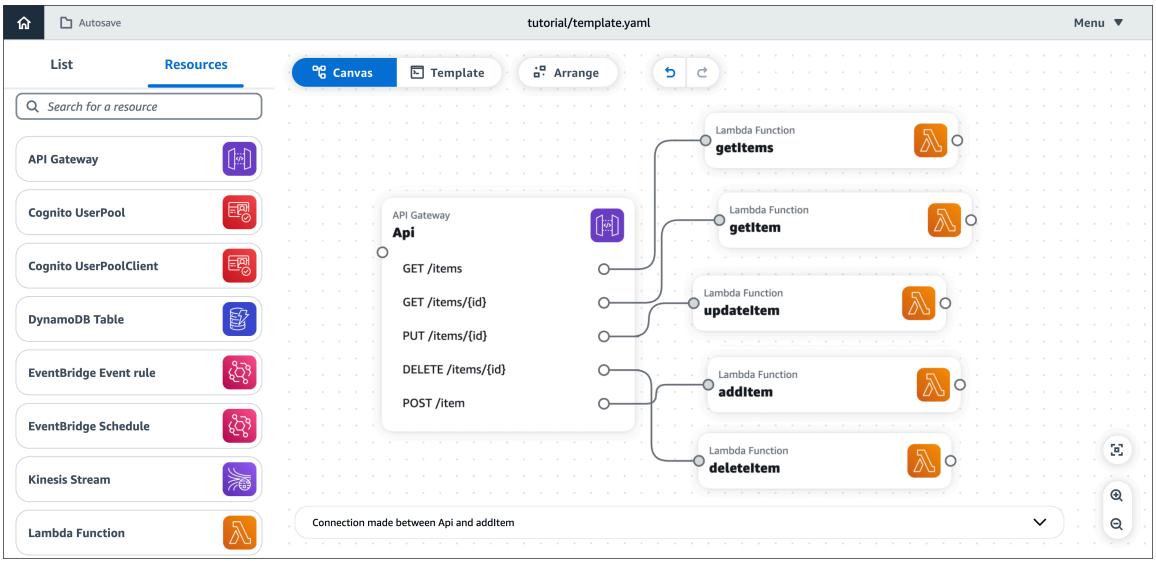


第 5 步：Connect 你的卡片

按照资源属性参考表中的指定，将 API Gatew ay 卡上的每条路由连接到其相关的 Lambda 函数卡。

连接您的卡片

1. 单击 API Gateway 卡上的右侧端口，然后将其拖到指定 Lambda 函数卡的左侧端口。例如，单击 GE T /item s 端口并将其拖到 get Items 的左侧端口。
2. 重复上一步操作，直到将 API Gat eway 卡上的所有五条路由都连接到相应的 Lambda 函数卡。



第 6 步：整理画布

通过将您的 Lambda 函数组合在一起并整理所有卡片来整理视觉画布。

将您的函数组合在一起

1. 按住 Shift 键，然后选择画布上的每张 Lambda 功能卡。
2. 选择分组。

为你的群组命名

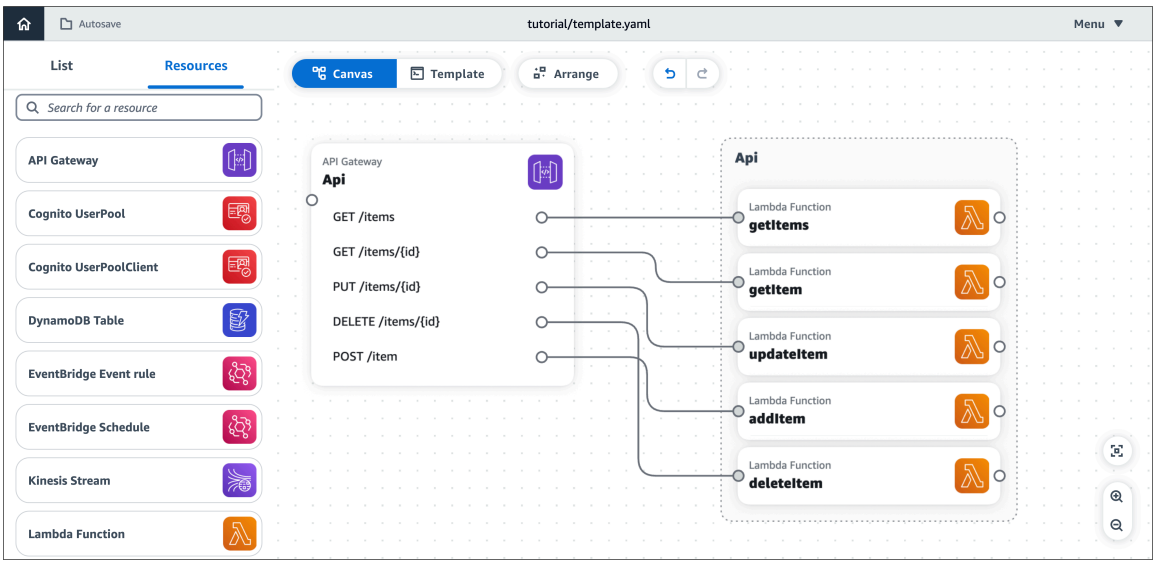
1. 双击群组顶部，靠近群组名称（群组）。

   群组属性面板打开。
2. 在 “群组属性” 面板上，在 “群组名称” 栏中输入**API**。
3. 选择保存。

整理你的卡片

在画布上，在主视图区域上方，选择排列。

Infrastructure Composer 排列和对齐视觉画布上的所有卡片，包括你的新群组 (API)，如下所示：

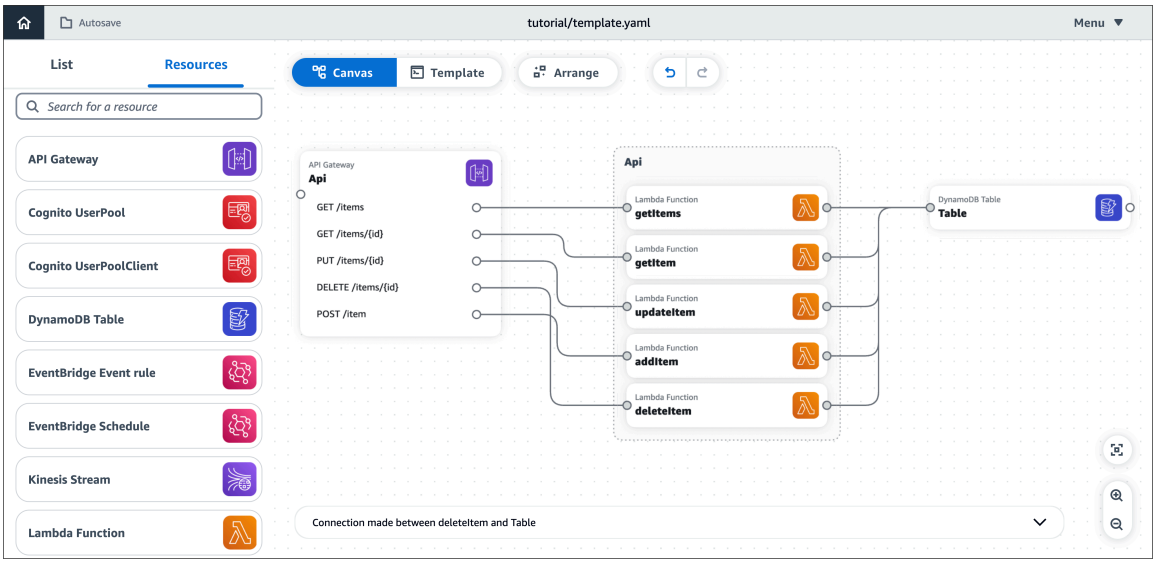


步骤 7：添加并连接 DynamoDB 表

现在，向您的应用程序架构中添加一个 DynamoDB 表，并将其连接到您的 Lambda 函数。

添加和连接 DynamoDB 表

1. 从资源选项板（资源）的“增强组件”部分下，将 D ynamoDB 表格卡片拖到画布上。
2. 单击 Lambda Func tion 卡上的右侧端口，然后将其拖到 DynamoDB 表卡的左侧端口。
3. 重复上一步操作，直到将所有五张 Lambda 功能卡都连接到 Dynam oDB 桌卡。
4. （可选）要重新整理和重新对齐画布上的卡片，请选择“排列”。

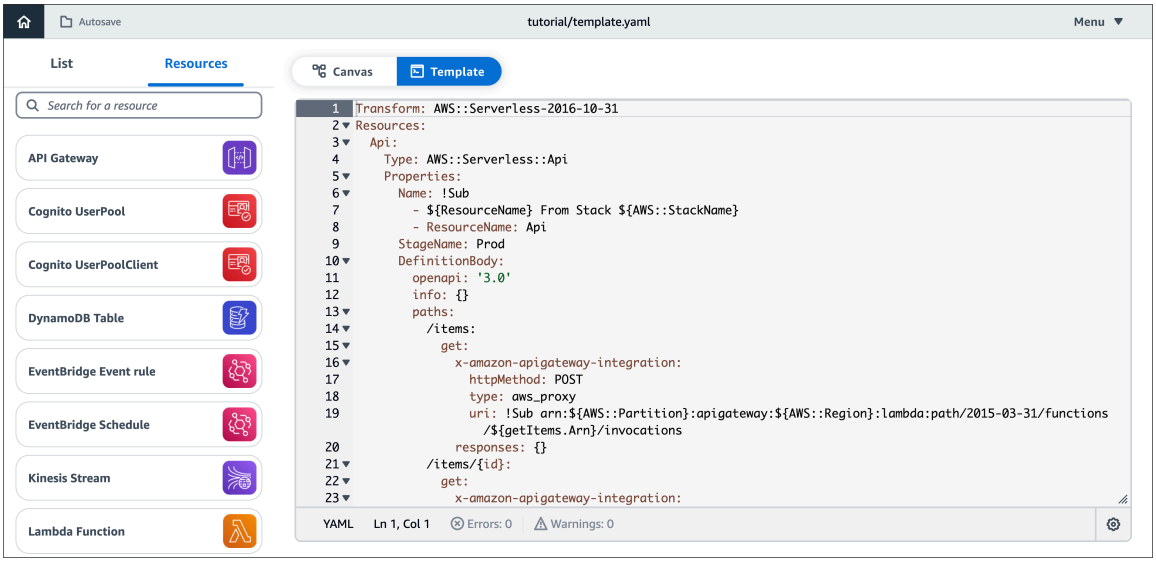


第 8 步：查看您的 AWS CloudFormation 模板

恭喜您！您已成功设计出可随时部署的无服务器应用程序。最后，选择 “模板”，查看基础架构编排器自动为您生成的 AWS CloudFormation 模板。

在模板中，基础设施编排器定义了以下内容：

- `Transform`声明，它将模板指定为 AWS Serverless Application Model (AWS SAM) 模板。有关更多信息，请参阅《AWS Serverless Application Model 开发人员指南》中的AWS SAM 模板剖析。
- 一种`AWS::Serverless::Api`资源，它指定你的 API Gateway REST API 及其五条路由。
- 五个`AWS::Serverless::Function`资源，用于指定您的 Lambda 函数的配置，包括其环境变量和权限策略。
- 一种`AWS::DynamoDB::Table`资源，用于指定您的 DynamoDB 表及其属性。
- 该`Metadata`部分包含有关您的资源组 (API) 的信息。有关本节的更多信息，请参阅《AWS CloudFormation 用户指南》中的元数据。



第 9 步：集成到您的开发工作流程中

使用 Infrastructure Composer 创建的模板文件和项目目录进行进一步的测试和部署。

- 通过本地同步，您可以将 Infrastructure Composer 连接到本地计算机上的 IDE 以加快开发速度。要了解更多信息，请参阅将基础架构编排控制台与本地 IDE 连接起来。

- 通过本地同步，您可以使用本地计算机上的 AWS Serverless Application Model 命令行界面 (AWS SAM CLI) 来测试和部署应用程序。要了解更多信息，请参阅将您的基础架构 Composer 无服务器应用程序部署到云端 AWS。

后续步骤

现在，您已经准备好使用基础架构编排器构建自己的应用程序了。有关使用基础架构编排器的详细资料，请参阅如何作曲 AWS 基础架构编辑器。准备好部署应用程序时，请参阅将您的基础架构 Composer 无服务器应用程序部署到云端 AWS。

你可以在哪里使用基础设施编排器

您可以从基础设施编排器的控制台中使用基础架构编排器 AWS Toolkit for Visual Studio Code，也可以通过控制 CloudFormation 台模式在基础设施编排器中使用。虽然每种用例略有不同，但总体而言，它们是相似的体验。本节提供每种体验的详细信息。

使用控制 AWS 基础架构编辑器 台本主题全面概述了默认控制台体验。该主题CloudFormation 控制台模式提供了与 CloudFormation 堆栈工作流集成的基础架构编排版本的详细信息。 AWS Toolkit for Visual Studio Code提供了有关在 VS Code 中访问和使用基础设施编排器的信息。

主题

- 使用控制 AWS 基础架构编辑器 台
- 在 CloudFormation 控制台模式下使用基础架构编排器
- 使用来自《基础设施编排器》 AWS Toolkit for Visual Studio Code

使用控制 AWS 基础架构编辑器 台

本节详细介绍了如何通过基础设施编排控制台进行访问和使用 AWS 基础架构编辑器 。这是基础设施编排器的默认体验，也是熟悉基础架构编排的好方法。您也可以将基础架构编排控制台与本地 IDE 集成。有关更多信息，请参阅 将基础架构编排控制台与本地 IDE 连接起来。

您还可以从 VS Code 中的 AWS 工具包中访问基础设施编排器，也可以使用专为在中使用而设计的基础设施编排器模式 CloudFormation。

有关使用基础架构编排器的一般文档，请参阅如何作曲。

主题

- AWS 基础架构编辑器 控制台视觉概述
- 通过基础设施编排控制台管理您的项目
- 将基础架构编排控制台与本地 IDE 连接起来
- 允许网页访问基础设施编排器中的本地文件
- 在基础设施编排控制台中本地同步并保存您的项目
- 从 Lambda 控制台将函数导入基础设施编排器
- 导出基础设施编排器可视画布的图像

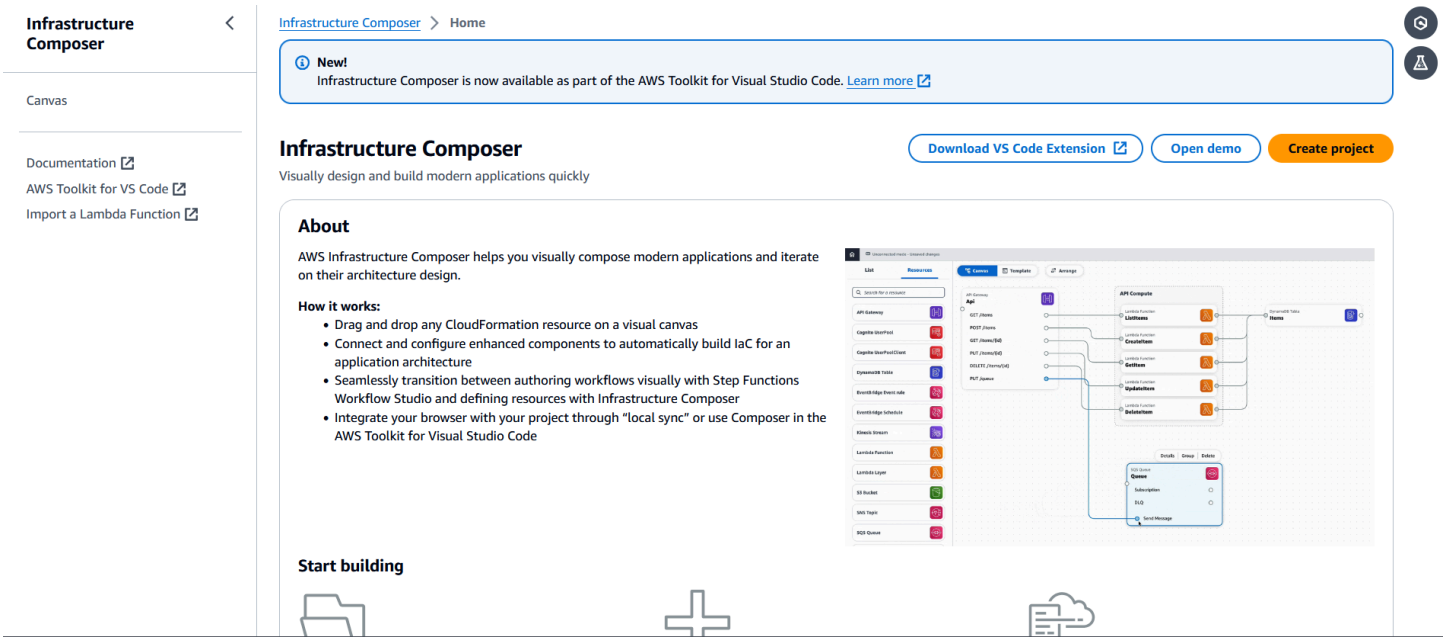
AWS 基础架构编辑器 控制台视觉概述

本节提供了 AWS 基础架构编辑器 控制台的直观概述。

主题

主页

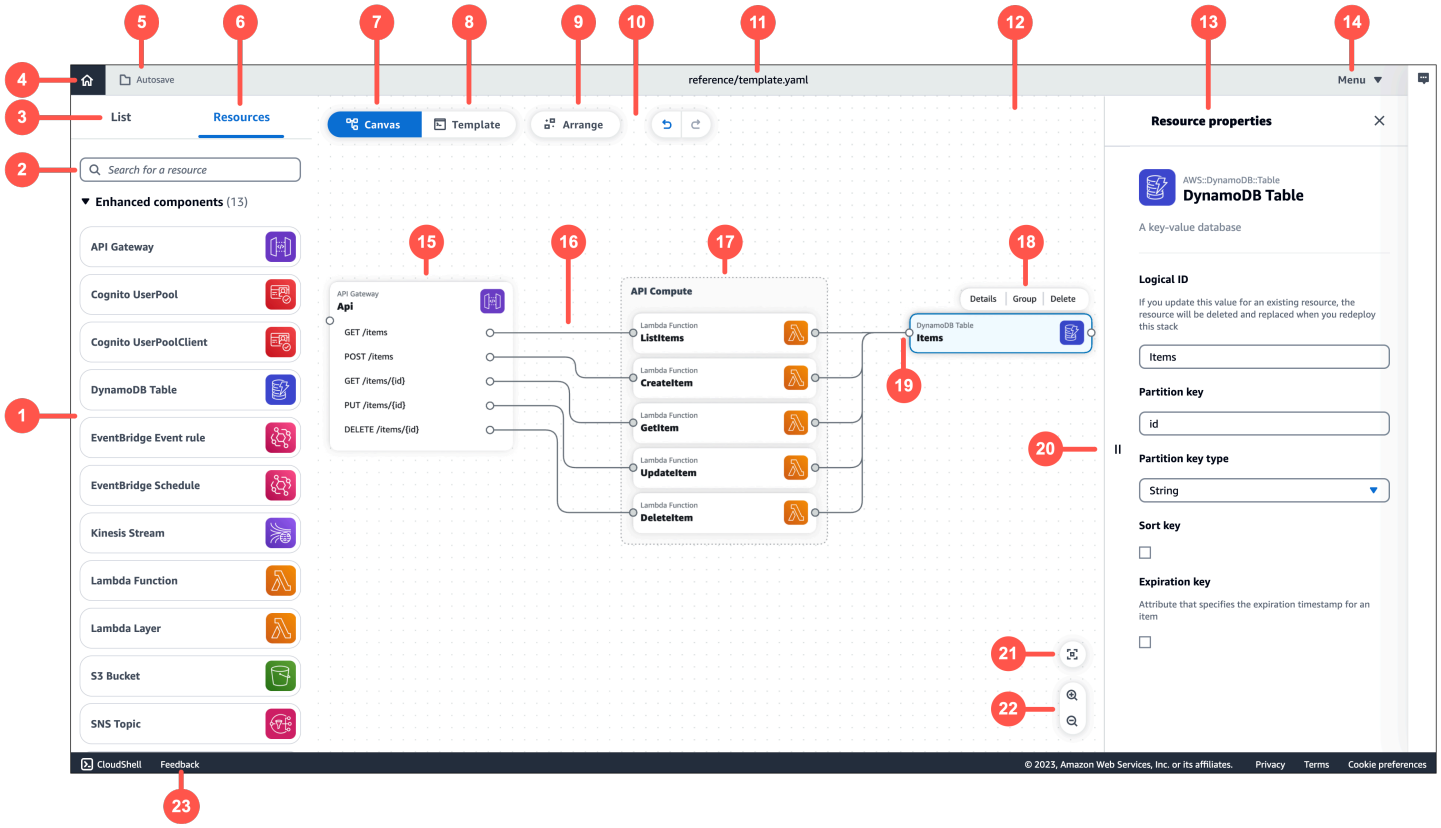
下图是基础设施编排控制台中的主页：



1. 文档-转至基础设施编排器文档。
2. 画布-进入画布并创建或加载项目。
3. 演示-打开基础架构编辑器演示应用程序。
4. 创建项目-创建或加载项目。
5. 开始构建-用于开始构建应用程序的快速链接。
6. 反馈 — 前往此处提交反馈。

视觉设计师和视觉画布

下图是 Infrastructure Composer 的可视化设计器和视觉画布的图像：



1. 资源调色板-显示可以用来设计的卡片。
2. 资源搜索栏-搜索可以添加到画布的卡片。
3. 列表-显示应用程序资源的树视图。
4. 主页 — 选择此处进入基础设施编排器主页。
5. 保存状态-表示是否将基础架构编排器更改保存到本地计算机。状态包括：
   - 自动保存-本地同步已激活，您的项目将自动同步和保存。
   - 更改已保存-您的应用程序模板已保存到本地计算机。
   - 未保存的更改-您的应用程序模板中有未保存到本地计算机的更改。
6. 资源-显示资源选项板。
7. 画布-在主视图区域中显示应用程序的画布视图。
8. 模板-在主视图区域显示应用程序的模板视图。
9. 排列-在画布中排列应用程序架构。
10. 撤消和重做-在支持时执行撤消和重做操作。
11. 模板名称-表示您正在设计的模板的名称。
12. 主视图区域-根据您的选择显示画布或模板。

13资源属性面板-显示在画布中选择的卡片的相关属性。此面板是动态的。在您配置卡片时，显示的属性将发生变化。

14菜单-提供常规选项，例如：

- 创建项目
- 打开模板文件或项目
- 保存模板文件
- 激活本地同步
- 导出画布
- 获取支持
- 键盘快捷键

15.卡片-在画布上显示卡片的视图。

16线-表示卡片之间的连接。

17分组 — 将选定的卡片组合在一起以进行可视化整理。

18.卡牌操作-提供您可以对卡片执行的操作。

a. 详细信息-打开资源属性面板。

b. 分组 — 将选定的卡片组合在一起。

c. 删除-从画布中删除卡片。

19端口-指向其他卡的连接点。

20资源属性字段-一组精心策划的属性字段，用于为您的卡片进行配置。

21重新居中 — 在视觉画布上将应用程序图表重新居中。

22缩放-在画布上放大和缩小。

23反馈 — 前往此处提交反馈。

通过基础设施编排控制台管理您的项目

本主题提供有关在基础设施编排控制台中管理项目时执行的基本任务的指导。这包括创建新项目、保存项目以及导入项目或模板等常见任务。如果您激活本地同步模式，也可以加载现有项目。激活本地同步模式后，您可以执行以下操作：

- 创建一个由起始模板和文件夹结构组成的新项目。
- 通过选择包含项目模板和文件的父文件夹，加载现有项目。
- 使用基础架构编排器管理您的模板和文件夹

在本地同步模式下，Infrastructure Composer 会自动将项目的模板和文件夹更改保存到本地计算机。如果您的浏览器不支持本地同步模式，或者您更喜欢在未激活本地同步模式的情况下使用 Infrastructure Composer，则可以创建新模板或加载现有模板。要保存更改，必须将模板导出到本地计算机。

> **Note**
>
> 基础架构 Composer 支持的应用程序包含以下内容：
>
> - 用于定义您的基础架构代码的 CloudFormation 或 AWS Serverless Application Model 模板。
> - 一种用于组织项目文件（例如 Lambda 函数代码、配置文件和生成文件夹）的文件夹结构。

主题

在基础设施编排控制台中创建新项目

创建新项目时，基础架构编排器会生成一个起始模板。当您在画布上设计应用程序时，您的模板会被修改。要保存您所做的工作，您必须导出模板或激活本地同步模式。

创建新的项目

1. 登录基础设施编排控制台。
2. 在主页上，选择创建项目。

> **Note**
>
> 您也可以在 Infrastructure Composer 中加载现有的，但必须先激活本地同步模式。激活后加载已激活本地同步的现有基础架构编排器项目，请参见加载现有项目。

在基础设施编排控制台中导入现有项目文件夹

使用本地同步模式，您可以导入现有项目的父文件夹。如果您的项目包含多个模板，则可以选择要加载的模板。

从主页导入现有项目

1. 登录基础设施编排控制台。
2. 在主页上，选择加载 CloudFormation 模板。
3. 对于项目位置，选择选择文件夹。选择项目的父文件夹，然后选择 “选择”。

> **Note**
>
> 如果您没有收到此提示，则您的浏览器可能不支持本地同步模式所需的文件系统访问 API。有关更多信息，请参阅 允许网页访问基础设施编排器中的本地文件。

4. 当浏览器出现提示时，选择 “查看文件”。
5. 对于模板文件，请从下拉列表中选择您的模板。如果您的项目包含单个模板，则基础架构编排器会自动为您选择该模板。
6. 选择创建。

从画布中导入现有项目

1. 在画布上选择 “菜单” 以打开菜单。
2. 在 “打开” 部分中，选择 “项目文件夹”。

> **Note**
>
> 如果 “项目文件夹” 选项不可用，则您的浏览器可能不支持本地同步模式所必需的 File System Access API。有关更多信息，请参阅 允许网页访问基础设施编排器中的本地文件。

3. 对于项目位置，选择选择文件夹。选择项目的父文件夹，然后选择 “选择”。
4. 当浏览器出现提示时，选择 “查看文件”。
5. 对于模板文件，请从下拉列表中选择您的模板。如果您的项目包含单个模板，则基础架构编排器会自动为您选择该模板。
6. 选择创建。

导入现有项目文件夹时，基础设施编排器会激活本地同步模式。对项目模板或文件所做的更改会自动保存到本地计算机上。

在基础设施编排控制台中导入现有项目模板

当您导入现有 CloudFormation 或 AWS SAM 模板时，Infrastructure Composer 会自动在画布上生成应用程序架构的可视化效果。

您可以从本地计算机导入项目模板。

导入现有项目模板

1. 登录基础设施编排控制台。
2. 选择 “创建项目” 以打开空白画布。
3. 选择 “菜单” 以打开菜单。
4. 在 “打开” 部分中，选择模板文件。
5. 选择您的模板并选择 “打开”。

要保存对模板的更改，必须导出模板或激活本地同步模式。

将现有项目模板保存到基础设施编排控制台中

如果您不使用本地同步模式，则必须导出模板才能保存更改。如果您已激活本地同步模式，则无需手动保存模板。更改会自动保存到您的本地计算机上。

保存现有项目模板

1. 在 “基础设施编排器” 画布中，选择 “菜单” 以打开菜单。
2. 在 “保存” 部分，选择 “保存模板文件”。
3. 为您的模板提供一个名称。
4. 选择保存模板的位置。
5. 选择保存。

将基础架构编排控制台与本地 IDE 连接起来

要将 Infrastructure Composer 控制台与本地集成开发环境 (IDE) 连接起来，请使用本地同步模式。此模式会自动将数据同步并保存到您的本地计算机。有关本地同步模式的更多信息，请参阅在基础设施编

排控制台中本地同步并保存您的项目。有关使用本地同步模式的说明，请参阅在基础设施编排控制台中本地同步并保存您的项目。

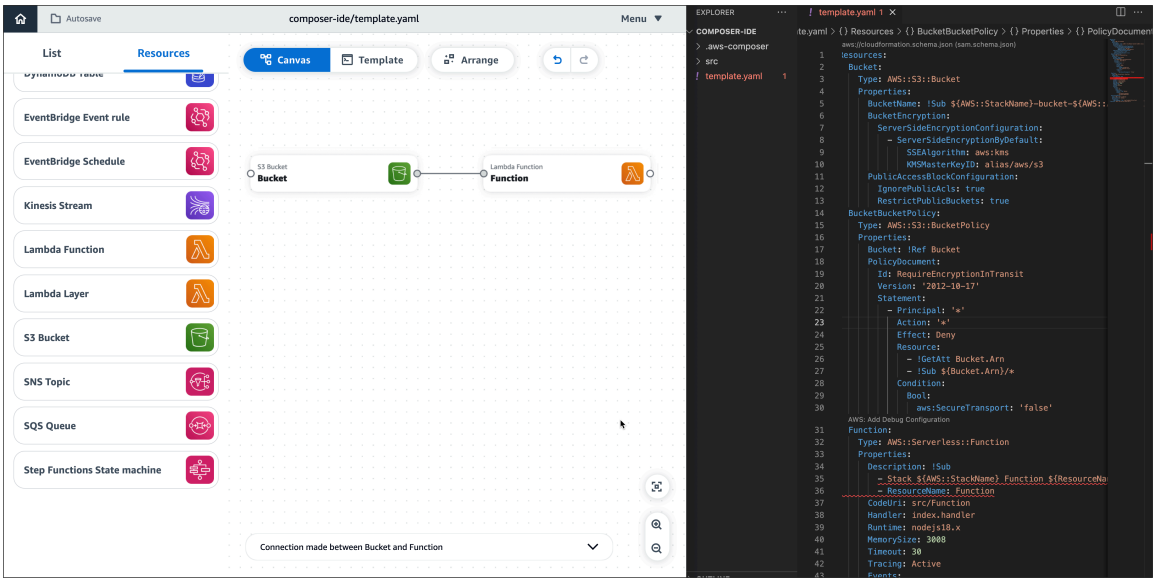
> **Note**
>
> 并非所有浏览器都提供“激活本地同步”选项。它在谷歌浏览器和微软 Edge 中可用。

在本地 IDE 中使用基础结构编排器的好处

当您在基础架构编辑器中进行设计时，您的本地模板和项目目录会自动同步和保存。

您可以使用本地 IDE 来查看更改和修改模板。您在本地所做的更改会自动同步到基础架构编排器。

您可以使用 AWS Serverless Application Model 命令行界面 (AWS SAM CLI) 等本地工具来构建、测试、部署应用程序等。以下示例显示了如何将资源拖放到基础架构编排器的可视化画布上，这反过来又会在本地 IDE 的 AWS SAM 模板中创建标记。



将基础架构编排器与本地 IDE 集成

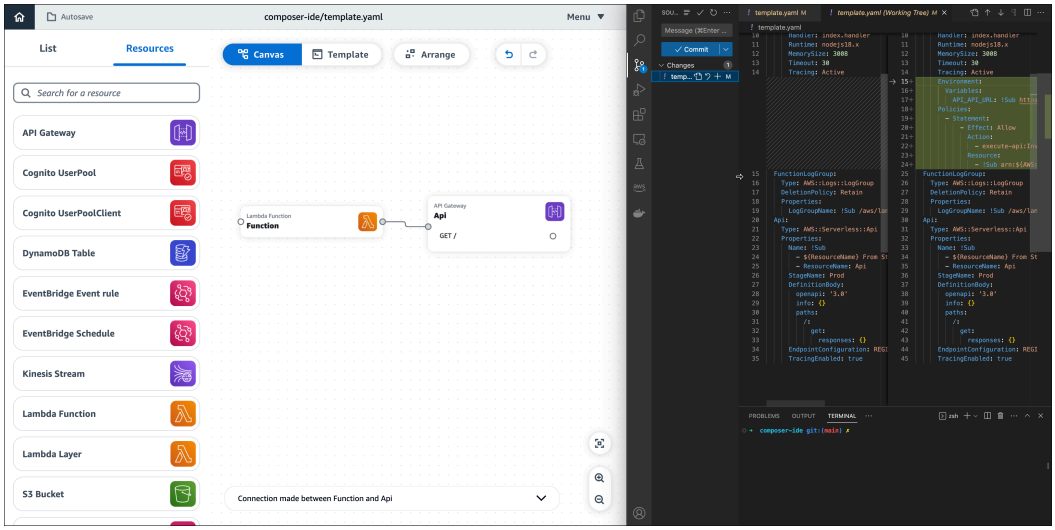
将基础结构编排器与本地 IDE 集成

1. 在 Infrastructure Composer 中，创建或加载项目，然后通过选择屏幕右上角的“菜单”按钮并选择“激活本地同步”来激活本地同步。

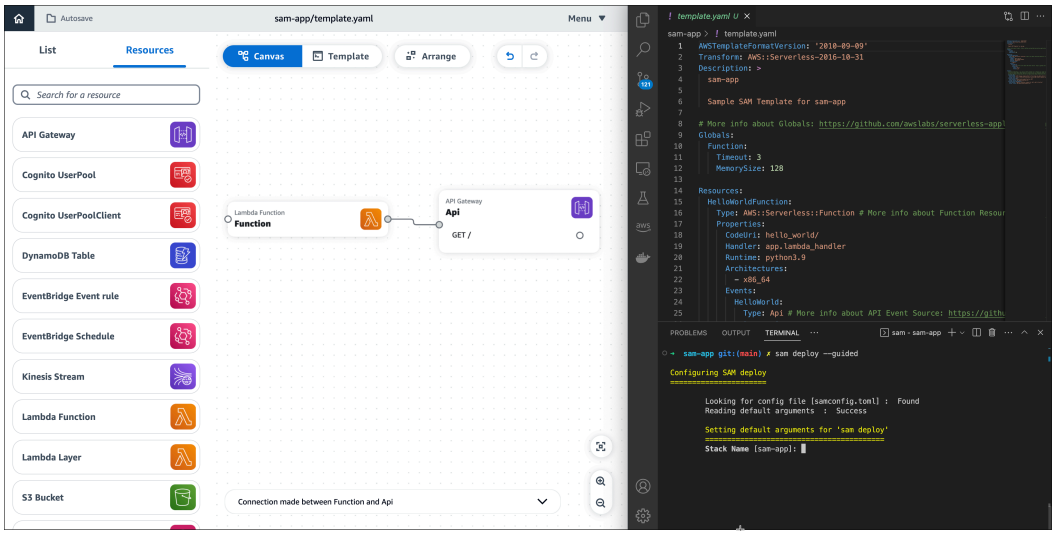
> **Note**
>
> 并非所有浏览器都提供 “激活本地同步” 选项。它在谷歌浏览器和微软 Edge 中可用。

2. 在本地 IDE 中，打开与基础结构编排器相同的项目文件夹。
3. 在本地 IDE 中使用基础结构编排器。在基础架构编排器中进行的更新将自动与您的本地计算机同步。以下是一些你可以做的事情的例子：

   a. 使用您选择的版本控制系统来跟踪基础设施编排器正在执行的更新。



   b. 在本地使用 AWS SAM CLI 来构建、测试、部署应用程序等。要了解更多信息，请参阅[将您的基础架构 Composer 无服务器应用程序部署到云端 AWS](#)。



允许网页访问基础设施编排器中的本地文件

基础设施编排控制台支持本地同步模式和从 Lambda 控制台导入函数。要使用这些功能，需要支持文件系统访问 API 的 Web 浏览器。任何最新版本的 Google Chrome 和 Microsoft Edge 都支持文件系统访问 API 的所有功能，并且可以在 Infrastructure Composer 中与本地同步模式一起使用。

File System Access API 允许网页访问您的本地文件系统，以便读取、写入或保存文件。默认情况下，此功能处于关闭状态，需要您通过视觉提示获得许可才能启用。一旦获得授权，此访问权限将在您的网页浏览器会话期间保留。

要了解有关文件系统访问 API 的更多信息，请参阅：

- mdn 网络文档中的@@ 文件系统访问 API。
- 文件系统访问 API：简化对 web.dev 网站中本地文件的访问。

本地同步模式

本地同步模式允许您在Infrastructure Composer中进行设计时自动同步和保存模板文件和项目文件夹。要使用此功能，需要支持文件系统访问 API 的 Web 浏览器。

数据基础架构编排器可以访问

Infrastructure Composer 获得对您允许的项目文件夹以及该项目文件夹的任何子文件夹的读写权限。此访问权限用于创建、更新和保存设计时生成的任何模板文件、项目文件夹和备份目录。Infrastructure Composer 访问的数据不用于任何其他目的，也不会存储在本地文件系统之外的任何地方。

访问敏感数据

File System Access API 排除或限制对可能包含敏感数据的特定目录的访问。如果您选择其中一个目录与 Infrastructure Composer 本地同步模式一起使用，则会出现错误。您可以选择其他本地目录进行连接，也可以在禁用本地同步的情况下在默认模式下使用 Infrastructure Composer。

有关更多信息，包括敏感目录的示例，请参阅用户授予的敏感文件访问权限超出文件系统访问权限 W3C 草稿社区组报告中的预期。

如果您使用Windows Subsystem for Linux (WSL)，则由于整个Linux目录位于您的Windows系统中，文件系统访问 API 将无法访问整个目录。您可以在禁用本地同步的情况下使用 Infrastructure Composer，也可以配置解决方案，将项目文件从您的WSL目录同步到中的Windows工作目录。然后，使用基础架构编排器本地同步模式来处理您的Windows目录。

在基础设施编排控制台中本地同步并保存您的项目

本节提供有关使用 Infrastruct ure Composer 的本地同步模式自动同步项目并将其保存到本地计算机的信息。

出于以下原因，我们建议您使用本地同步：

您可以为新项目激活本地同步，也可以在激活本地同步的情况下加载现有项目。

- 默认情况下，您需要在设计时手动保存应用程序模板。在您进行更改时，使用本地同步功能将应用程序模板自动保存到本地计算机。
- 本地同步可管理您的项目文件夹、备份文件夹和支持的外部文件并将其自动同步到本地计算机。
- 使用本地同步时，您可以将 Infrastructure Composer 与本地 IDE 连接以加快开发速度。要了解更多信息，请参阅将基础架构编排控制台与本地 IDE 连接起来。

本地同步模式可以保存什么

本地同步模式会自动同步以下内容并将其保存到本地计算机：

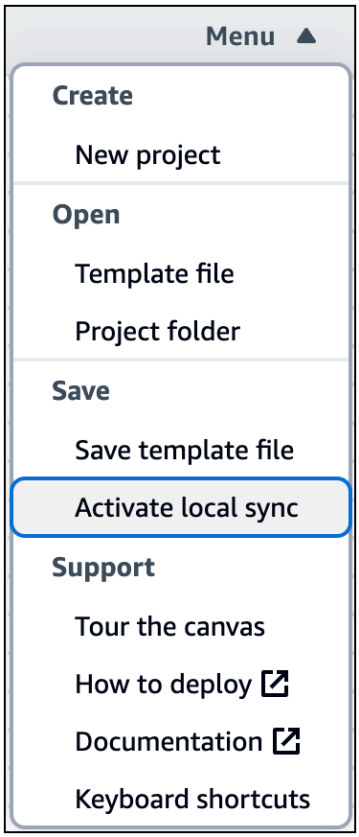
- 应用程序模板文件-包含基础架构即代码 AWS Serverless Application Model (IaC AWS SAM) 的 AWS CloudFormation 或 () 模板。
- 项目文件夹-用于组织 AWS Lambda 函数的通用目录结构。
- Backup 目录 — 名为的备份目录`.aws-composer`，在项目位置的根目录中创建。此目录包含应用程序模板文件和项目文件夹的备份副本。
- 外部文件-支持可在基础架构编排器中使用的外部文件。要了解更多信息，请参阅在基础架构编辑器中引用外部文件。

浏览器要求

本地同步模式需要支持文件系统访问 API 的浏览器。有关更多信息，请参阅 允许网页访问基础设施编排器中的本地文件。

激活本地同步模式

默认情况下，本地同步模式处于停用状态。您可以通过基础设施编辑器菜单激活本地同步模式。



有关激活本地同步和现有加载项目的说明，请参阅以下主题：。

- 在基础设施编辑器中激活本地同步
- 加载已激活本地同步的现有基础架构编排器项目

在基础设施编辑器中激活本地同步

要激活本地同步，请完成以下步骤：

1. 在基础设施编排器主页上，选择创建项目。
2. 从 “基础设施编排” 菜单中，选择 “激活本地同步”。
3. 在 “项目位置” 中，按选择文件夹，然后选择一个目录。基础设施编排器将在您设计时保存和同步您的模板文件和文件夹。

   > **Note**
   >
   > 项目位置不得包含现有的应用程序模板。

4. 当系统提示允许访问时，选择 “查看文件”。

5. 按激活。当系统提示保存更改时，选择保存更改。

   激活后，自动保存指示器将显示在画布的左上角区域。

加载已激活本地同步的现有基础架构编排器项目

要加载已激活本地同步的现有项目，请完成以下步骤：

1. 在基础设施编排器主页上，选择加载 CloudFormation 模板。
2. 从 “基础架构编辑器” 菜单中，选择 “打开” > “项目文件夹”。
3. 对于项目位置，按选择文件夹，然后选择项目的根文件夹。
4. 当系统提示允许访问时，选择 “查看文件”。
5. 对于模板文件，选择您的应用程序模板并按创建。
6. 当系统提示保存更改时，选择保存更改。

   激活后，自动保存指示器将显示在画布的左上角区域。

从 Lambda 控制台将函数导入基础设施编排器

基础架构编排器提供了与 AWS Lambda 控制台的集成。您可以将 Lambda 函数从 Lambda 控制台导入到基础设施编排控制台。然后，使用基础设施编排器画布进一步设计您的应用程序架构。

- 此集成需要支持文件系统访问 API 的浏览器。有关更多信息，请参阅 允许网页访问基础设施编排器中的本地文件。
- 将 Lambda 函数导入基础设施编排器时，必须激活本地同步模式才能保存所有更改。有关更多信息，请参阅 在基础设施编排控制台中本地同步并保存您的项目。

要开始使用此集成，请参阅《AWS Lambda 开发人员指南》 AWS 基础架构编辑器中的 “AWS Lambda 与一起使用”。

导出基础设施编排器可视画布的图像

本主题介绍 AWS 基础架构编辑器 控制台导出画布功能。

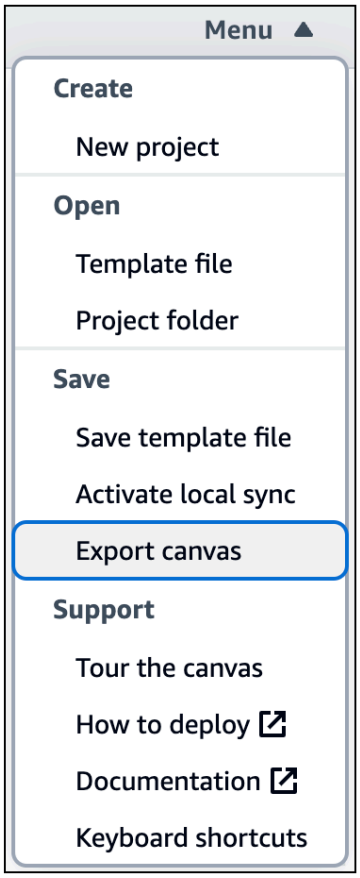
有关基础设施编排器所有功能的直观概述，请参阅AWS 基础架构编辑器 控制台视觉概述。

关于导出画布

导出画布功能将应用程序的画布作为图像导出到本地计算机。

- Infrastructure Composer 会移除可视化设计器用户界面元素，并仅导出应用程序
- 默认图像文件格式为png。
- 该文件将导出到本地计算机的默认下载位置。

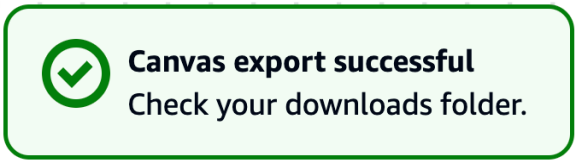
您可以从菜单访问导出画布功能。



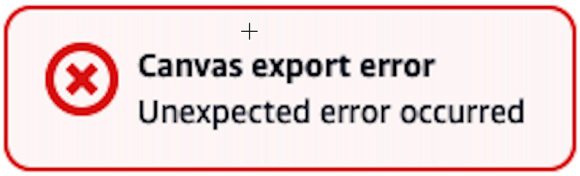
导出画布

导出画布时，基础设施编排器会显示一条状态消息。

如果导出成功，您将看到以下消息：



如果导出失败，您将看到一条错误消息。如果您收到错误消息，请再次尝试导出。



在 CloudFormation 控制台模式下使用基础架构编排器

CloudFormation 控制台模式下的基础架构编辑器是用于可视化 CloudFormation 模板的推荐工具。您也可以使用此工具来创建和编辑 CloudFormation 模板。

该模式与基础设施编辑器控制台有何不同？

CloudFormation 控制台模式下的 Infrastructure C omposer 通常具有与默认基础架构编排控制台相同的功能，但有一些区别需要注意。

- 此模式已与 CloudFormation 控制台中的堆栈工作流程集成。这允许您直接在中使用基础设施编排器 CloudFormation。
- 在基础设施编排控制台中本地同步并保存您的项目不支持自动将数据同步并保存到本地计算机的功能。
- 与 Lambda 相关的卡（Lambda 函数和 L ambda 层）需要在此模式下不可用的代码构建和打包解决方案。

> **Note**
>
> 这些卡片和本地同步可以在基础架构 Composer 控制台或 AWS Toolkit for Visual Studio Code.

当您从 CloudFormation 控制台打开基础设施编排器时，基础设施编排器将在 CloudFormation 控制台模式下打开。在此模式下，您可以使用基础设施编排器来可视化、创建和更新您的模板。

如何在 CloudFormation 控制台模式下访问基础架构编排器

CloudFormation 控制台模式下的基础架构编排器是从 D CloudFormation esigner 升级而来的。我们建议使用基础架构编排器来可视化您的 CloudFormation 模板。您也可以使用此工具来创建和编辑 CloudFormation 模板。

1. 前往 Cloudformation 控制台并登录。

2. 从左侧导航菜单中选择 “基础设施编排”。这将带你进入 CloudFormation 控制台模式下的基础设施编排器。

> **Note**
>
> 有关在 CloudFormation 控制台模式下使用基础设施编排器的信息，请参阅在 CloudFormation 控制台模式下使用基础架构编排器。

在 CloudFormation 控制台模式下在基础设施编排器中可视化部署

按照本主题中的说明可视化已部署的 CloudFormation 堆栈/基础设施编排器模板。

1. 转到CloudFormation 控制台并登录。
2. 选择要编辑的堆栈。
3. 选择 “模板” 选项卡。
4. 选择 “基础架构编排”。

基础架构编排器将可视化您的堆栈/模板。也可以在此处进行更改。

在 CloudFormation 控制台模式下的基础架构编排器中创建新模板

按照本主题中的说明创建新模板。

1. 转到CloudFormation 控制台并登录。
2. 从左侧导航菜单中选择 “基础设施编排”。这将在 CloudFormation 控制台模式下打开基础架构编排器。
3. 从 “资源” 面板中拖放、配置和连接所需的资源（卡片）。

> **Note**
>
> 有关使用基础设施编排器的详细信息，请参阅如何作曲；请注意，与 Lambda 相关的卡（Lamb da 函数和 Lamb da 层）需要在控制台模式下的基础设施编排器中不可用的代码构建和打包解决方案。 CloudFormation 这些卡片可以在基础设施编排控制台中使用，或者 AWS Toolkit for Visual Studio Code。有关使用这些工具的信息，请参阅你可以在哪里使用基础设施编排器。

4. 双击卡片可使用资源属性面板来指定卡片的配置方式。
5. 连接您的卡片以指定应用程序的事件驱动工作流程。
6. 选择 “模板” 以查看和编辑您的基础设施代码。更改会自动与您的画布视图同步。
7. 模板准备好导出到堆栈后，选择创建模板。
8. 选择 “确认并导出至 CloudFormation” 按钮。这将带您回到创建堆栈的工作流程，并显示一条确认您的模板已成功导入的消息。

> **Note**
>
> 只能导出包含资源的模板。

9. 在创建堆栈工作流程中，选择下一步。
10. 提供堆栈名称，查看所有列出的参数，然后选择下一步。

> **Note**
>
> 堆栈名称必须以字母开头，并且只能包含字母、数字、破折号。

11. 提供以下信息后，选择 “下一步”：

    - 与堆栈关联的标签
    - 堆栈权限
    - 堆栈的故障选项

> **Note**
>
> 有关管理堆栈的指导，请参阅《CloudFormation 用户指南》中的CloudFormation 最佳实践。

12. 确认您的堆栈详细信息正确无误，查看页面底部的确认信息，然后选择 “提交” 按钮。

CloudFormation 将开始根据模板中的数据创建堆栈。

在 CloudFormation 控制台模式下更新基础设施编排器中的现有堆栈

按照本主题中的说明更新现有 CloudFormation 堆栈。

> **Note**
>
> 如果您的文件保存在本地，我们建议使用AWS Toolkit for Visual Studio Code。

1. 转到CloudFormation 控制台并登录。
2. 选择要编辑的堆栈。
3. 点击更新按钮。执行此操作将带您进入更新堆栈向导。
4. 在右侧，选择 “在基础设施编排器中编辑”。
5. 选择下面标有 “在基础设施编排器中编辑” 的按钮。这将带你进入 CloudFormation 控制台模式下的基础设施编排器。
6. 在这里，您可以从资源面板中拖放、配置和连接资源（卡片）。

   > **Note**
   >
   > 有关使用基础设施编排器的详细信息，请参阅如何作曲；请注意，与 Lambda 相关的卡（Lamb da 函数和 Lamb da 层）需要在控制台模式下的基础设施编排器中不可用的代码构建和打包解决方案。 CloudFormation 这些卡片可以在基础设施编排控制台中使用，或者 AWS Toolkit for Visual Studio Code。有关使用这些工具的信息，请参阅你可以在哪里使用基础设施编排器。

7. 准备好将更改导出到时 CloudFormation，请选择更新模板。
8. 选择 “确认” 并继续 CloudFormation。这将带您返回更新堆栈工作流程，并显示一条确认您的模板已成功导入的消息。

   > **Note**
   >
   > 只能导出包含资源的模板。

9. 在更新堆栈工作流程中，选择下一步。
10. 查看所有列出的参数，然后选择 “下一步”。
11. 提供以下信息后，选择 “下一步”：

    - 与堆栈关联的标签
    - 堆栈权限
    - 堆栈的故障选项

> **Note**
>
> 有关管理堆栈的指导，请参阅《CloudFormation 用户指南》中的CloudFormation 最佳实践。

12. 确认您的堆栈详细信息正确无误，查看页面底部的确认信息，然后选择 “提交” 按钮。

CloudFormation 将根据您在模板中所做的更新开始更新堆栈。

使用来自《基础设施编排器》 AWS Toolkit for Visual Studio Code

本节介绍如何 AWS 基础架构编辑器 从中使用AWS Toolkit for Visual Studio Code。其中包括中对基础架构编排器的可视化概述 AWS Toolkit for Visual Studio Code。它还包括说明如何访问此体验以及如何将项目从 VS Code 同步到 AWS 云端。要进行同步，请使用中的sam sync命令 AWS SAMCLI。本节还提供了有关在《基础架构编排器》中使用 Amazon Q while 的指导 AWS Toolkit for Visual Studio Code。

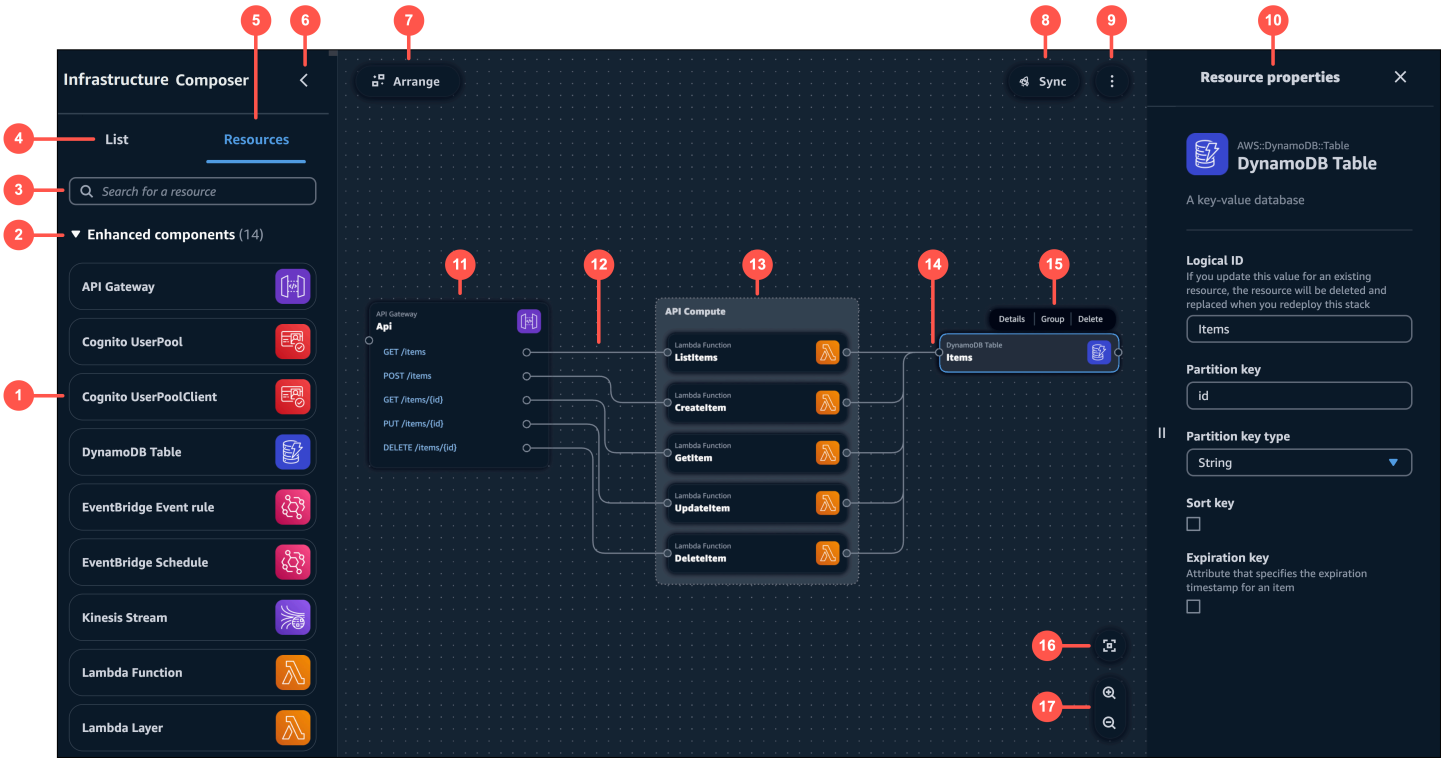
有关使用中基础设施编排器的更多指导 AWS Toolkit for Visual Studio Code，请参阅如何作曲。本节中的内容适用于此体验以及 Infrastructure Composer 控制台体验。

主题

- 来自《基础架构编排器》的可视化概述 AWS Toolkit for Visual Studio Code
- 从访问基础架构编排器 AWS Toolkit for Visual Studio Code
- 同步基础架构编辑器以部署到 AWS 云
- AWS 基础架构编辑器 与一起使用 Amazon Q Developer

来自《基础架构编排器》的可视化概述 AWS Toolkit for Visual Studio Code

中的 Infrastructure Composer 的可视化设计器 AWS Toolkit for Visual Studio Code 包括一个可视化画布，其中包括在下图中编号并在下面列出的组件。



1. 资源调色板-显示可以用来设计的卡片。
2. 卡片类别 — 卡片按基础架构编辑器特有的类别进行组织。
3. 资源搜索栏-搜索可以添加到画布的卡片。
4. 列表-显示应用程序资源的树视图。
5. 资源-显示资源选项板。
6. 左窗格切换-隐藏或显示左侧窗格。
7. 排列-在画布中排列应用程序架构。
8. 同步-启动 AWS Serverless Application Model (AWS SAM) CLI `sam sync` 命令以部署您的应用程序。
9. 菜单-提供常规选项，例如：
   - 导出画布
   - 游览画布
   - 指向文档的链接
   - 键盘快捷键
10. 资源属性面板-显示在画布中选择的卡片的相关属性。此面板是动态的。在您配置卡片时，显示的属性将发生变化。
11. 卡片-在画布上显示卡片的视图。

12线-表示卡片之间的连接。

13组 — 一组牌。您可以对卡片进行分组以进行视觉整理。

14端口-指向其他卡的连接点。

15.卡牌操作-提供您可以对卡片执行的操作。

- 详细信息-打开 “资源属性” 面板。
- 分组 — 将选定的卡片组合在一起。
- 删除-从画布和模板中删除卡片。

16重新居中 — 在视觉画布上将应用程序图表重新居中。

17缩放-在画布上放大和缩小。

从访问基础架构编排器 AWS Toolkit for Visual Studio Code

按照本主题中的说明从访问基础架构编排器 AWS Toolkit for Visual Studio Code。

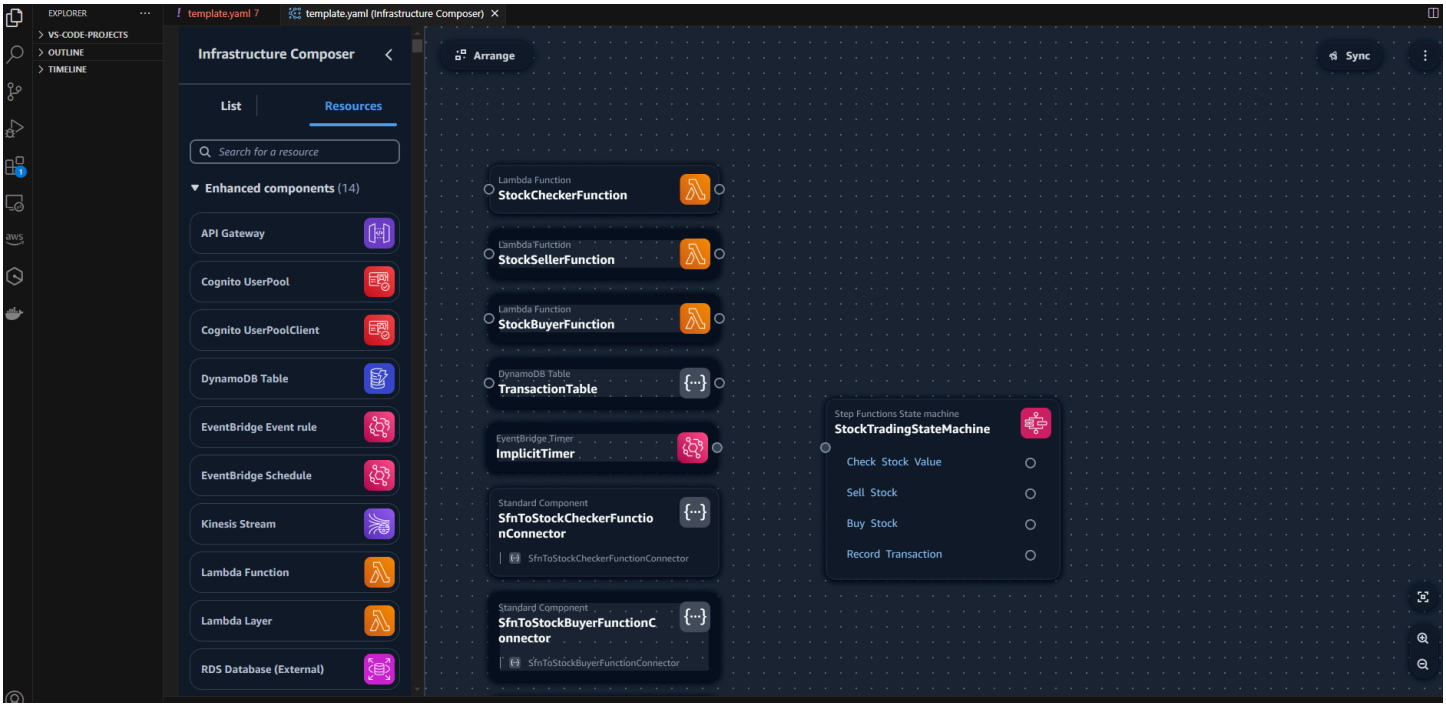
> **Note**
>
> 必须先下载并安装适用于 VS Code 的 AWS Toolkit for Visual Studio Code Toolkit for VS Code，然后才能从访问基础架构编排器。有关说明，请参阅下载适用于 VS Code 的工具包。

从适用于 VS Code 的 Toolkit 访问基础设施编排器

您可以通过以下任一方式访问基础架构编排器：

1. 从任意 CloudFormation 或 AWS SAM 模板中选择 “基础设施编排” 按钮。
2. 通过右键单击您的 CloudFormation 或 AWS SAM 模板，进入上下文菜单。
3. 来自 VS Code 命令面板。

以下是通过基础设施编排器按钮访问基础设施编排器的示例：



有关访问基础架构编排器的更多信息，请参阅AWS 基础架构编辑器 从工具包访问。

同步基础架构编辑器以部署到 AWS 云

使用 AWS 基础架构编辑器 中的同步按钮 AWS Toolkit for Visual Studio Code 将您的应用程序部署到 AWS 云。

同步按钮从`sam sync`命令行界面 (CLI) 启动 AWS SAM 命令。

该`sam sync`命令可以部署新应用程序或将您在本地所做的更改快速同步到 AWS 云。跑步`sam sync`可能包括以下内容：

- 通过创建或更新本地`.aws-sam`目录`sam build`来构建应用程序，为部署做好本地应用程序文件的准备。
- 对于支持 AWS 服务的资源 APIs， AWS SAM CLI将使用 APIs 来部署您的更改。这样 AWS SAM CLI做是为了快速更新您在云中的资源。
- 如有必要， AWS SAM CLI会执行 AWS CloudFormation 部署以通过更改集更新整个堆栈。

该`sam sync`命令最适合快速开发环境，因为快速更新您的云资源可以使您的开发和测试工作流程受益。

要了解更多信息`sam sync`，请参阅AWS Serverless Application Model 开发者指南中的使用 sam sync。

设置

要在 Infrastructure Composer 中使用同步功能，您必须在本地计算机上 AWS SAM CLI安装同步功能。有关说明，请参阅《AWS Serverless Application Model 开发人员指南》 AWS SAMCLI中的安装。

当你在 Infrastructure Composer 中使用同步功能时，会 AWS SAM CLI引用你的配置文件，获取将你的应用程序同步到所需的信息 AWS 云。有关创建、修改和使用配置文件的说明，请参阅AWS Serverless Application Model 开发人员指南中的配置项目设置。

同步和部署您的应用程序

要将您的应用程序同步到 AWS 云

1. 在 “基础设施编排” 画布上选择 “同步” 按钮。
2. 您可能会收到一条提示，要求您确认您正在使用开发堆栈。选择 “确定” 继续。
3. 基础架构编排器可能会提示您配置以下选项：

   - AWS 区域— 要将应用程序同步到的区域。
   - CloudFormation 堆栈名称- CloudFormation 堆栈的名称。您可以选择现有堆栈名称或创建新堆栈名称。
   - 亚马逊简单存储服务 (Amazon S3) S ervice 存储桶 — 您的亚马逊 S3 存储桶的名称。 AWS SAM CLI将在此打包并存储您的应用程序文件和函数代码。您可以选择现有存储桶或创建新存储桶。

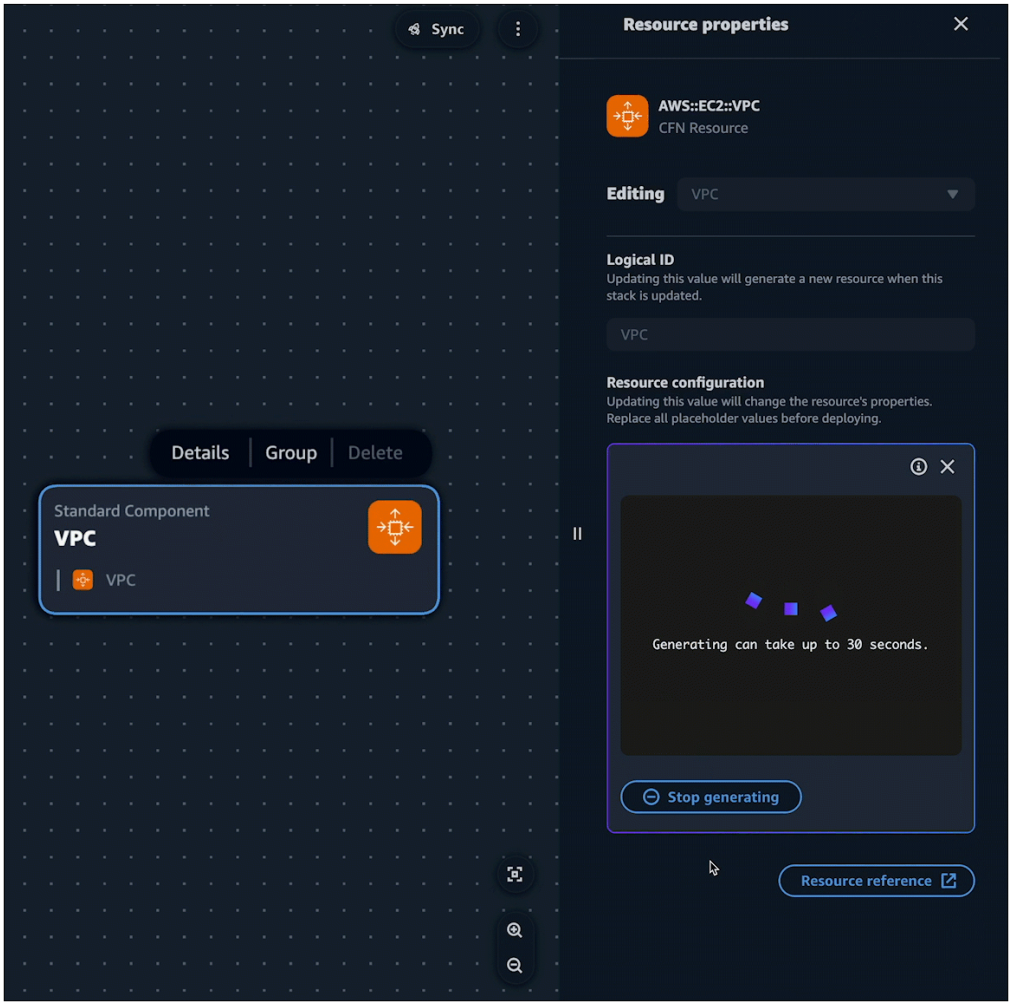
Infrastructure Composer 将启动该 AWS SAM CLI`sam sync`命令并在 IDE 中打开一个终端窗口以输出其进度。

AWS 基础架构编辑器 与一起使用 Amazon Q Developer

AWS 基础架构编辑器 来自 AWS Toolkit for Visual Studio Code 提供了与的集成Amazon Q。在设计应用程序时，您可以Amazon Q在 Infrastructure Composer 中使用来为您的 AWS 资源生成基础架构代码。

Amazon Q是一款由机器学习驱动的通用代码生成器。要了解更多信息，请参阅什么是Amazon Q？ 在《Amazon Q Developer用户指南》中。

对于标准资源和标准组件卡，您可以使用Amazon Q为资源生成基础设施代码建议。



标准资源和标准组件卡片可以表示 CloudFormation 资源或 CloudFormation 资源集合。要了解更多信息，请参阅在基础设施编排器中配置和修改卡片。

设置

要Amazon Q在基础设施编排器中使用，您必须在工具包Amazon Q中使用进行身份验证。有关说明，请参阅《VS Code 入门 JetBrains》和《Amazon Q Developer用户指南》。Amazon Q

Amazon Q Developer在基础架构编排器中使用

您可以Amazon Q Developer从任何标准资源或标准组件卡的资源属性面板中使用。

要Amazon Q在基础架构编辑器中使用

1. 在标准资源或标准组件卡片中，打开资源属性面板。
2. 找到 “资源配置” 字段。此字段包含卡片的基础设施代码。

3. 选择 “生成建议” 按钮。 Amazon Q将生成建议。

> **Note**
>
> 在此阶段生成的代码不会覆盖模板中的现有基础架构代码。

4. 要生成更多建议，请选择重新生成。您可以切换样本以比较结果。
5. 要选择一个选项，请选择 “选择”。在将代码保存到应用程序之前，您可以在此处对其进行修改。要在不保存的情况下退出，请选择退出图标 (X)。
6. 要将代码保存到应用程序模板中，请从 “资源属性” 面板中选择 “保存”。

了解详情

要了解有关 Amazon Q 的更多信息，请参阅《Amazon Q Developer 用户指南》中的什么是 Amazon Q？。

如何作曲 AWS 基础架构编辑器

本节介绍使用基础架构编排控制台、和中的基础架构编排CloudFormation 控制台模式器的基础知识AWS Toolkit for Visual Studio Code。更具体地说，本节中的主题提供了有关如何使用 Infrastructure Composer 编写应用程序的关键细节，并包括有关其他功能和快捷方式的详细信息。主机和 VS Code 体验在功能上存在一些差异，本节中的主题确定并描述了这些变化发生的地方。

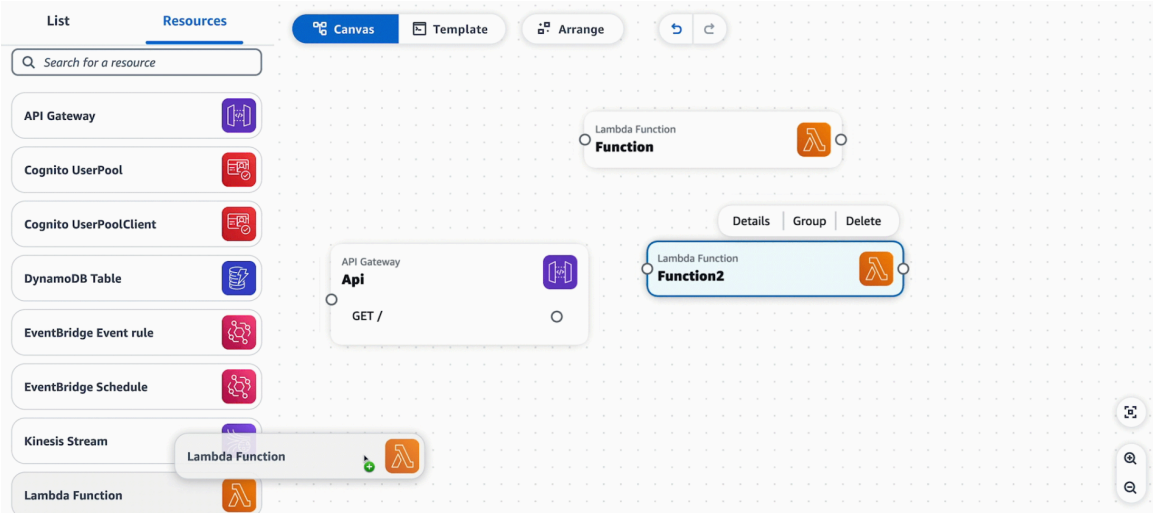
编写应用程序后，您可以随时查看将您的基础架构 Composer 无服务器应用程序部署到云端 AWS有关部署应用程序的信息。

主题

将卡片放在基础设施编排器的可视画布上

本节介绍如何在其可视化画布中选择和拖动 “基础架构编排器” 卡片。在开始之前，请确定您的应用程序需要哪些资源以及它们需要如何进行交互。有关执行此操作的提示，请参阅使用基础架构编排器构建您的第一个应用程序。

要向应用程序添加卡片，请将其从资源调色板中拖放到可视画布上。



您可以从两种类型的卡中进行选择：增强型组件卡和标准 IaC 资源卡。

将卡片放在视觉画布上后，你就可以分组、连接、排列和配置卡片了。有关执行此操作的信息，请参阅以下主题：

- 在基础设施编排器的可视画布上将卡片组合在一起
- 在基础设施编辑器的可视化画布上连接卡片
- 在基础设施编辑器的可视画布上排列卡片
- 在基础设施编排器中配置和修改卡片

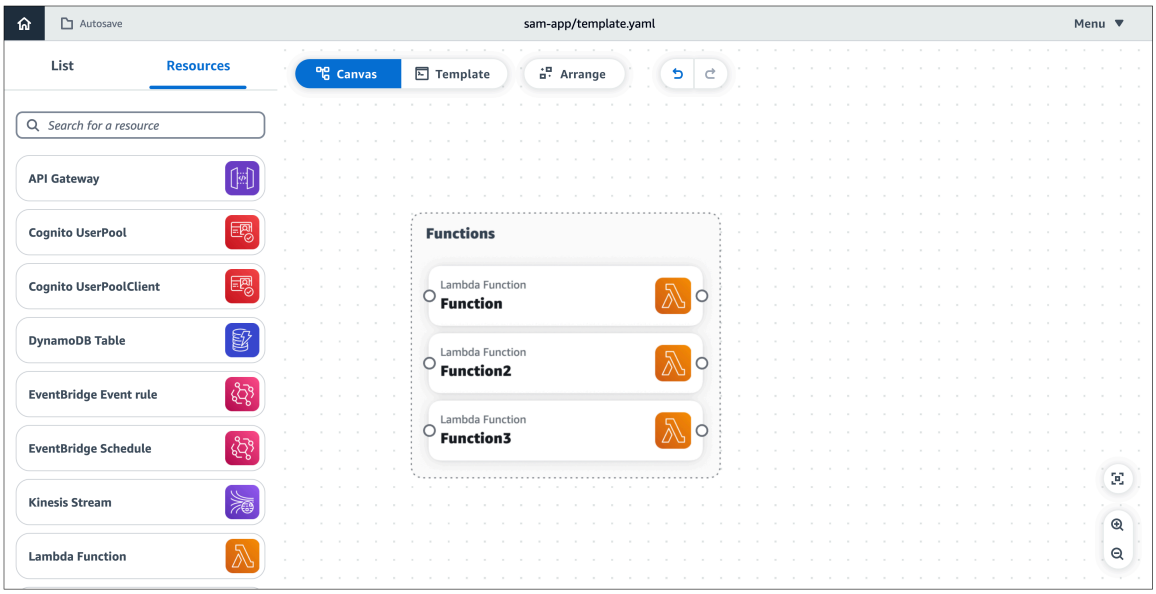
在基础设施编排器的可视画布上将卡片组合在一起

本主题包含有关对增强组件卡和标准组件卡进行分组的详细信息。分组卡片可帮助您对资源进行分类和整理，而无需考虑需要编写的代码或标记。

对增强型组件卡进行分组

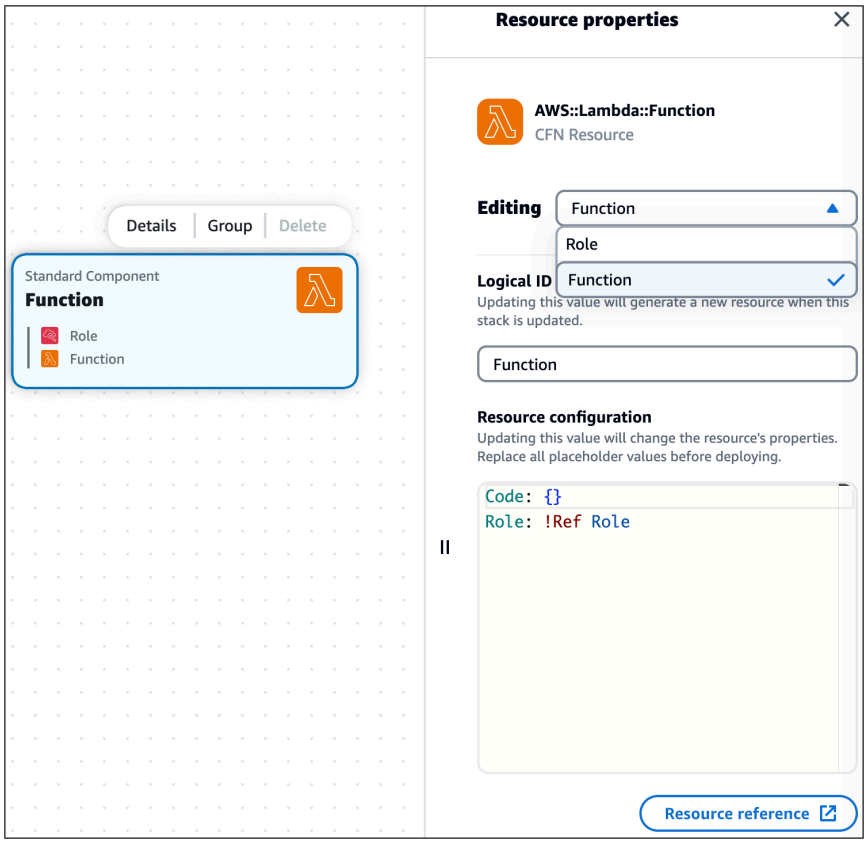
有两种方法可以将增强型组件卡组合在一起：

- 按住 Shift 键的同时，选择要分组的卡片。然后，从资源操作菜单中选择 “分组”。
- 选择要加入群组的卡片。从出现的菜单中选择 “群组”。这将创建一个群组，您可以将其他卡片拖放到该组中。



将标准组件卡分组为另一张

以下示例显示了在资源属性面板中将标准组件卡分组为另一张卡片的一种方式：



在资源属性面板的资源配置字段中，Lambda 函数中引用Role了。这会导致角色卡片分组到画布上的功能卡片中。

在基础设施编辑器的可视化画布上连接卡片

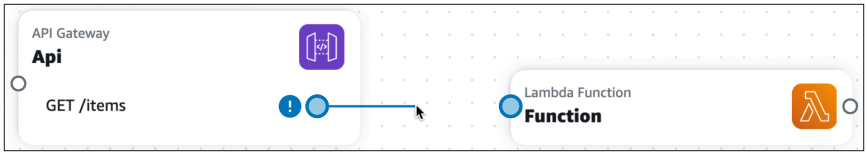
使用本主题来了解如何在基础架构编排器中连接卡片。本节包括有关连接增强型组件卡和标准组件卡的详细信息。它还提供了一些示例，说明了连接卡的不同方式。

连接增强型组件卡

在增强型组件卡上，端口可直观地识别可以连接的位置。

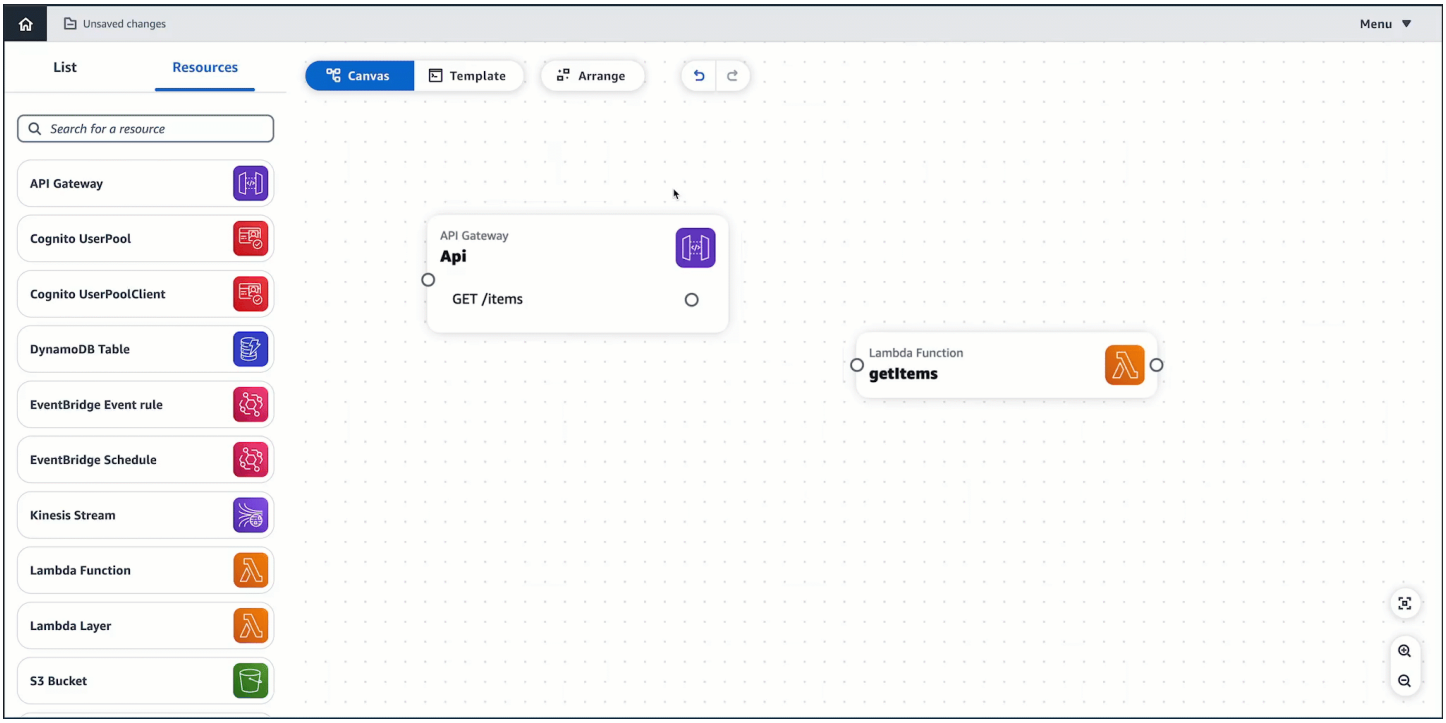
- 卡片右侧的端口表示该卡有机会调用另一张卡。
- 卡片左侧的端口表示该卡有机会被另一张卡调用。

通过单击一张卡的右侧端口，然后将其拖动到另一张卡的左侧端口上，将卡片连接在一起。



创建连接时，将显示一条消息，告知您连接是否已成功建立。选择消息以查看基础架构编排器更改了哪些内容以配置连接。如果连接失败，则可以选择模板视图手动更新基础架构代码以配置连接。

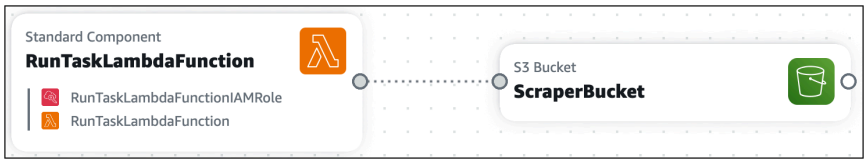
- 成功后，单击消息以查看 “更改” 检查器。在这里，您可以看到基础设施编排器修改了哪些内容以配置您的连接。
- 失败时，将显示一条消息。您可以选择 “模板” 视图并手动更新基础设施代码以配置连接。



当您将增强的组件卡连接在一起时，Infrastructure Composer 会自动在您的模板中创建基础架构代码，以便在资源之间配置事件驱动的关系。

连接标准组件卡（标准 IaC 资源卡）

标准 IaC 资源卡不包括用于与其他资源建立连接的端口。在配置卡片期间，您可以在应用程序的模板中指定事件驱动的关系，Infrastructure Composer 将自动检测这些连接，并在卡片之间用虚线将它们可视化。以下是标准组件卡和增强组件卡之间的连接示例：



以下示例显示了如何将 Lambda 函数与 Amazon API Gateway rest API 连接：

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MyApi:
    Type: 'AWS::ApiGateway::RestApi'
    Properties:
      Name: MyApi

  ApiGatewayMethod:
```

```
    Type: 'AWS::ApiGateway::Method'
    Properties:
      HttpMethod: POST  # Specify the HTTP method you want to use (e.g., GET, POST,
 PUT, DELETE)
      ResourceId: !GetAtt MyApi.RootResourceId
      RestApiId: !Ref MyApi
      AuthorizationType: NONE
      Integration:
        Type: AWS_PROXY
        IntegrationHttpMethod: POST
        Uri: !Sub
          - arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/
${LambdaFunctionArn}/invocations
          - { LambdaFunctionArn: !GetAtt MyLambdaFunction.Arn }
      MethodResponses:
        - StatusCode: 200

  MyLambdaFunction:
    Type: 'AWS::Lambda::Function'
    Properties:
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Runtime: nodejs14.x
      Code:
        S3Bucket: your-bucket-name
        S3Key: your-lambda-zip-file.zip

  LambdaExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: 'sts:AssumeRole'
      Policies:
        - PolicyName: LambdaExecutionPolicy
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
```

```
                - 'logs:CreateLogGroup'
                - 'logs:CreateLogStream'
                - 'logs:PutLogEvents'
              Resource: 'arn:aws:logs:*:*:*'
            - Effect: Allow
              Action:
                - 'lambda:InvokeFunction'
              Resource: !GetAtt MyLambdaFunction.Arn
```

在上面的示例中，下面列出的代码片段Integration:指定了连接两张卡片的事件驱动关系。ApiGatewayMethod:

在基础设施编排器中连接卡片的示例

使用本节中的示例来了解如何在基础设施编排器中连接卡片。

将商品放入亚马逊简单存储服务 (Amazon S3) 存储桶时调用 AWS Lambda 函数

在此示例中，A mazon S3 存储桶卡连接到 Lambda 功能卡。将项目放入 Amazon S3 存储桶后，将调用该函数。然后，该函数可用于处理该项目或触发应用程序中的其他事件。

这种交互需要为函数定义一个事件。以下是基础架构编排器规定的内容：

```
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    ...
  MyBucketBucketPolicy:
    Type: AWS::S3::BucketPolicy
    ...
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Events:
        MyBucket:
          Type: S3
```

```
      Properties:
        Bucket: !Ref MyBucket
        Events:
          - s3:ObjectCreated:* # Event that triggers invocation of function
          - s3:ObjectRemoved:* # Event that triggers invocation of function
```

从 Lambda 函数调用 Amazon S3 存储桶

在此示例中，Lambda 函数卡调用 Amazon S3 存储桶卡。Lambda 函数可用于对 Amazon S3 存储桶中的项目执行 CRUD 操作。

此交互需要以下内容，这些内容由基础架构编排器配置：

- 允许 Lambda 函数与 Amazon S3 存储桶进行交互的 IAM 策略。
- 影响 Lambda 函数行为的环境变量。

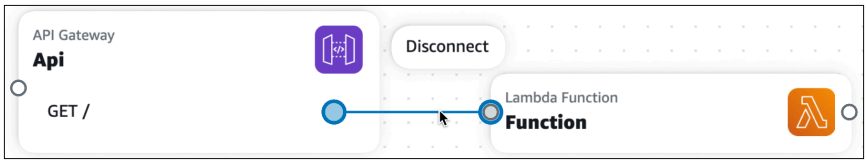
```
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    ...
  MyBucketBucketPolicy:
    Type: AWS::S3::BucketPolicy
    ...
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Environment:
        Variables:
          BUCKET_NAME: !Ref MyBucket
          BUCKET_ARN: !GetAtt MyBucket.Arn
      Policies:
        - S3CrudPolicy:
          BucketName: !Ref MyBucket
```

在基础架构编辑器中断开卡的连接

在 Infrastructure Composer 中，您可以使用增强的组件卡和标准组件卡来连接和断开 AWS 资源。本节介绍如何断开两种类型的卡的连接。

增强的组件卡

要断开增强型组件卡的连接，请选择该线路并选择 “断开连接”。



Infrastructure Composer 将自动修改您的模板，以从您的应用程序中删除事件驱动的关系。

标准组件卡

标准组件卡不包括用于与其他资源建立连接的端口。在配置卡片期间，您可以在应用程序的模板中指定事件驱动的关系，Infrastructure Composer 将自动检测这些连接，并在卡片之间用虚线将它们可视化。要断开标准组件卡的连接，请移除应用程序模板中的事件驱动关系。

以下示例显示了与 Amazon API Gateway rest API 关联的 Lambda 函数：

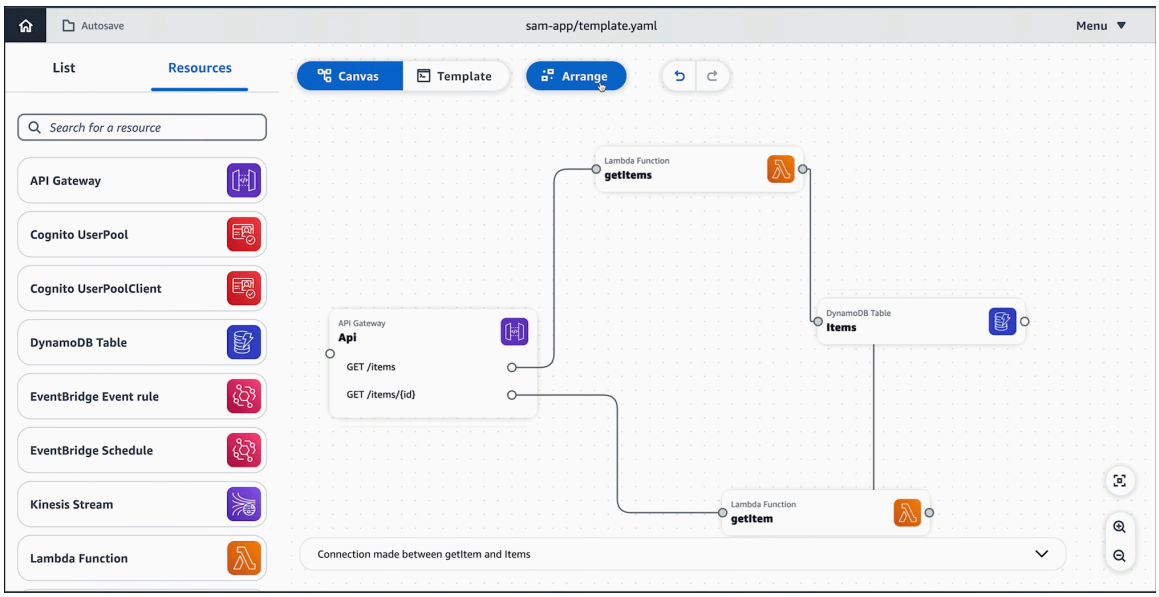
```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  MyApi:
    Type: 'AWS::ApiGateway::RestApi'
    Properties:
      Name: MyApi

  ApiGatewayMethod:
    Type: 'AWS::ApiGateway::Method'
    Properties:
      HttpMethod: POST  # Specify the HTTP method you want to use (e.g., GET, POST,
 PUT, DELETE)
      ResourceId: !GetAtt MyApi.RootResourceId
      RestApiId: !Ref MyApi
      AuthorizationType: NONE
      Integration:
        Type: AWS_PROXY
        IntegrationHttpMethod: POST
        Uri: !Sub
```

```
          - arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/
${LambdaFunctionArn}/invocations
          - { LambdaFunctionArn: !GetAtt MyLambdaFunction.Arn }
      MethodResponses:
        - StatusCode: 200

  MyLambdaFunction:
    Type: 'AWS::Lambda::Function'
    Properties:
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Runtime: nodejs14.x
      Code:
        S3Bucket: your-bucket-name
        S3Key: your-lambda-zip-file.zip

  LambdaExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: 'sts:AssumeRole'
      Policies:
        - PolicyName: LambdaExecutionPolicy
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - 'logs:CreateLogGroup'
                  - 'logs:CreateLogStream'
                  - 'logs:PutLogEvents'
                Resource: 'arn:aws:logs:*:*:*'
              - Effect: Allow
                Action:
                  - 'lambda:InvokeFunction'
                Resource: !GetAtt MyLambdaFunction.Arn
```

要移除两张卡片之间的连接，请移除MyLambdaFunction下面列出的ApiGatewayMethod:内容。Integration

在基础设施编辑器的可视画布上排列卡片

选择 “排列” 可在画布上直观地排列和整理卡片。当画布上有许多卡片和连接时，使用 “排列” 按钮特别有用。



在基础设施编排器中配置和修改卡片

在 Infrastructure Composer 中，卡片代表您用来设计应用程序架构的资源。在 Infrastructure Composer 中配置卡片时，需要定义应用程序中资源的详细信息。这包括卡片的逻辑 ID 和分区密钥等详细信息。这些信息的定义方式因增强型组件卡和标准卡而异。

增强组件卡是一组 CloudFormation 资源，这些资源已组合成单个精选卡片，可增强易用性和功能性，专为各种用例而设计。标准 IaC 资源卡代表单一 AWS CloudFormation 资源。每张标准 IaC 资源卡一旦拖到画布上，就会被标记为标准组件。

本主题提供有关配置增强组件卡和标准组件卡的详细信息。

> **Note**
>
> 本主题适用于使用基础设施编排控制台中的卡片、 AWS Toolkit for Visual Studio Code 扩展模块以及在 Infrastructure Composer 中的 CloudFormation 控制台模式下使用卡片。与

Lambda 相关的卡（Lamb da 函数和 Lambda 层）需要在控制台模式下的 Infrastructure Composer 中不可用的代码构建和打包解决方案。 CloudFormation 有关更多信息，请参阅 在 CloudFormation 控制台模式下使用基础架构编排器。

主题

基础架构编排器中的增强组件卡

要配置增强的组件卡，基础设施编排器在资源属性面板中提供了一个表单。此表单经过精心策划，可指导您完成每张增强组件卡的配置。在您填写表单时，基础设施编排器会修改您的基础架构代码。

一些增强的组件卡确实具有其他功能。本节回顾了使用增强型组件卡的基础知识，并提供了有关具有其他功能的卡片的详细信息。

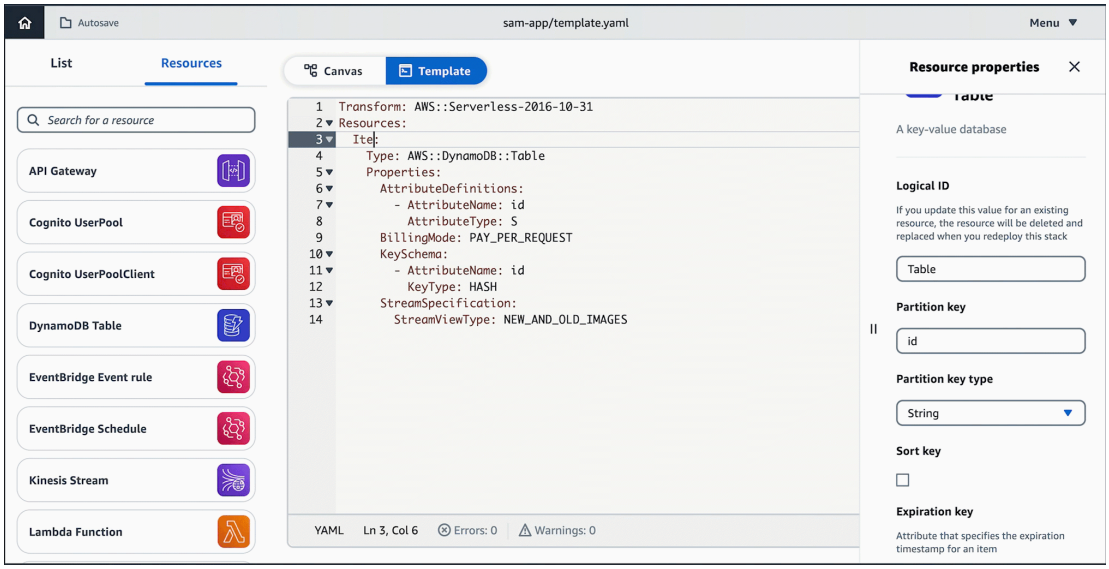
有关增强组件卡的更多信息，请参阅基础架构编排器中的增强组件卡和 基础架构编排器中的增强组件卡

过程

资源属性面板简化了配置，并添加了简化卡配置的指南。要使用此面板，请执行以下步骤：

1. 双击卡片以打开资源属性面板。
2. 单击卡片并选择详细信息以打开资源属性面板。
3. 对于中的基础架构编排器 AWS 管理控制台，选择模板以显示您的应用程序代码。直接从此处进行配置。

   下图显示了如何做到这一点：



将 Infrastructure Composer 与亚马逊关系数据库服务 (Amazon RDS) 配合使用

AWS 基础架构编辑器 具有与亚马逊关系数据库服务（Amazon RDS）的集成。使用 Infrastructure Composer 中的 RD S 数据库（外部）增强组件卡，您可以将应用程序连接到在另一个 CloudFormation 或 AWS Serverless Application Model (AWS SAM) 模板上定义的 Amazon RDS DB 集群、实例和代理。

RDS 数据库（外部）增强型组件卡片表示在其他模板上定义的 Amazon RDS 资源。这包括：

- 在其他模板上定义的 Amazon RDS DB 集群或实例
- 亚马逊 RDS DB 代理

RDS 数据库（外部）增强型组件卡可从资源选项板中获得。

要使用此卡片，请将其拖到基础设施编排器画布上，对其进行配置，然后将其连接到其他资源。

您可以通过 Lambda 函数将您的应用程序连接到外部 Amazon RDS DB 集群或实例。

要求

要使用此功能，必须满足以下要求：

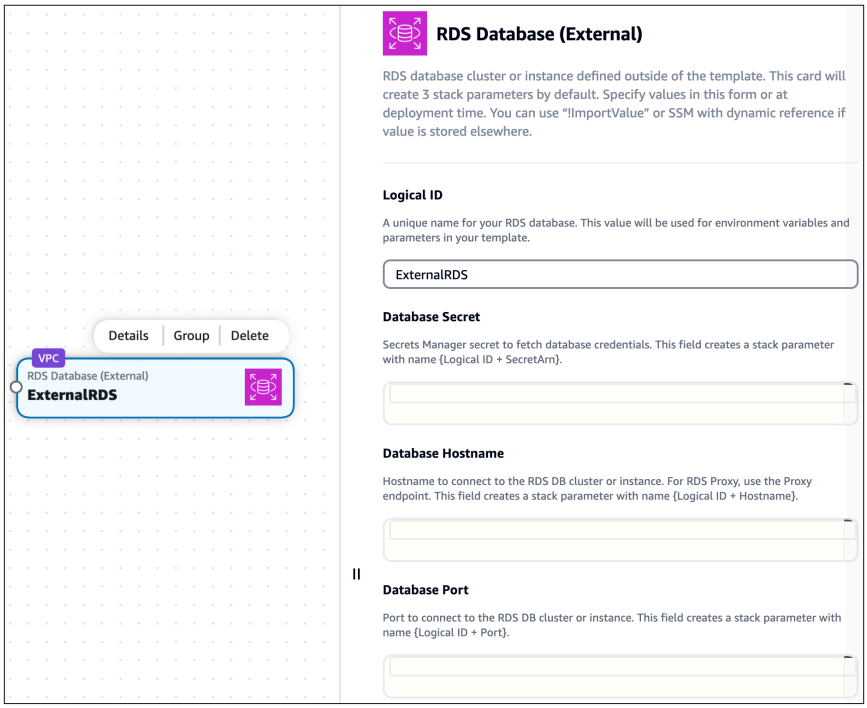
1. 您的外部 Amazon RDS DB 集群、实例或代理必须使用 AWS Secrets Manager 来管理用户密码。要了解更多信息，请参阅使用 Amazon RDS 管理密码和 AWS Secrets Manager Amazon RDS 用户指南。
2. 您在基础架构编排器中的应用程序必须是新项目，或者必须是最初在基础设施编排器中创建的。

过程

步骤 1：配置外部 RDS 数据库卡

从 “资源” 选项板中，将 RDS 数据库（外部）增强型组件卡片拖到画布上。

选择卡片并选择 “详细信息”，或者双击卡片以打开 “资源属性” 面板。卡片的资源属性面板将出现：



您可以在此处配置以下内容：

- 逻辑 ID — 外部 Amazon RDS DB 集群、实例或代理的唯一名称。此 ID 不必与您的外部 Amazon RDS DB 资源的逻辑 ID 值相匹配。
- 数据库密钥 — 与您的 Amazon RDS DB 集群、实例或代理关联的 AWS Secrets Manager 密钥的标识符。此字段接受以下值：
  - 静态值-数据库密钥的唯一标识符，例如机密 ARN。以下是示例：`arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-name-1a2b3c`有关更多信息，请参阅《AWS Secrets Manager 开发人员指南》中的 AWS Secrets Manager 概念。

  - 输出值-将 Secrets Manager 密钥部署到时 AWS CloudFormation，将创建一个输出值。您可以使用Fn::ImportValue内部函数在此处指定输出值。例如 !ImportValue MySecret。
  - 来自 SSM 参数存储区的值 — 您可以将您的密钥存储在 SSM 参数存储库中，并使用动态引用指定其值。例如 {{resolve:ssm:MySecret}}。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的 SSM 参数。
- 数据库主机名-可用于连接您的 Amazon RDS DB 集群、实例或代理的主机名。此值是在定义您的 Amazon RDS 资源的外部模板中指定的。接受以下值：
  - 静态值-数据库主机名的唯一标识符，例如端点地址。以下是示例：mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-2.rds.amazonaws.com
  - 输出值-已部署的 Amazon RDS DB 集群、实例或代理的输出值。您可以使用Fn::ImportValue内部函数指定输出值。例如 !ImportValue myStack-myDatabase-abcd1234。
  - 来自 SSM 参数存储区的值 — 您可以将数据库主机名存储在 SSM 参数存储中，并使用动态引用指定其值。例如 {{resolve:ssm:MyDatabase}}。
- 数据库端口-可用于连接您的 Amazon RDS DB 集群、实例或代理的端口号。此值是在定义您的 Amazon RDS 资源的外部模板中指定的。接受以下值：
  - 静态值-数据库端口。例如 3306。
  - 输出值-已部署的 Amazon RDS DB 集群、实例或代理的输出值。例如 !ImportValue myStack-MyRDSInstancePort。
  - 来自 SSM 参数存储的值-您可以将数据库主机名存储在 SSM 参数存储中，并使用动态引用指定其值。例如 {{resolve:ssm:MyRDSInstancePort}}。

> **Note**
>
> 必须在此处配置逻辑 ID 值。如果您愿意，可以在部署时配置其他属性。

第 2 步：连接 Lambda 函数卡

从“资源”面板中，将 Lambda 函数增强型组件卡片拖到画布上。

将 Lambda 函数卡的左侧端口连接到 RDS 数据库（外部）卡的右侧端口。

基础架构编排器将配置您的模板以促进此连接。

基础架构编排器为创建连接所做的工作

完成上面列出的步骤后，基础设施编排器会执行特定操作将您的 Lambda 函数连接到您的数据库。

指定外部 Amazon RDS DB 集群、实例或代理时

当您将 RDS 数据库（外部）卡片拖到画布上时，基础设施编排器会根据需要更新模板的`Metadata`和`Parameters`部分。以下是示例：

```
Metadata:
  AWS::Composer::ExternalResources:
    ExternalRDS:
      Type: externalRDS
      Settings:
        Port: !Ref ExternalRDSPort
        Hostname: !Ref ExternalRDSHostname
        SecretArn: !Ref ExternalRDSSecretArn
Parameters:
  ExternalRDSPort:
    Type: Number
  ExternalRDSHostname:
    Type: String
  ExternalRDSSecretArn:
    Type: String
```

元数据是一个 CloudFormation 模板部分，用于存储有关您的模板的详细信息。基础架构编排器特有的元数据存储在`AWS::Composer::ExternalResources`元数据密钥下。在这里，Infrastructure Composer 存储您为 Amazon RDS DB 集群、实例或代理指定的值。

CloudFormation 模板的 “参数” 部分用于存储自定义值，这些值可以在部署时插入到整个模板中。根据您提供的值类型，Infrastructure Composer 可能会在此处存储您的 Amazon RDS DB 集群、实例或代理的值，并在整个模板中指定这些值。

`Metadata`和`Parameters`部分中的字符串值使用您在 RDS 数据库（外部）卡上指定的逻辑 ID 值。如果更新逻辑 ID，则字符串值将发生变化。

将 Lambda 函数连接到您的数据库时

当您将 Lambda 函数卡连接到 RDS 数据库（外部）卡时，基础设施编排器会配置环境变量和 AWS Identity and Access Management (IAM) 策略。以下是示例：

```
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Environment:
        Variables:
          EXTERNALRDS_PORT: !Ref ExternalRDSPort
          EXTERNALRDS_HOSTNAME: !Ref ExternalRDSHostname
          EXTERNALRDS_SECRETARN: !Ref ExternalRDSSecretArn
      Policies:
        - AWSSecretsManagerGetSecretValuePolicy:
            SecretArn: !Ref ExternalRDSSecretArn
```

环境变量是您的函数可以在运行时使用的变量。要了解更多信息，请参阅AWS Lambda 开发人员指南中的使用 Lambda 环境变量。

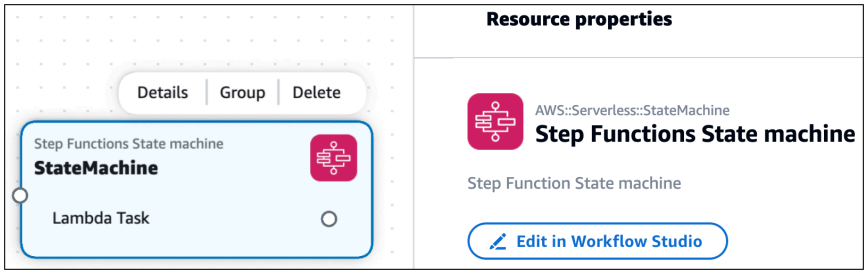
策略为您的函数提供权限。在这里，Infrastructure Composer 创建了一个策略，允许您的函数对 Secrets Manager 进行读取访问以获取您访问 Amazon RDS DB 集群、实例或代理的密码。

AWS 基础架构编辑器 与一起使用 AWS Step Functions

AWS 基础架构编辑器 具有与集成的功能AWS Step FunctionsWorkflow Studio。使用基础设施编排器执行以下操作：

- Workflow Studio直接在基础设施编排器中启动 Step Functions。
- 创建和管理新工作流程或将现有工作流导入基础架构编排器。
- 使用基础设施编排器画布将您的工作流程与其他 AWS 资源集成。

下图是 Step Functions 状态机卡



借助 Infrastructure Composer Workflow Studio 中的 Step Functions，你可以在一个地方使用两个强大的视觉设计器的优势。在您设计工作流程和应用程序时，Infrastructure Composer 会创建您的基础设施即代码 (IaC)，以指导您进行部署。

主题

- IAM 策略
- 基础架构 Composer Workflow Studio 中的 Step Functions 入门
- Workflow Studio在基础设施编排器中使用 Step Functions
- 了解详情

IAM 策略

当您将工作流程中的任务连接到资源时，Infrastructure Composer 会自动创建授权资源之间交互所需的 AWS Identity and Access Management (IAM) 策略。以下是示例：

```
Transform: AWS::Serverless-2016-10-31
Resources:
  StockTradingStateMachine:
    Type: AWS::Serverless::StateMachine
    Properties:
      ...
      Policies:
        - LambdaInvokePolicy:
            FunctionName: !Ref CheckStockValue
      ...
  CheckStockValue:
    Type: AWS::Serverless::Function
    ...
```

如有必要，您可以向模板添加更多 IAM 策略。

基础架构 Composer Workflow Studio 中的 Step Functions 入门

首先，您可以创建新的工作流程或导入现有工作流程。

创建新工作流程

1. 从 “资源” 面板中，将 Step Functions 状态机增强型组件卡片拖到画布上。

当你将 Step Functions 状态机卡拖到画布上时，Infrastructure Composer 会创建以下内容：

- 定义状态机的 `AWS::Serverless::StateMachine`资源。默认情况下，基础架构编排器创建标准工作流程。要创建快速工作流程，请将模板中的`Type`值从更改`STANDARD`为`EXPRESS`。
- 一种为状态机定义 Amazon CloudWatch 日志组的`AWS::Logs::LogGroup`资源。

2. 打开卡片的资源属性面板，然后选择在 Workfl ow Studio 中编辑，在基础架构编辑器Workflow Studio中打开。

   Step Fun Workflow Studio ctions 将在设计模式下打开。要了解更多信息，请参阅《AWS Step Functions 开发者指南》中的设计模式。

   > **Note**
   >
   > 您可以修改基础架构编排器以将状态机定义保存在外部文件中。要了解更多信息，请参阅处理外部文件。

3. 创建您的工作流程并选择 “保存”。要退出Workflow Studio，请选择返回到基础架构编排器。

   基础架构编排器使用`AWS::Serverless::StateMachine`资源的`Defintion`属性定义您的工作流程。

4. 您可以通过执行以下任一操作来修改工作流程：

   - Workflow Studio再次打开并修改您的工作流程。
   - 对于控制台中的 Infrastructure Composer，您可以打开应用程序的模板视图并修改您的模板。如果使用本地同步，则可以在本地 IDE 中修改工作流程。基础设施编排器将在基础设施编排器中检测您的更改并更新您的工作流程。
   - 对于 VS Code Toolkit for VS Code 中的基础设施编排器，您可以直接修改您的模板。基础设施编排器将在基础设施编排器中检测您的更改并更新您的工作流程。

导入现有工作流程

您可以从使用 AWS Serverless Application Model (AWS SAM) 模板定义的应用程序中导入工作流。使用任何用`AWS::Serverless::StateMachine`资源类型定义的状态机，它将可视化为 Step Functions 状态机增强的组件卡，你可以用它来启动Workflow Studio。

该AWS::Serverless::StateMachine资源可以使用以下任一属性来定义工作流程：

- Definition— 工作流在 AWS SAM 模板中定义为一个对象。
- DefinitionUri— 该工作流程是使用亚马逊州语言在外部文件上定义的。然后使用此属性指定文件的本地路径。

定义属性

控制台中的基础架构编译器

对于使用该Definition属性定义的工作流程，您可以导入单个模板或整个项目。

- 模板-有关导入模板的说明，请参阅在基础设施编排控制台中导入现有项目模板。要保存您在基础设施编排器中所做的更改，必须导出您的模板。
- 项目-导入项目时，必须激活本地同步。您所做的更改会自动保存到本地计算机中。有关导入项目的说明，请参阅在基础设施编排控制台中导入现有项目文件夹。

来自 VS Code 的 Toolkit 中的基础设施编译器

对于使用该Definition属性定义的工作流，您可以从模板中打开 “基础设施编排”。有关说明，请参阅从访问基础架构编排器 AWS Toolkit for Visual Studio Code。

DefinitionUri 财产

控制台中的基础架构编译器

对于使用该DefinitionUri属性定义的工作流程，必须导入项目并激活本地同步。有关导入项目的说明，请参阅在基础设施编排控制台中导入现有项目文件夹。

来自 VS Code 的 Toolkit 中的基础设施编译器

对于使用该DefinitionUri属性定义的工作流，您可以从模板中打开 “基础设施编排”。有关说明，请参阅从访问基础架构编排器 AWS Toolkit for Visual Studio Code。

Workflow Studio在基础设施编排器中使用 Step Functions

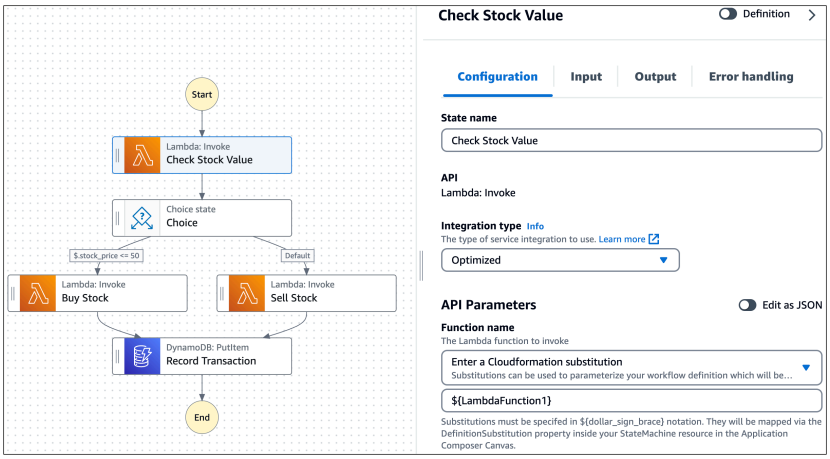
构建工作流程

Infrastructure Composer 使用定义替换将工作流任务映射到应用程序中的资源。要了解有关定义替换的更多信息，请参阅AWS Serverless Application Model 开发者指南*DefinitionSubstitutions*中的。

在中创建任务时Workflow Studio，请为每个任务指定一个定义替换。然后，您可以将任务连接到基础设施编排器画布上的资源。

要在中指定定义替换 Workflow Studio

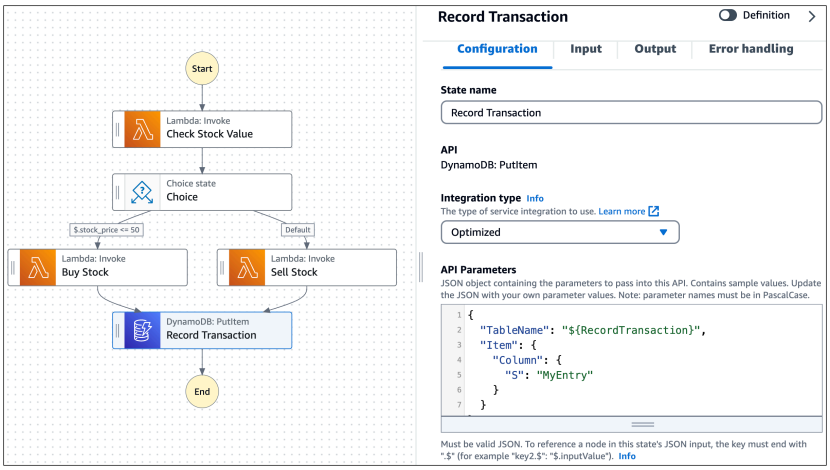
1. 打开任务的 “配置” 选项卡，找到 “API 参数” 字段。



2. 如果 “API 参数” 字段有下拉选项，请选择 “输入 CloudFormation 替换”。然后，提供一个唯一的名称。

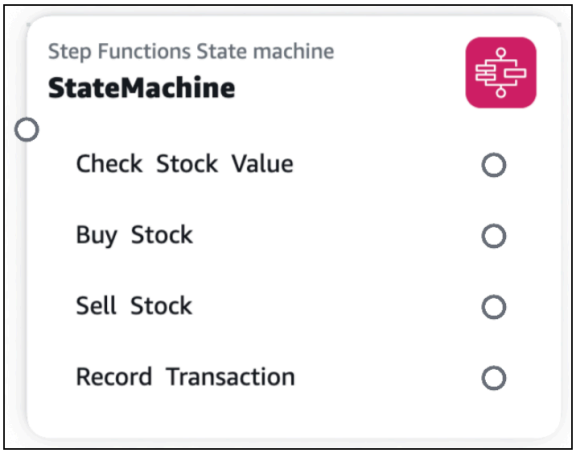
   对于连接到相同资源的任务，请为每个任务指定相同的定义替换。要使用现有的定义替换，请选择 “选择 CloudFormation 替代”，然后选择要使用的替代。

3. 如果 API 参数字段包含 JSON 对象，请修改指定资源名称的条目以使用定义替换。在以下示例中，我们更改`"MyDynamoDBTable"`为`"${RecordTransaction}"`。



4. 选择保存并返回到基础架构编排器。

工作流程中的任务将在 Ste p Functions 状态机卡上可视化。



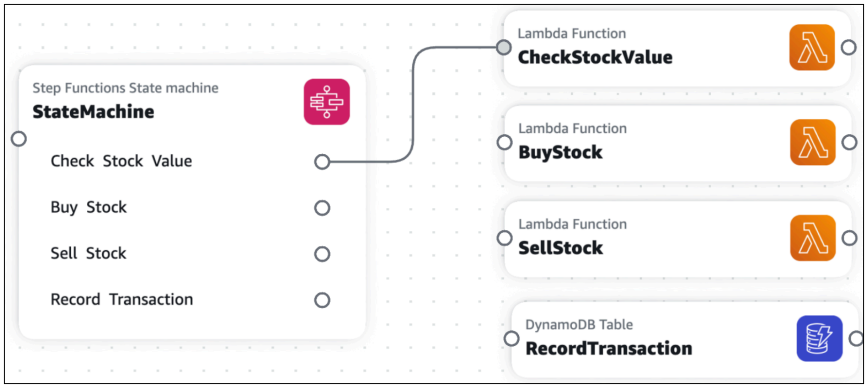
Connect 将资源连接到工作流程任务

你可以在基础架构编排器中创建支持的工作流任务和支持的基础设施编排卡片之间的连接。

- 支持的工作流任务-针对 Step F AWS 服务 unctions 进行了优化的任务。要了解更多信息，请参阅《AWS Step Functions 开发者指南》中的 Step Functions 优化集成。
- 支持的基础设施合成器卡-支持增强型组件卡。要了解有关基础设施编排器中卡片的更多信息，请参阅在基础设施编排器中配置和修改卡片。

创建连接时，任务和卡片 AWS 服务 的必须匹配。例如，您可以将调用 Lambda 函数的工作流程任务连接到 Lambda 函数增强型组件卡。

要创建连接，请单击任务的端口并将其拖动到增强组件卡的左侧端口。



基础架构 Composer 将自动更新您的`DefinitionSubstitution`值以定义您的连接。以下是示例：

```
Transform: AWS::Serverless-2016-10-31
Resources:
  StateMachine:
```

```
    Type: AWS::Serverless::StateMachine
    Properties:
      Definition:
        StartAt: Check Stock Value
        States:
          Check Stock Value:
            Type: Task
            Resource: arn:aws:states:::lambda:invoke
            Parameters:
              Payload.$: $
              FunctionName: ${CheckStockValue}
            Next: Choice
          ...
      DefinitionSubstitutions:
        CheckStockValue: !GetAtt CheckStockValue.Arn
        ...
  CheckStockValue:
    Type: AWS::Serverless::Function
    Properties:
      ...
```

处理外部文件

当您通过 Step Functions 状态机卡创建工作流程时，Infrastructure Composer 会使用该`Definition`属性将状态机定义保存在模板中。您可以将基础架构编排器配置为将状态机定义保存在外部文件中。

> **Note**
>
> 要将此功能与中的 Infrastructure Composer 配合使用 AWS 管理控制台，必须激活本地同步。有关更多信息，请参阅 在基础设施编排控制台中本地同步并保存您的项目。

将状态机定义保存在外部文件上

1. 打开 Ste p Functions 状态机卡的资源属性面板。
2. 选择 “使用外部文件定义状态机” 选项。
3. 为您的状态机定义文件提供相对路径和名称。
4. 选择保存。

基础架构编排器将执行以下操作：

1. 将状态机定义从`Definition`字段移至外部文件。
2. 使用 Amazon States 语言将状态机定义保存在外部文件中。
3. 使用`DefinitionUri`字段修改模板以引用外部文件。

了解详情

要了解有关基础架构编排器中的 Step Functions 的更多信息，请参阅以下内容：

- Workflow Studio在《AWS Step Functions 开发人员指南》的《基础设施编排器》中使用。
- DefinitionSubstitutions 在《AWS Step Functions 开发人员指南》的 AWS SAM 模板中。

基础架构编辑器中的标准卡

所有 CloudFormation 资源都可用作 “资源” 选项板中的标准 IaC 资源卡。拖到视觉画布上后，标准 IaC 资源卡将变成标准组件卡。这只是意味着该卡是一个或多个标准 IaC 资源。有关更多示例和详细信息，请参阅本节的主题。

您可以通过 “模板” 视图和 “资源属性” 窗口修改基础架构代码。例如，以下是`Alexa::ASK::Skill`标准 IaC 资源的起始模板示例：

```
Resources:
  Skill:
    Type: Alexa::ASK::Skill
    Properties:
      AuthenticationConfiguration:
        RefreshToken: <String>
        ClientSecret: <String>
        ClientId: <String>
      VendorId: <String>
      SkillPackage:
        S3Bucket: <String>
        S3Key: <String>
```

标准 IaC 资源卡起始模板包含以下内容：

- CloudFormation 资源类型。
- 必填或常用属性。

- 为每个属性提供的所需值类型。

> **Note**
>
> 您可以使用Amazon Q为标准资源卡生成基础架构代码建议。要了解更多信息，请参阅AWS 基础架构编辑器 与一起使用 Amazon Q Developer。

过程

您可以通过资源属性面板修改标准组件卡片中每种资源的基础设施代码。

修改标准组件卡

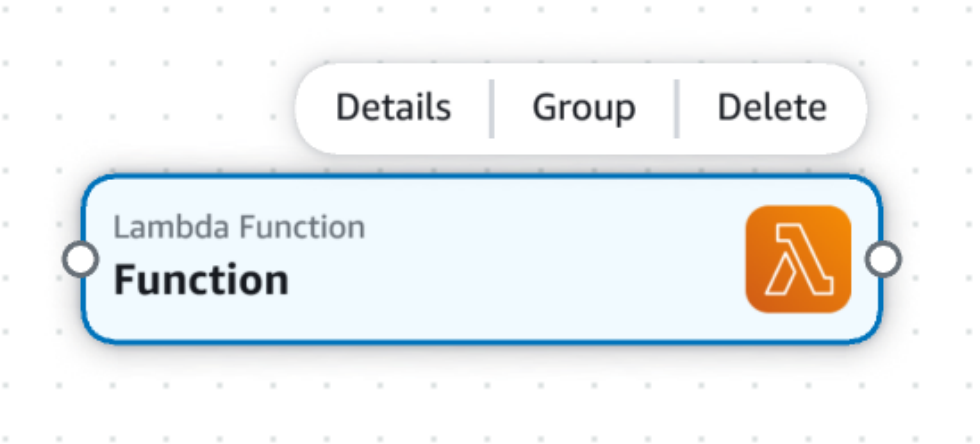
1. 打开标准 IaC 组件卡的资源属性面板。
2. 在 “编辑” 字段中，从下拉列表中选择要编辑的标准 IaC 资源。
3. 修改您的基础设施代码并保存。

在基础设施编辑器中删除卡片

本节提供有关在中删除卡片的说明 AWS 基础架构编辑器。

增强的组件卡

要删除增强的组件卡，请选择您在视觉画布上放置的卡片。从 “卡片操作” 菜单中，选择 “删除”。



标准组件卡

要删除标准组件卡，必须手动从模板中删除每个 CloudFormation 资源的基础设施代码。以下是实现此目的的简单方法：

1. 记下要删除的资源的逻辑 ID。
2. 在模板上，通过`Resources`或`Outputs`部分中的逻辑 ID 找到资源。
3. 从模板中删除资源。这包括资源逻辑 ID 及其嵌套值，例如`Type`和`Properties`。
4. 检查 “画布” 视图，确认该资源已从画布中移除。

使用基础架构编排器中的变更检查器查看代码更新

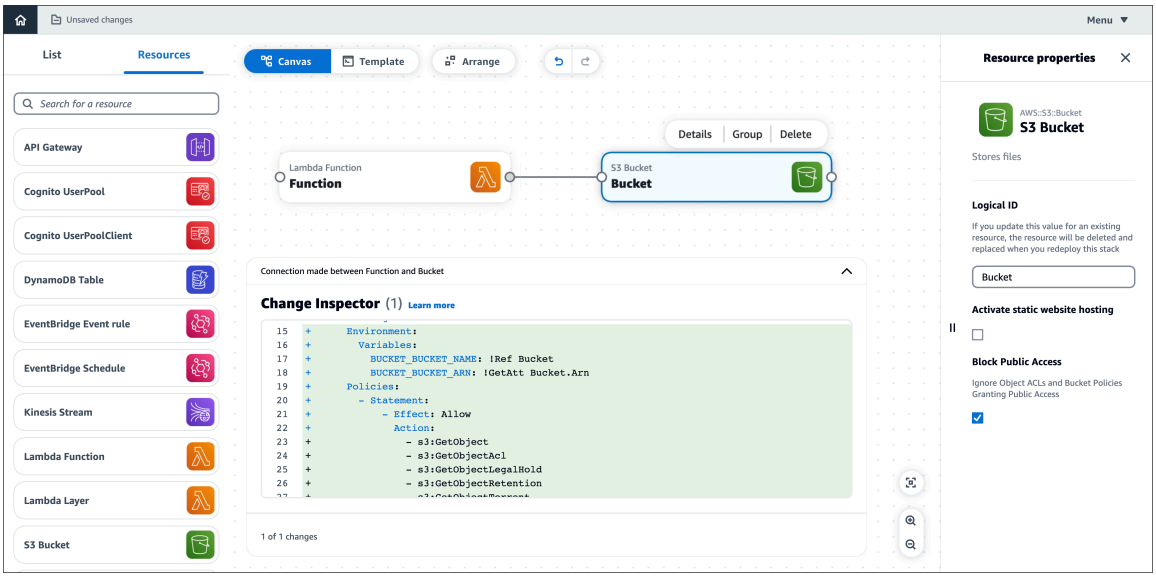
当您在基础架构编排控制台中进行设计时，您的基础架构代码会自动创建。使用 Change Inspec tor 查看您的模板代码更新，并了解 Infrastructure Composer 正在为您创建什么。

本主题介绍如何使用 AWS 管理控制台 或 AWS Toolkit for Visual Studio Code 扩展中的基础架构编排器。

Change Inspector 是 In frastructure Composer 中的一个可视化工具，可向您显示

- 在设计应用程序时，消息会显示在视觉画布的底部。这些消息提供了对您正在执行的操作的评论。
- 如果支持，您可以展开消息以查看 Change Inspector。
- Change Inspector 会显示您最近一次交互中的代码更改。

以下示例演示了变更检查器的工作原理：



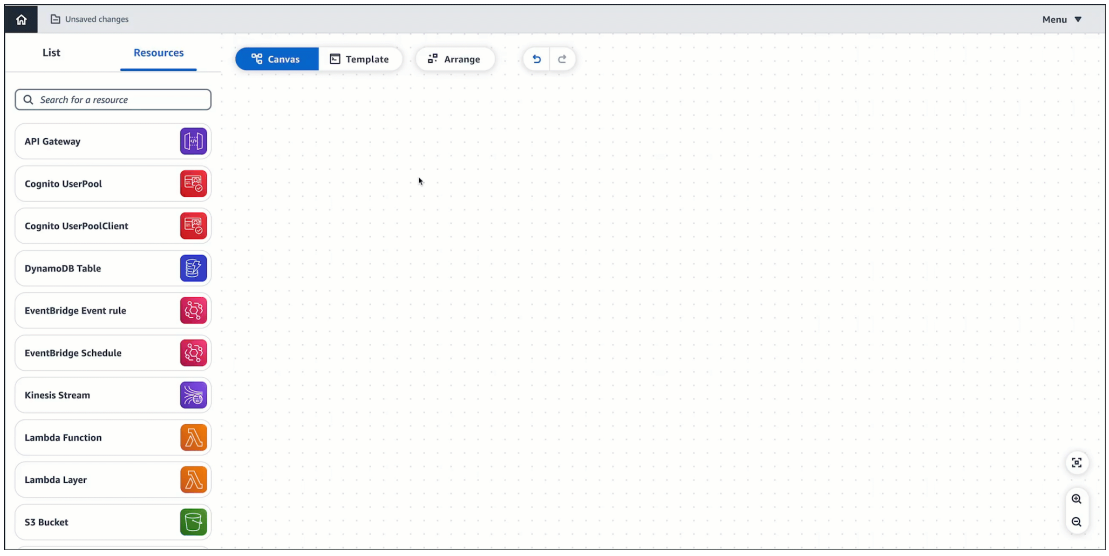
Change Inspector 的好处

Change Inspector 是查看基础设施编排器为您创建的模板代码的好方法。这也是学习如何编写基础架构代码的好方法。在 Infrastructure Composer 中设计应用程序时，请在 Change Inspec tor 中查看代码更新，以了解配置设计所需的代码。

过程

使用 Change Inspector

1. 展开一条消息以打开 Change Inspector。



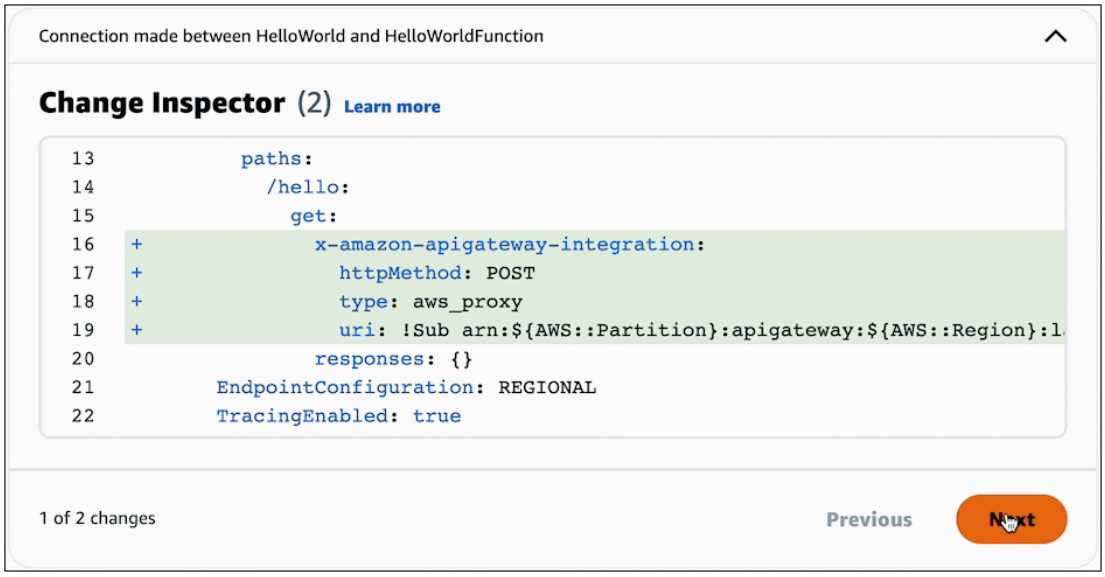
2. 查看自动为您编写的代码。

a. 绿色突出显示的代码表示新添加的代码。

b. 红色突出显示的代码表示新删除的代码。

c. 行号表示模板中的位置。

3. 更新模板的多个部分后，Change Inspector 会对其进行整理。选择 “上一页” 和 “下一页” 按钮以查看所有更改。



> **Note**
>
> 对于控制台中的 Infrastructure Composer，您可以使用模板视图在整个模板的上下文中查看代码更改。您还可以将 Infrastructure Composer 与本地 IDE 同步，并在本地计算机上查看整个模板。要了解更多信息，请参阅将基础架构编排控制台与本地 IDE 连接起来。

了解详情

有关基础架构编排器创建的代码的更多信息，请参阅以下内容：

- 基础架构编辑器中的卡片连接.

在基础架构编辑器中引用外部文件

您可以将外部文件与 AWS Serverless Application Model (AWS SAM) 模板配合使用，以重复使用代码并组织项目。例如，您可能有多个由OpenAPI规范描述的 Amazon API Gateway REST API 资源。您可以创建一个外部文件并为每个资源引用该文件，而不必在模板中复制OpenAPI规范代码。

AWS 基础架构编辑器 支持以下外部文件用例：

- 由外部OpenAPI规范文件定义的 API Gateway REST API 资源。
- AWS Step Functions 由外部状态机定义文件定义的状态机资源。

要了解有关为支持的资源配置外部文件的更多信息，请参阅以下内容：

- 用于 `AWS::Serverless::Api` 的 `DefinitionBody`。
- 用于 `AWS::Serverless::StateMachine` 的 `DefinitionUri`。

> **Note**
>
> 要通过基础设施编排控制台使用基础设施编排器引用外部文件，您必须在本地同步模式下使用基础设施编排器。有关更多信息，请参阅 在基础设施编排控制台中本地同步并保存您的项目。

主题

- 基础设施编排器外部参考文件的最佳实践

基础设施编排器外部参考文件的最佳实践

在本地 IDE 中使用基础架构编排器

在本地同步模式下将基础设施编排器与本地 IDE 配合使用时，可以使用本地 IDE 来查看和修改外部文件。您的模板中引用的受支持的外部文件中的内容将在基础架构编辑器画布中自动更新。要了解更多信息，请参阅将基础架构编排控制台与本地 IDE 连接起来。

将外部文件保存在项目的父目录中

可以在项目的父目录中创建子目录来整理外部文件。Infrastructure Composer 无法访问存储在项目父目录之外的目录中的外部文件。

使用部署您的应用程序 AWS SAMCLI

将应用程序部署到时 AWS 云，需要先将本地外部文件上传到可访问的位置，例如亚马逊简单存储服务 (Amazon S3) Service。您可以使用 AWS SAM CLI 自动简化此过程。要了解更多信息，请参阅《AWS Serverless Application Model 开发人员指南》中的部署时上传本地文件。

在基础设施编排器中创建外部文件引用

您可以从支持的资源的资源属性面板中创建外部文件引用。

创建外部文件引用

1. 从 API Gateway 或 Step Fun ctions 增强组件卡中，选择详细信息以打开资源属性面板。
2. 找到并选择 “使用外部文件” 选项。
3. 指定外部文件的相对路径。这是从您的`template.yaml`文件到外部文件的路径。

   例如，要引用以下项目结构中的`api-spec.yaml`外部文件，请指定`./api-spec.yaml`为相对路径。

   ```
   demo
   ```

```
### api-spec.yaml
### src
# ### Function
# ### index.js
# ### package.json
### template.yaml
```

> **Note**
>
> 如果外部文件及其指定路径不存在，则基础架构编排器将创建该文件。

4. 保存您的更改。

在基础架构编辑器中加载带有外部文件引用的项目

按照本页列出的步骤加载带有外部文件引用的基础架构编排器项目。

从基础设施编排控制台中

1. 完成 在基础设施编排控制台中导入现有项目模板 中列出的步骤。
2. 确认基础架构 Composer 会提示你连接到项目的根文件夹

如果您的浏览器支持 File System Access API，Infrastructure Composer 将提示您连接到项目的根文件夹。Infrastructure Composer 将在本地同步模式下打开您的项目，以支持您的外部文件。如果不支持引用的外部文件，您将收到一条错误消息。有关错误消息的更多信息，请参阅问题排查。

来自 VS Code 的 Toolkit for

1. 完成 从访问基础架构编排器 AWS Toolkit for Visual Studio Code 中列出的步骤。
2. 在基础设施编辑器中打开要查看的模板。

当您从模板访问基础设施编排器时，基础设施编排器将自动检测您的外部文件。如果不支持引用的外部文件，您将收到一条错误消息。有关错误消息的更多信息，请参阅问题排查。

在基础架构编排器中创建引用外部文件的应用程序

此示例使用创建了一个引用外部文件作为其状态机定义的应用程序。 AWS SAMCLI然后，在基础架构编排器中加载项目，并正确引用外部文件。

示例

1. 首先，使用 AWS SAM CLI sam init 命令初始化名为的新应用程序demo。在交互流程中，选择多步骤工作流快速入门模板。

```
$ sam init

...

Which template source would you like to use?
        1 - AWS Quick Start Templates
        2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
        1 - Hello World Example
        2 - Multi-step workflow
        3 - Serverless API
        4 - Scheduled task
        ...
Template: 2

Which runtime would you like to use?
        1 - dotnet6
        2 - dotnetcore3.1
        ...
        15 - python3.7
        16 - python3.10
        17 - ruby2.7
Runtime: 16

Based on your selections, the only Package type available is Zip.
We will proceed to selecting the Package type as Zip.

Based on your selections, the only dependency manager available is pip.
We will proceed copying the template using pip.

Would you like to enable X-Ray tracing on the function(s) in your application?  [y/
N]: ENTER

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view https://docs.aws.amazon.com/AmazonCloudWatch/latest/
monitoring/cloudwatch-application-insights.html [y/N]: ENTER
```

```
Project name [sam-app]: demo

    -----------------------
    Generating application:
    -----------------------
    Name: demo
    Runtime: python3.10
    Architectures: x86_64
    Dependency Manager: pip
    Application Template: step-functions-sample-app
    Output Directory: .
    Configuration file: demo/samconfig.toml

    Next steps can be found in the README file at demo/README.md

...
```

此应用程序引用状态机定义的外部文件。

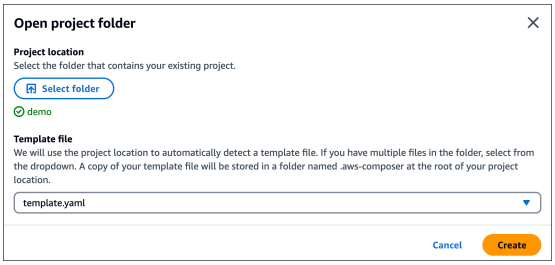
```
...
Resources:
  StockTradingStateMachine:
    Type: AWS::Serverless::StateMachine
    Properties:
      DefinitionUri: statemachine/stock_trader.asl.json
...
```

外部文件位于我们应用程序的statemachine子目录中。

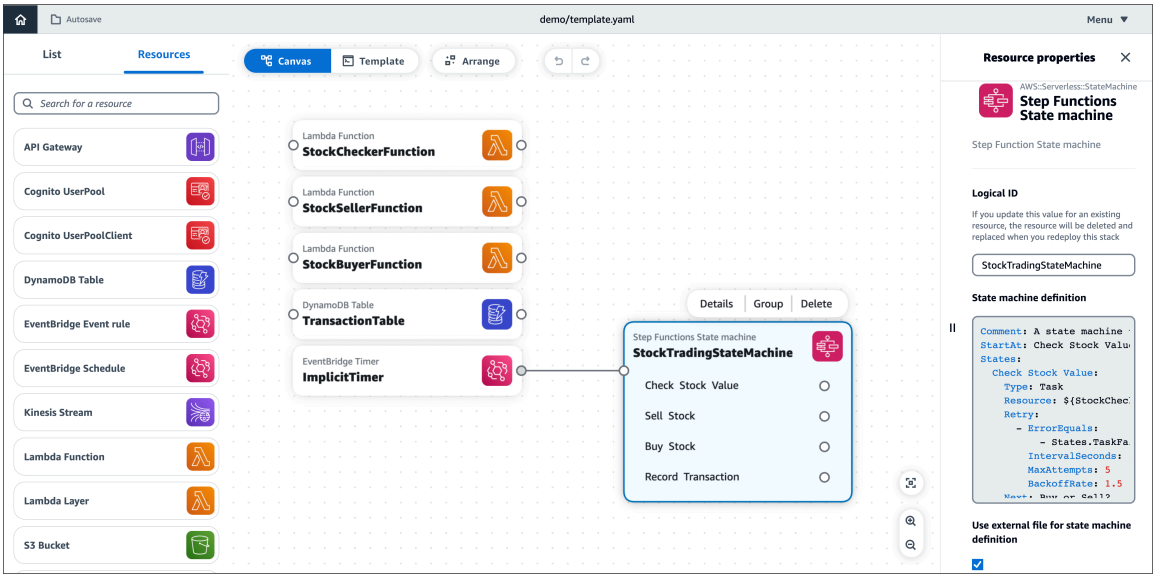
```
demo
### README.md
### __init__.py
### functions
#   ### __init__.py
#   ### stock_buyer
#   ### stock_checker
#   ### stock_seller
### samconfig.toml
### statemachine
#   ### stock_trader.asl.json
### template.yaml
```

```
### tests
```

2. 接下来，从控制台在基础设施编排器中加载您的应用程序。在基础设施编排器主页上，选择加载 CloudFormation 模板。
3. 选择我们的demo项目文件夹，然后允许提示查看文件。选择我们的`template.yaml`文件并选择“创建”。出现提示时，选择保存更改。



基础架构 Composer 会自动检测外部状态机定义文件并将其加载。选择我们的StockTradingStateMachine资源并选择详细信息以显示资源属性面板。在这里，您可以看到基础设施编排器已自动连接到我们的外部状态机定义文件。



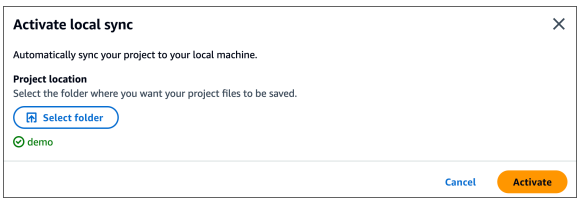
对状态机定义文件所做的任何更改都将自动反映在基础设施编排器中。

使用基础设施编排器引用OpenAPI规范外部文件

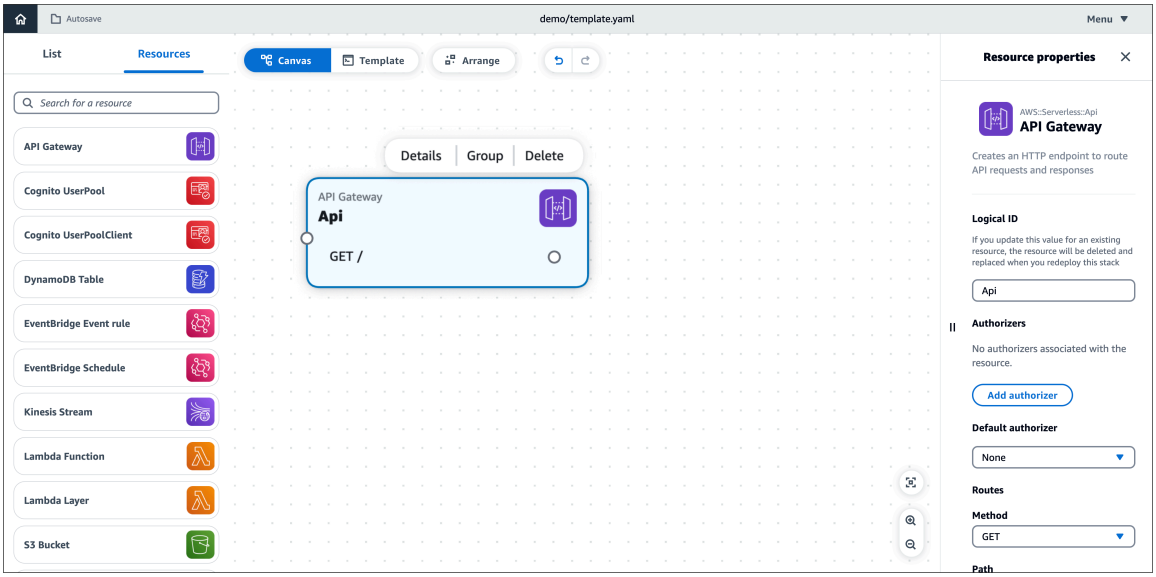
此示例使用控制台中的 Infrastructure Composer 来引用定义 API Gateway 的外部OpenAPI规范文件 REST API。

首先，从基础设施编排器主页创建一个新项目。

接下来，从菜单中选择 “激活本地同步”，激活本地同步。创建一个名为的新文件夹demo，允许提示查看文件，然后选择 “激活”。出现提示时，选择保存更改。

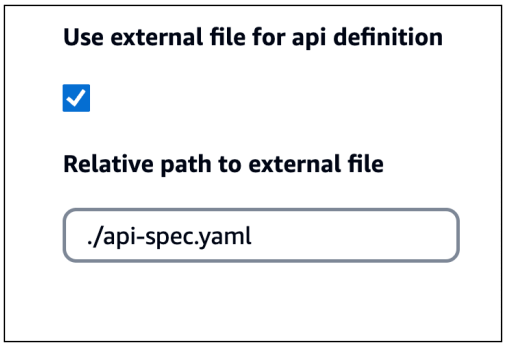


接下来，将 Amazon API Gateway 卡片拖到画布上。选择详细信息以打开资源属性面板。



在资源属性面板中，配置以下内容并保存。

- 选择 “使用外部文件进行 API 定义” 选项。
- 输入`./api-spec.yaml`作为外部文件的相对路径



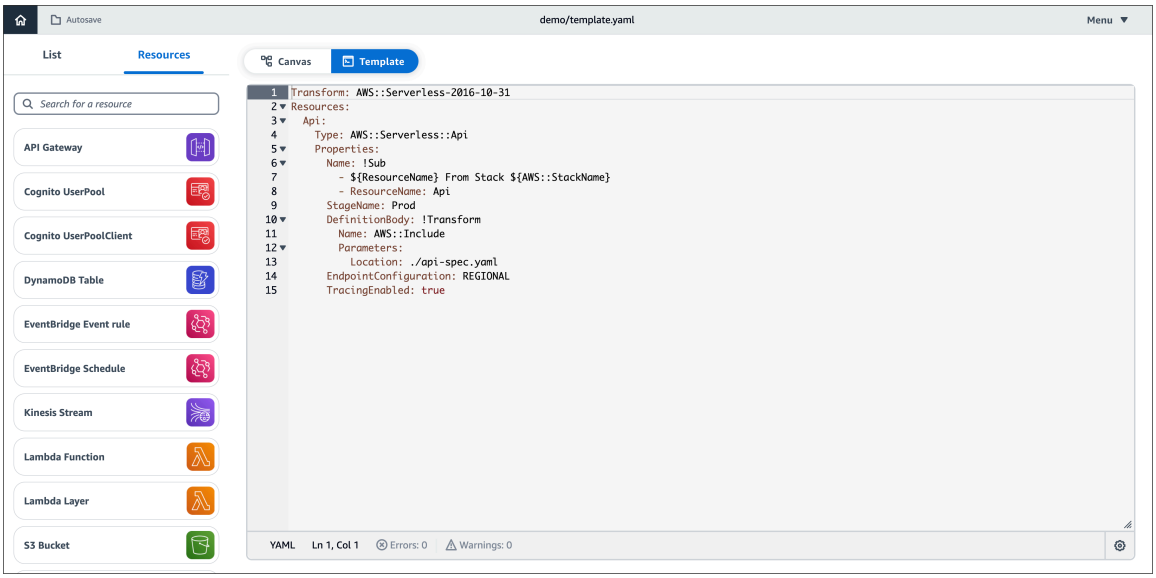
这将在我们的本地计算机上创建以下目录：

```
demo
### api-spec.yaml
```

现在，你可以在我们的本地计算机上配置外部文件。使用我们的 IDE，打开api-spec.yaml位于您的项目文件夹中的。将其内容替换为以下内容：

```
openapi: '3.0'
info: {}
paths:
  /:
    get:
      responses: {}
    post:
      x-amazon-apigateway-integration:
        credentials:
          Fn::GetAtt:
            - ApiQueuesendmessageRole
            - Arn
        httpMethod: POST
        type: aws
        uri:
          Fn::Sub: arn:${AWS::Partition}:apigateway:${AWS::Region}:sqs:path/
${AWS::AccountId}/${Queue.QueueName}
        requestParameters:
          integration.request.header.Content-Type: '''application/x-www-form-
urlencoded'''
        requestTemplates:
          application/json: Action=SendMessage&MessageBody={"data":$input.body}
        responses:
          default:
            statusCode: 200
      responses:
        '200':
          description: 200 response
```

在基础设施编排器模板视图中，您可以看到基础设施编排器已自动更新您的模板以引用外部文件。



将 Infrastructure Composer 与亚马逊虚拟私有云（亚马逊 VPC）集成

AWS 基础架构编辑器 具有与亚马逊虚拟私有云（亚马逊 VPC）服务的集成。使用基础设施编排器，您可以执行以下操作：

- 通过可视化 VPC 标签识别画布上位于 V PC 中的资源。
- 使用 VPCs 外部模板配置 AWS Lambda 函数。

下图显示了一个带有 VPC 的 Lambda 函数的应用程序示例。

要了解有关亚马逊 VPC 的更多信息，请参阅什么是亚马逊 VPC？ 在《亚马逊 VPC 用户指南》中。

主题

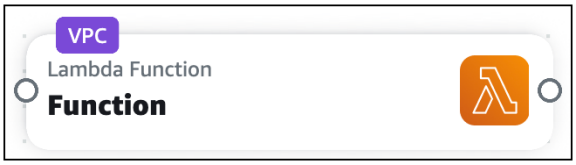
- 识别 VPC 中的基础设施编排资源和相关信息
- 在基础设施编排器 VPCs 中使用外部功能配置 Lambda 函数
- 使用基础设施编排器为外部 VPC 导入的模板中的参数
- 使用基础架构编排器向导入的模板添加新参数

- 使用基础设施编排器配置 Lambda 函数和另一个模板中定义的 VPC

识别 VPC 中的基础设施编排资源和相关信息

要将 Infrastructure Composer 与 Amazon VPC 集成，您必须首先确定 VPC 中的资源以及完成集成所需的信息。这还包括与安全组、子网标识符、参数类型、SSM 类型、静态值类型相关的配置信息。

基础设施编排器使用 VPC 标签可视化 VP C 中的资源。此标签适用于画布上的卡片。以下是带有 VPC 标签的 Lambda 函数的示例：



当您执行以下操作时，VPC 标签将应用于画布上的卡片：

- 在基础设施编排器中使用 VPC 配置 Lambda 函数。
- 导入包含配置了 VPC 的资源的模板。

安全组和子网标识符

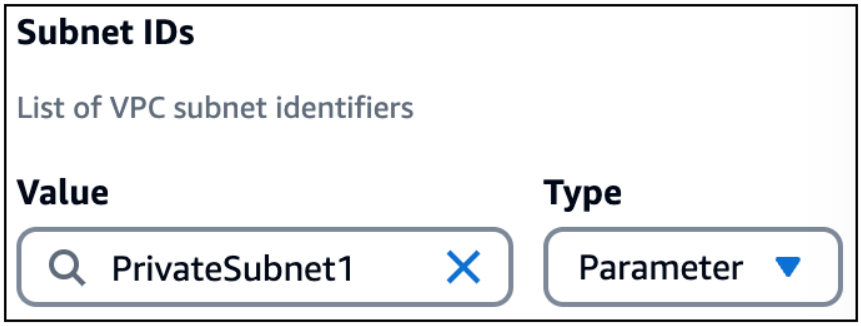
一个 Lambda 函数可以配置多个安全组和子网。要为 Lambda 函数配置安全组或子网，请提供值和类型。

- 值-安全组或子网的标识符。可接受的值将因类型而异。
- 类型-允许使用以下类型的值：
  - 参数名称
  - AWS Systems Manager (SSM) 参数存储
  - 静态值

参数类型

AWS CloudFormation 模板的`Parameters`部分可用于跨多个模板存储资源信息。有关参数的更多信息，请参阅《AWS CloudFormation 用户指南》中的参数。

对于参数类型，您可以提供参数名称。在以下示例中，我们提供了一个`PrivateSubnet1`参数名称值：



当您提供参数名称时，基础设施编排器会在模板的Parameters部分中对其进行定义。然后，基础设施编排器会引用您的 Lambda 函数资源中的参数。以下是示例：

```
...
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SubnetIds:
          - !Ref PrivateSubnet1
Parameters:
  PrivateSubnet1:
    Type: AWS::EC2::Subnet::Id
    Description: Parameter is generated by Infrastructure Composer
```

SSM 类型

SSM Parameter Store 为配置数据管理和密钥管理提供了安全的分层存储。有关更多信息，请参阅《AWS Systems Manager 用户指南》中的 AWS Systems Manager Parameter Store。

对于 SSM 类型，您可以提供以下值：

- 对 SSM 参数存储中的值的动态引用。
- 模板中定义的AWS::SSM::Parameter资源的逻辑 ID。

动态参考

您可以使用以下格式的动态引用来引用 SSM 参数存储中的值：{{resolve:ssm:reference-key}}。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的 SSM 参数。

Infrastructure Composer 创建基础设施代码，使用来自 SSM 参数存储的值来配置您的 Lambda 函数。以下是示例：

```
...
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SecurityGroupIds:
          - '{{resolve:ssm:demo-app/sg-0b61d5c742dc2c773}}'
  ...
```

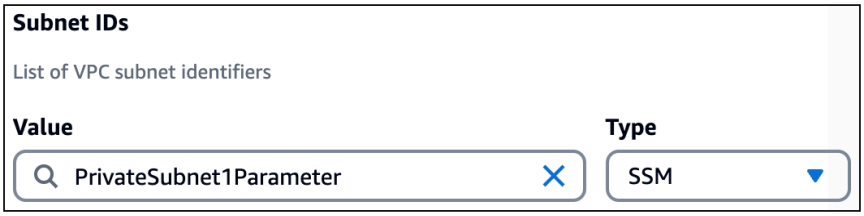
逻辑 ID

您可以通过逻辑 ID 在同一模板中引用`AWS::SSM::Parameter`资源。

以下是存储子网 ID 的名为`PrivateSubnet1Parameter`的`AWS::SSM::Parameter`资源的示例`PrivateSubnet1`：

```
...
Resources:
  PrivateSubnet1Parameter:
    Type: AWS::SSM::Parameter
    Properties:
      Name: /MyApp/VPC/SubnetIds
      Description: Subnet ID for PrivateSubnet1
      Type: String
      Value: subnet-04df123445678a036
```

以下是由 Lambda 函数的逻辑 ID 提供的此资源值的示例：



基础设施编排器创建基础设施代码，以便使用 SSM 参数配置您的 Lambda 函数：

```
...
```

```
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SubnetIds:
          - !Ref PrivateSubnet1Parameter
  ...
  PrivateSubnet1Parameter:
    Type: AWS::SSM::Parameter
    Properties:
      ...
```

静态值类型

将安全组或子网部署到时 CloudFormation，会创建一个 ID 值。您可以将此 ID 作为静态值提供。

对于静态值类型，以下是有效值：

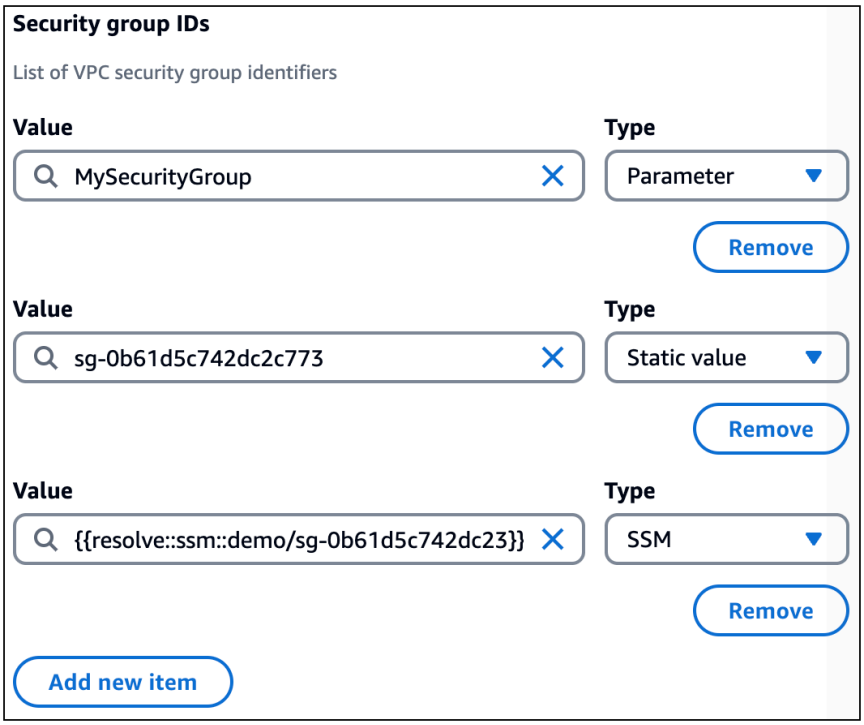
- 对于安全组，请提供`GroupId`。有关更多信息，请参阅AWS CloudFormation 用户指南中的返回值。以下是示例：`sg-0b61d5c742dc2c773`
- 对于子网，请提供。`SubnetId`有关更多信息，请参阅AWS CloudFormation 用户指南中的返回值。以下是示例：`subnet-01234567890abcdef`

基础设施编排器创建基础设施代码，用静态值配置您的 Lambda 函数。以下是示例：

```
...
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SecurityGroupIds:
          - subnet-01234567890abcdef
        SubnetIds:
          - sg-0b61d5c742dc2c773
  ...
```

使用多种类型

对于安全组和子网，您可以同时使用多种类型。以下示例通过提供不同类型的值为 Lambda 函数配置三个安全组：



基础架构编排器引用了该SecurityGroupIds属性下的所有三个值：

```
...
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SecurityGroupIds:
          - !Ref MySecurityGroup
          - sg-0b61d5c742dc2c773
          - '{{resolve::ssm::demo/sg-0b61d5c742dc23}}'
      ...
Parameters:
  MySecurityGroup:
    Type: AWS::EC2::SecurityGroup::Id
    Description: Parameter is generated by Infrastructure Composer
```

在基础设施编排器 VPCs 中使用外部功能配置 Lambda 函数

要开始使用在其他模板上定义的 VPC 配置 Lambda 函数，请使用 Lambda 函数增强型组件卡。此卡片表示使用 AWS Serverless Application Model (AWS SAM) `AWS::Serverless::Function` 资源类型的 Lambda 函数。

使用来自外部模板的 VPC 配置 Lambda 函数

1. 在 Lambda 函数资源属性面板中，展开 VP C 设置（高级）下拉部分。
2. 选择分配给外部 VPC。
3. 为要为 Lambda 函数配置的安全组和子网提供值。有关详细信息，请参阅 安全组和子网标识符。
4. 保存您的更改。

使用基础设施编排器为外部 VPC 导入的模板中的参数

当您导入包含为外部 VPC 的安全组和子网定义的参数的现有模板时，Infrastructure Composer 会提供一个下拉列表供您从中选择参数。

以下是导入模板`Parameters`部分的示例：

```
...
Parameters:
  VPCSecurityGroups:
    Description: Security group IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::SecurityGroup::Id>
  VPCSubnets:
    Description: Subnet IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::Subnet::Id>
  VPCSubnet:
    Description: Subnet Id generated by Infrastructure Composer
    Type: AWS::EC2::Subnet::Id
...
```

在画布上为新的 Lambda 函数配置外部 VPC 时，这些参数将显示在下拉列表中。以下是示例：

导入列表参数类型时的限制

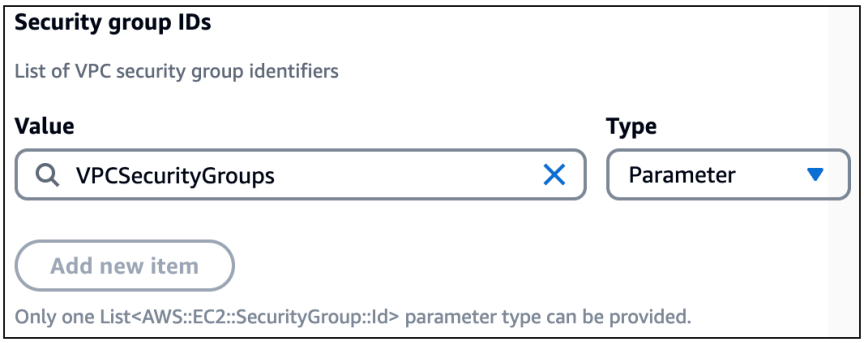
通常，您可以为每个 Lambda 函数指定多个安全组和子网标识符。如果现有模板包含列表参数类型，例如`List<AWS::EC2::SecurityGroup::Id>`或`List<AWS::EC2::Subnet::Id>`，则只能指定一个标识符。

有关参数列表类型的更多信息，请参阅《AWS CloudFormation 用户指南》中AWS支持的特定参数类型。

以下是定义`VPCSecurityGroups`为列表参数类型的模板示例：

```
...
Parameters:
  VPCSecurityGroups:
    Description: Security group IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::SecurityGroup::Id>
...
```

在 Infrastructure Composer 中，如果您选择该`VPCSecurityGroups`值作为 Lambda 函数的安全组标识符，您将看到以下消息：



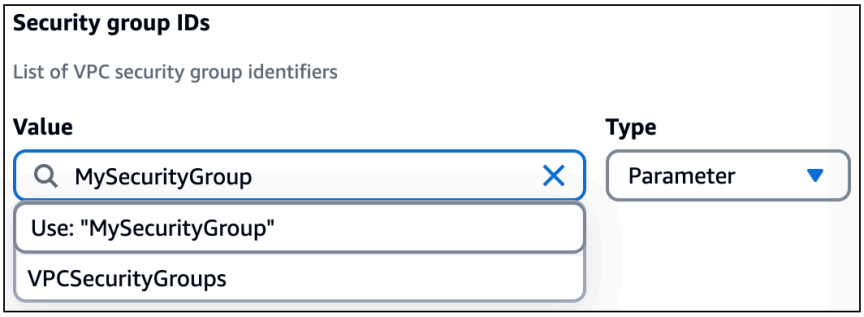
之所以出现这种限制，是因为`AWS::Lambda::Function VpcConfig`对象的`SecurityGroupIds`和`SubnetIds`属性都只接受字符串值列表。由于单个列表参数类型包含字符串列表，因此它可以是指定时提供的唯一对象。

对于列表参数类型，以下是使用 Lambda 函数配置时如何在模板中定义它们的示例：

```
...
Parameters:
  VPCSecurityGroups:
    Description: Security group IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::SecurityGroup::Id>
  VPCSubnets:
    Description: Subnet IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::Subnet::Id>
Resources:
  ...
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SecurityGroupIds: !Ref VPCSecurityGroups
        SubnetIds: !Ref VPCSubnets
```

使用基础架构编排器向导入的模板添加新参数

导入已定义参数的现有模板时，也可以创建新参数。与其从下拉列表中选择现有参数，不如提供新的类型和值。以下是创建名为的新参数的示例`MySecurityGroup`：



对于您在资源属性面板中为 Lambda 函数提供的所有新值，基础设施编排器在 Lambda 函数的`SecurityGroupIds`或`SubnetIds`属性下的列表中对其进行定义。以下是示例：

```
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
```

```
      VpcConfig:
        SecurityGroupIds:
          - sg-94b3a1f6
        SubnetIds:
          - !Ref SubnetParameter
          - !Ref VPCSubnet
```

如果要从外部模板引用列表参数类型的逻辑 ID，我们建议您使用模板视图并直接修改您的模板。列表参数类型的逻辑 ID 应始终作为单个值和唯一值提供。

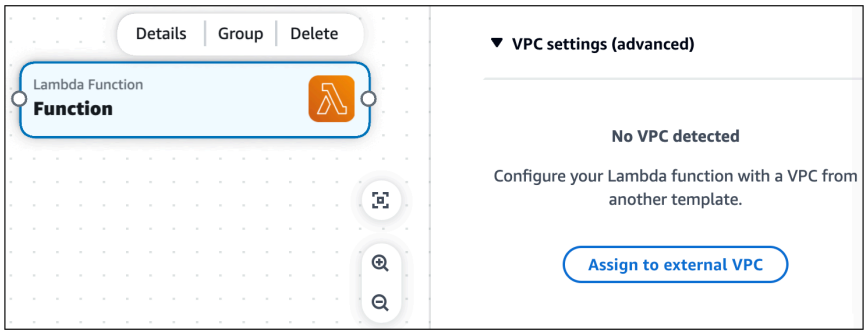
```
...
Parameters:
  VPCSecurityGroups:
    Description: Security group IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::SecurityGroup::Id>
  VPCSubnets:
    Description: Subnet IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::Subnet::Id>
Resources:
  ...
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SecurityGroupIds: !Ref VPCSecurityGroups # Valid syntax
        SubnetIds:
          - !Ref VPCSubnets # Not valid syntax
```

使用基础设施编排器配置 Lambda 函数和另一个模板中定义的 VPC

在此示例中，我们在基础设施编排器中使用在另一个模板上定义的 VPC 来配置 Lambda 函数。

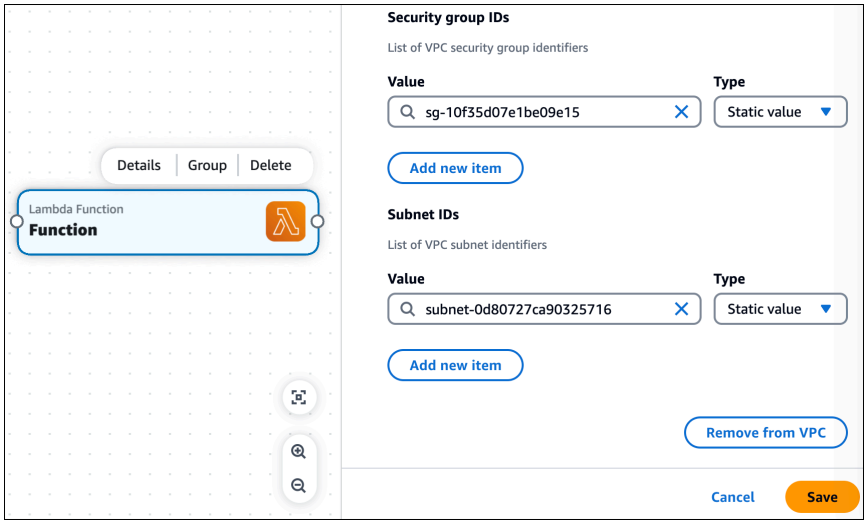
我们首先将 Lambda 函数增强型组件卡片拖到画布上。

接下来，我们打开卡片的资源属性面板并展开 VPC 设置（高级）下拉部分。

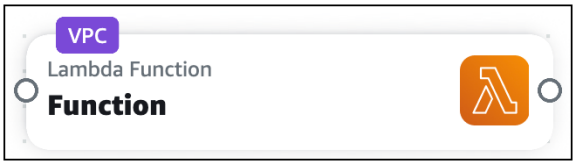


接下来，我们选择 “分配给外部 VPC”，开始从外部模板配置 VPC。

在此示例中，我们引用了安全组 ID 和子网 ID。这些值是在部署定义 VPC 的模板时创建的。我们选择静态值类型并输入我们的值 IDs。完成后我们选择 “保存”。



现在，我们的 Lambda 函数已使用我们的 VPC 进行配置，VPC 标签将显示在我们的卡片上。



Infrastructure Composer 创建了基础设施代码，用于使用外部 VPC 的安全组和子网来配置 Lambda 函数。

```
Transform: AWS::Serverless-2016-10-31
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      Description: !Sub
```

```
        - Stack ${AWS::StackName} Function ${ResourceName}
        - ResourceName: Function
      CodeUri: src/Function
      Handler: index.handler
      Runtime: nodejs18.x
      MemorySize: 3008
      Timeout: 30
      Tracing: Active
      VpcConfig:
        SecurityGroupIds:
          - sg-10f35d07e1be09e15
        SubnetIds:
          - subnet-0d80727ca90325716
  FunctionLogGroup:
    Type: AWS::Logs::LogGroup
    DeletionPolicy: Retain
    Properties:
      LogGroupName: !Sub /aws/lambda/${Function}
```

将您的基础架构 Composer 无服务器应用程序部署到云端 AWS

用于 AWS 基础架构编辑器 设计部署就绪型无服务器应用程序。要进行部署，请使用任何 AWS CloudFormation 兼容的服务。我们建议使用 AWS Serverless Application Model (AWS SAM)。

AWS SAM 是一个开源框架，它为开发者提供了用于在上 AWS构建和运行无服务器应用程序的工具。使用 AWS SAM速记语法，开发人员可以声明 CloudFormation 资源和专门的无服务器资源，这些资源和在部署期间转换为基础架构的无服务器资源。

重要 AWS SAM 概念

在使用之前 AWS SAM，请务必熟悉它的一些基本概念。

- AWS SAM 工作原理：本主题位于《AWS Serverless Application Model 开发人员指南》中，提供了有关用于创建无服务器应用程序的主要组件的重要信息： AWS SAMCLI、 AWS SAM 项目和 AWS SAM 模板。
- 如何使用 AWS Serverless Application Model (AWS SAM)：本主题位于《AWS Serverless Application Model 开发人员指南》中，提供了将应用程序部署 AWS SAM 到 AWS 云端所需完成的步骤的高级概述。

在 Infrastructure Composer 中设计应用程序时，您可以使用sam sync命令 AWS SAMCLI自动检测本地更改并将这些更改部署到 CloudFormation。要了解更多信息，请参阅AWS Serverless Application Model 开发者指南中的使用 sam sync。

后续步骤

请参阅 使用 AWS SAMCLI和基础架构编排器进行部署设置 “准备部署应用程序”。

使用 AWS SAMCLI和基础架构编排器进行部署设置

要使用部署应用程序 AWS SAM，您首先需要安装并访问 AWSCLI和 AWS SAMCLI。本节中的主题提供了有关执行此操作的详细信息。

安装 AWS CLI

我们建议在安装 AWSCLI之前安装和设置 AWS SAMCLI。有关说明，请参阅《AWS Command Line Interface 用户指南》 AWS CLI中的安装或更新到最新版本的。

> **Note**
>
> 安装完成后 AWSCLI，必须配置 AWS 凭据。要了解更多信息，请参阅《AWS Command Line Interface 用户指南》中的快速设置。

安装 AWS SAM CLI

要安装 AWS SAMCLI，请参阅《AWS Serverless Application Model 开发人员指南》 AWS SAMCLI中的安装。

访问 AWS SAMCLI

如果您使用的是中的基础架构编排器 AWS 管理控制台，则有以下选项可供使用 AWS SAMCLI。

激活本地同步模式

在本地同步模式下，您的项目文件夹（包括 AWS SAM 模板）会自动保存到本地计算机。基础架构编排器以一种可 AWS SAM 识别的方式构建您的项目目录。您可以 AWS SAMCLI从项目的根目录运行。

有关本地同步模式的更多信息，请参阅在基础设施编排控制台中本地同步并保存您的项目。

导出您的模板

您可以将模板导出到本地计算机。然后， AWS SAMCLI从包含模板的父文件夹中运行。您也可以将该`--template-file`选项与任何 AWS SAMCLI命令一起使用，并提供模板的路径。

使用来自《基础架构编排器》 AWS Toolkit for Visual Studio Code

您可以使用适用于 VS Code 的 Toolkit 中的基础设施编排器将基础设施编排器引入本地计算机。然后，使用基础架构编排器和来 AWS SAMCLI自 VS Code 的。

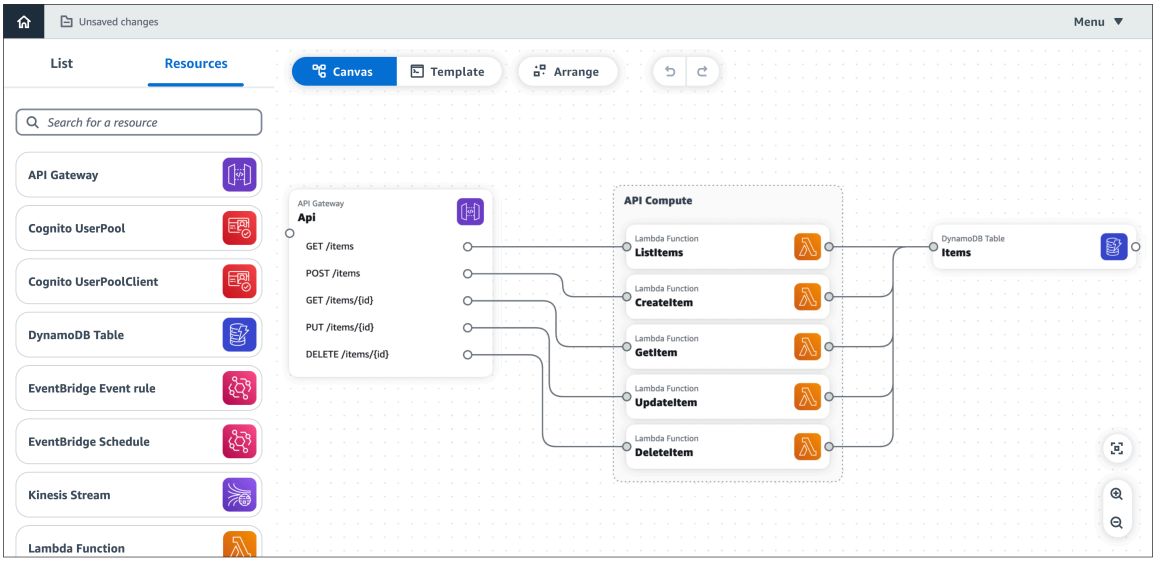
后续步骤

要部署应用程序，请参阅使用基础架构编排器 AWS SAM 进行构建和部署。

使用基础架构编排器 AWS SAM 进行构建和部署

现在您已经完成了使用 AWS SAMCLI和基础架构编排器进行部署设置，您可以使用 AWS SAM 和基础架构编排器来部署您的应用程序。本节提供了一个示例，详细说明了如何执行此操作。您也可以参阅《AWS Serverless Application Model 开发人员指南》 AWS SAM中的使用部署应用程序和资源，了解有关使用部署应用程序的说明 AWS SAM。

此示例向您展示如何构建和部署 Infrastructure Composer 演示应用程序。该演示应用程序具有以下资源：



> **Note**
>
> - 要了解有关演示应用程序的更多信息，请参阅加载和修改基础设施编排器演示项目。
> - 在本示例中，我们使用已激活本地同步的基础设施编排器。

1. 使用sam build命令来构建应用程序。

```
$ sam build
...
Build Succeeded

Built Artifacts  : .aws-sam/build
Built Template   : .aws-sam/build/template.yaml

Commands you can use next
```

```
=========================
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided
```

在项目文件夹中 AWS SAMCLI创建./aws-sam目录。此目录包含应用程序的 Lambda 函数的构建工件。以下是项目目录的输出：

```
.
### README.md
### samconfig.toml
### src
#   ### CreateItem
#   #   ### index.js
#   #   ### package.json
#   ### DeleteItem
#   #   ### index.js
#   #   ### package.json
#   ### GetItem
#   #   ### index.js
#   #   ### package.json
#   ### ListItems
#   #   ### index.js
#   #   ### package.json
#   ### UpdateItem
#       ### index.js
#       ### package.json
### template.yaml
```

2. 现在，该应用程序已准备就绪，可以部署了。我们将使用sam deploy --guided。这将通过一系列提示为您的应用程序做好部署准备。

```
$ sam deploy --guided
...
Configuring SAM deploy
======================

    Looking for config file [samconfig.toml] :  Found
    Reading default arguments  :  Success

    Setting default arguments for 'sam deploy'
```

```
    ========================================
    Stack Name [aws-app-composer-basic-api]: AWS Region [us-west-2]:
    #Shows you resources changes to be deployed and require a 'Y' to initiate
 deploy
    Confirm changes before deploy [y/N]:
    #SAM needs permission to be able to create roles to connect to the resources in
 your template
    Allow SAM CLI IAM role creation [Y/n]:
    #Preserves the state of previously provisioned resources when an operation
 fails
    Disable rollback [y/N]:
    ListItems may not have authorization defined, Is this okay? [y/N]: y
    CreateItem may not have authorization defined, Is this okay? [y/N]: y
    GetItem may not have authorization defined, Is this okay? [y/N]: y
    UpdateItem may not have authorization defined, Is this okay? [y/N]: y
    DeleteItem may not have authorization defined, Is this okay? [y/N]: y
    Save arguments to configuration file [Y/n]:
    SAM configuration file [samconfig.toml]:
    SAM configuration environment [default]:
```

AWS SAMCLI显示了将要部署的内容的摘要：

```
Deploying with following values
    ===============================
    Stack name                   : aws-app-composer-basic-api
    Region                       : us-west-2
    Confirm changeset            : False
    Disable rollback             : False
    Deployment s3 bucket         : aws-sam-cli-managed-default-samclisam-s3-
demo-1b3x26zbcdkqr
    Capabilities                 : ["CAPABILITY_IAM"]
    Parameter overrides          : {}
    Signing Profiles             : {}
```

首先通过创建 CloudFormation 变更集来 AWS SAMCLI部署应用程序：

```
Initiating deployment
=====================
Uploading to aws-app-composer-basic-api/4181c909ee2440a728a7a129dafb83d4.template
 7087 / 7087  (100.00%)

Waiting for changeset to be created..
```

```
CloudFormation stack changeset
-------------------------------------------------------------------------------------------
Operation                               LogicalResourceId
 ResourceType                            Replacement
-------------------------------------------------------------------------------------------
+ Add                                   ApiDeploymentcc153d135b
 AWS::ApiGateway::Deployment             N/A
+ Add                                   ApiProdStage
 AWS::ApiGateway::Stage                  N/A
+ Add                                   Api
 AWS::ApiGateway::RestApi                N/A
+ Add                                   CreateItemApiPOSTitemsPermissionP
 AWS::Lambda::Permission                 N/A
                                        rod
+ Add                                   CreateItemRole
 AWS::IAM::Role                          N/A
+ Add                                   CreateItem
 AWS::Lambda::Function                   N/A
+ Add                                   DeleteItemApiDELETEitemsidPermiss
 AWS::Lambda::Permission                 N/A
                                        ionProd
+ Add                                   DeleteItemRole
 AWS::IAM::Role                          N/A
+ Add                                   DeleteItem
 AWS::Lambda::Function                   N/A
+ Add                                   GetItemApiGETitemsidPermissionPro
 AWS::Lambda::Permission                 N/A
                                        d
+ Add                                   GetItemRole
 AWS::IAM::Role                          N/A
+ Add                                   GetItem
 AWS::Lambda::Function                   N/A
+ Add                                   Items
 AWS::DynamoDB::Table                    N/A
+ Add                                   ListItemsApiGETitemsPermissionPro
 AWS::Lambda::Permission                 N/A
                                        d
+ Add                                   ListItemsRole
 AWS::IAM::Role                          N/A
+ Add                                   ListItems
 AWS::Lambda::Function                   N/A
+ Add                                   UpdateItemApiPUTitemsidPermission
 AWS::Lambda::Permission                 N/A
                                        Prod
```

```
+ Add                                 UpdateItemRole
 AWS::IAM::Role                        N/A
+ Add                                 UpdateItem
 AWS::Lambda::Function                 N/A
-----------------------------------------------------------------------------------------

Changeset created successfully. arn:aws:cloudformation:us-
west-2:513423067560:changeSet/samcli-deploy1677472539/967ab543-f916-4170-b97d-
c11a6f9308ea
```

然后，AWS SAMCLI部署应用程序：

```
CloudFormation events from stack operations (refresh every 0.5 seconds)
-----------------------------------------------------------------------------------------
ResourceStatus                        ResourceType
 LogicalResourceId                     ResourceStatusReason
-----------------------------------------------------------------------------------------
CREATE_IN_PROGRESS                    AWS::DynamoDB::Table                 Items
                          -
CREATE_IN_PROGRESS                    AWS::DynamoDB::Table                 Items
                         Resource creation Initiated
CREATE_COMPLETE                       AWS::DynamoDB::Table                 Items
                          -
CREATE_IN_PROGRESS                    AWS::IAM::Role
 DeleteItemRole                        -
CREATE_IN_PROGRESS                    AWS::IAM::Role
 ListItemsRole                         -
CREATE_IN_PROGRESS                    AWS::IAM::Role
 UpdateItemRole                        -
CREATE_IN_PROGRESS                    AWS::IAM::Role                       GetItemRole
                          -
CREATE_IN_PROGRESS                    AWS::IAM::Role
 CreateItemRole                        -
CREATE_IN_PROGRESS                    AWS::IAM::Role
 DeleteItemRole                        Resource creation Initiated
CREATE_IN_PROGRESS                    AWS::IAM::Role
 ListItemsRole                         Resource creation Initiated
CREATE_IN_PROGRESS                    AWS::IAM::Role                       GetItemRole
                         Resource creation Initiated
CREATE_IN_PROGRESS                    AWS::IAM::Role
 UpdateItemRole                        Resource creation Initiated
CREATE_IN_PROGRESS                    AWS::IAM::Role
 CreateItemRole                        Resource creation Initiated
```

```
CREATE_COMPLETE                    AWS::IAM::Role
 DeleteItemRole                     -
CREATE_COMPLETE                    AWS::IAM::Role
 ListItemsRole                      -
CREATE_COMPLETE                    AWS::IAM::Role                    GetItemRole
                         -
CREATE_COMPLETE                    AWS::IAM::Role
 UpdateItemRole                     -
CREATE_COMPLETE                    AWS::IAM::Role
 CreateItemRole                     -
CREATE_IN_PROGRESS                 AWS::Lambda::Function             DeleteItem
                         -
CREATE_IN_PROGRESS                 AWS::Lambda::Function             CreateItem
                         -
CREATE_IN_PROGRESS                 AWS::Lambda::Function             ListItems
                         -
CREATE_IN_PROGRESS                 AWS::Lambda::Function             UpdateItem
                         -
CREATE_IN_PROGRESS                 AWS::Lambda::Function             DeleteItem
                        Resource creation Initiated
CREATE_IN_PROGRESS                 AWS::Lambda::Function             GetItem
                         -
CREATE_IN_PROGRESS                 AWS::Lambda::Function             ListItems
                        Resource creation Initiated
CREATE_IN_PROGRESS                 AWS::Lambda::Function             CreateItem
                        Resource creation Initiated
CREATE_IN_PROGRESS                 AWS::Lambda::Function             UpdateItem
                        Resource creation Initiated
CREATE_IN_PROGRESS                 AWS::Lambda::Function             GetItem
                        Resource creation Initiated
CREATE_COMPLETE                    AWS::Lambda::Function             DeleteItem
                         -
CREATE_COMPLETE                    AWS::Lambda::Function             ListItems
                         -
CREATE_COMPLETE                    AWS::Lambda::Function             CreateItem
                         -
CREATE_COMPLETE                    AWS::Lambda::Function             UpdateItem
                         -
CREATE_COMPLETE                    AWS::Lambda::Function             GetItem
                         -
CREATE_IN_PROGRESS                 AWS::ApiGateway::RestApi          Api
                         -
CREATE_IN_PROGRESS                 AWS::ApiGateway::RestApi          Api
                        Resource creation Initiated
```

```
CREATE_COMPLETE                    AWS::ApiGateway::RestApi           Api
                          -
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 GetItemApiGETitemsidPermissionPro  -
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 ListItemsApiGETitemsPermissionPro  -
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 DeleteItemApiDELETEitemsidPermiss  -
                                                                       ionProd
CREATE_IN_PROGRESS                 AWS::ApiGateway::Deployment
 ApiDeploymentcc153d135b            -
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 UpdateItemApiPUTitemsidPermission  -
                                                                       Prod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 CreateItemApiPOSTitemsPermissionP  -
                                                                       rod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 GetItemApiGETitemsidPermissionPro  Resource creation Initiated
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 UpdateItemApiPUTitemsidPermission  Resource creation Initiated
                                                                       Prod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 CreateItemApiPOSTitemsPermissionP  Resource creation Initiated
                                                                       rod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 ListItemsApiGETitemsPermissionPro  Resource creation Initiated
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 DeleteItemApiDELETEitemsidPermiss  Resource creation Initiated
                                                                       ionProd
CREATE_IN_PROGRESS                 AWS::ApiGateway::Deployment
 ApiDeploymentcc153d135b            Resource creation Initiated
CREATE_COMPLETE                    AWS::ApiGateway::Deployment
 ApiDeploymentcc153d135b            -
CREATE_IN_PROGRESS                 AWS::ApiGateway::Stage
 ApiProdStage                       -
CREATE_IN_PROGRESS                 AWS::ApiGateway::Stage
 ApiProdStage                       Resource creation Initiated
CREATE_COMPLETE                    AWS::ApiGateway::Stage
 ApiProdStage                       -
```

```
CREATE_COMPLETE                     AWS::Lambda::Permission
 CreateItemApiPOSTitemsPermissionP   -
                                                                    rod
CREATE_COMPLETE                     AWS::Lambda::Permission
 UpdateItemApiPUTitemsidPermission   -
                                                                    Prod
CREATE_COMPLETE                     AWS::Lambda::Permission
 ListItemsApiGETitemsPermissionPro   -
                                                                    d
CREATE_COMPLETE                     AWS::Lambda::Permission
 DeleteItemApiDELETEitemsidPermiss   -
                                                                    ionProd
CREATE_COMPLETE                     AWS::Lambda::Permission
 GetItemApiGETitemsidPermissionPro   -
                                                                    d
CREATE_COMPLETE                     AWS::CloudFormation::Stack          aws-app-
composer-basic-api          -
-------------------------------------------------------------------------------------------
```

最后，将显示一条消息，通知您部署已成功：

```
Successfully created/updated stack - aws-app-composer-basic-api in us-west-2
```

使用基础架构编排器 AWS SAM 来删除堆栈

此示例说明如何使用sam delete命令删除 CloudFormation 堆栈。

sam delete在中输入命令 AWS SAMCLI并确认是否要删除堆栈和模板：

```
$ sam delete
 Are you sure you want to delete the stack aws-app-composer-basic-api in the region us-
west-2 ? [y/N]: y
 Do you want to delete the template file 30439348c0be6e1b85043b7a935b34ab.template in
 S3? [y/N]: y
 - Deleting S3 object with key eb226ca86d1bc4e9914ad85eb485fed8
 - Deleting S3 object with key 875e4bcf4b10a6a1144ad83158d84b6d
 - Deleting S3 object with key 20b869d98d61746dedd9aa33aa08a6fb
 - Deleting S3 object with key c513cedc4db6bc184ce30e94602741d6
 - Deleting S3 object with key c7a15d7d8d1c24b77a1eddf8caebc665
 - Deleting S3 object with key e8b8984f881c3732bfb34257cdd58f1e
 - Deleting S3 object with key 3185c59b550594ee7fca7f8c36686119.template
```

```
 - Deleting S3 object with key 30439348c0be6e1b85043b7a935b34ab.template
 - Deleting Cloudformation stack aws-app-composer-basic-api

Deleted successfully
```

AWS 基础架构编辑器 故障排除

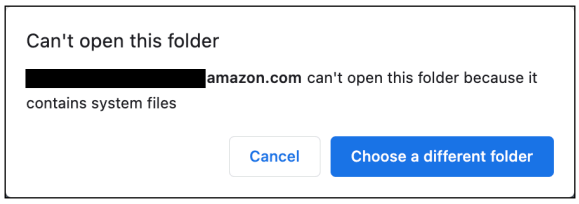
本节中的主题提供了有关在使用时对错误消息进行故障排除的指导 AWS 基础架构编辑器。

主题

错误消息

“无法打开此文件夹”

错误示例：



可能的原因：Infrastructure Composer 无法使用本地同步模式访问敏感目录。

要了解有关此错误的更多信息，请参阅数据基础架构编排器可以访问。

尝试连接到其他本地目录，或者在停用本地同步的情况下使用基础设施编排器。

“模板不兼容”

错误示例：在 Infrastructure Composer 中加载新项目时，您会看到以下内容：

可能的原因：您的项目包含一个外部引用的文件，该文件在基础设施编排器中不受支持。

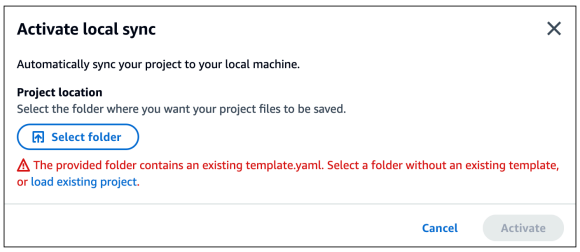
要了解基础设施编排器中支持的外部文件，请参阅引用外部文件。

可能的原因：您的项目链接到其他本地目录中的外部文件。

将外部引用的文件移至您选择在 Infrastructure Composer 本地同步模式下使用的目录的子目录中。

“提供的文件夹包含现有的模板.yaml”

尝试激活本地同步时，您会看到以下错误：



可能的原因：您选择的文件夹中已包含模板.yaml 文件。

选择另一个不包含应用程序模板的目录，或者创建一个新目录。

“您的浏览器无权将您的项目保存在该文件夹中...”

可能的原因：Infrastructure Composer 无法使用本地同步模式访问敏感目录。

要了解有关此错误的更多信息，请参阅数据基础架构编排器可以访问。

尝试连接到其他本地目录，或者在停用本地同步的情况下使用基础设施编排器。

安全性 AWS 基础架构编辑器

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 负责保护在云中运行 AWS 服务的基础架构 AWS 云。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用的合规计划 AWS 基础架构编辑器，请参阅按合规计划划分的范围内的 AWSAWS 服务按合规计划。
- 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基础架构编排器时如何应用分担责任模型。以下主题向您介绍如何配置基础架构编排器以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的基础设施 Composer 资源。

主题

- 中的数据保护 AWS 基础架构编辑器
- AWS Identity and Access Management 对于 AWS 基础架构编辑器
- 合规性验证 AWS 基础架构编辑器
- 韧性在 AWS 基础架构编辑器

中的数据保护 AWS 基础架构编辑器

分 AWS 担责任模型适用于 AWS 基础架构编排器中的数据保护。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 云。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。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

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 或 AWS 服务 使用基础架构编排器或其他 AWS CLI工具时 AWS SDKs。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 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。

> **Note**
>
> 您输入到 Infrastructure Composer 的所有数据仅用于在基础架构编排器中提供功能以及生成保存在计算机本地的项目文件和目录。基础设施 Composer 不会保存、存储或传输任何此类数据。

数据加密

Infrastructure Composer 不会对客户内容进行加密，因为数据不会被保存、存储或传输。

静态加密

Infrastructure Composer 不会对客户内容进行加密，因为数据不会被保存、存储或传输。

传输中加密

Infrastructure Composer 不会对客户内容进行加密，因为数据不会被保存、存储或传输。

密钥管理

Infrastructure Composer 不支持密钥管理，因为不保存、存储或传输客户内容。

互联网络流量隐私

Infrastructure Composer 不会生成本地客户端和应用程序的流量。

AWS Identity and Access Management 对于 AWS 基础架构编辑器

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用基础设施编排器资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- 如何 AWS 基础架构编辑器 与 IAM 配合使用

受众

基础架构 Composer 至少需要对的只读访问权限 AWS 管理控制台。任何拥有此权限的用户都可以使用基础设施编排器的所有功能。不支持精细访问基础架构编排器的特定功能。

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center （例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的如何登录您的 AWS 账户。

对于编程访问， AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的AWS 签名版本 4。

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务 和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。 AWS IAM Identity Center有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？。

IAM 用户和群组

IAM 用户是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问。

IAM 组指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 IAM 用户使用案例。

IAM 角色

IAM 角色是具有特定权限的身份，可提供临时凭证。您可以通过从用户切换到 IAM 角色（控制台）或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。 AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概述。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中指定主体。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACLs)

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持的服务示例 ACLs。 AWS WAF要了解更多信息 ACLs，请参阅《亚马逊简单存储服务开发者指南》中的访问控制列表 (ACL) 概述。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《 IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。

- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的资源控制策略 (RCPs)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

如何 AWS 基础架构编辑器 与 IAM 配合使用

AWS 基础架构编辑器 至少需要对的只读访问权限 AWS 管理控制台。任何拥有此权限的用户都可以使用基础设施编排器的所有功能。不支持精细访问基础架构编排器的特定功能。

将项目模板和文件部署到时 AWS CloudFormation，需要具备必要的权限才能到位。要了解更多信息，请参阅《AWS CloudFormation 用户指南》中的使用 AWS Identity and Access Management控制访问。

下表显示了可以与哪些 IAM 功能一起使用 AWS 基础架构编辑器。

IAM 功能	基础架构编排器支持
基于身份的策略	否
基于资源的策略	否
策略操作	否
策略资源	否
策略条件密钥	否
ACLs	否
ABAC（策略中的标签）	否
临时凭证	是

IAM 功能	基础架构编排器支持
主体权限	否
服务角色	否
服务关联角色	否

要全面了解基础设施编排器和其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

基础架构编排器的基于身份的策略

支持基于身份的策略：否

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

基础架构编排器中基于资源的策略

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

基础设施编排器的策略操作

支持策略操作：否

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看基础设施编排器操作列表，请参阅《服务授权参考》中的 “ AWS 基础设施编排器定义的操作”。

基础设施编排器的策略资源

支持策略资源：否

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

要查看基础设施编排器资源类型及其列表 ARNs，请参阅《服务授权参考》中的 “AWS 基础设施编排器定义的资源”。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅AWS 基础设施编排器定义的操作。

基础架构编排器的策略条件密钥

支持特定于服务的策略条件键：否

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看基础设施编排器条件密钥列表，请参阅《服务授权参考》中的 “AWS 基础设施编排器条件密钥”。要了解您可以使用哪些操作和资源使用条件键，请参阅AWS 基础架构编排器定义的操作。

ACLs 在基础设施编辑器中

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

带有基础设施编辑器的 ABAC

支持 ABAC（策略中的标签）：否

基于属性的访问权限控制（ABAC）是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/`*`key-name`*`aws:RequestTag/`*`key-name`* 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

在基础设施编排器中使用临时证书

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的临时安全凭证和使用 IAM 的。AWS 服务

您可以使用临时证书通过访问基础设施编排器 AWS 管理控制台。有关示例，请参阅 IAM 用户指南中的启用自定义身份代理对 AWS 控制台的访问权限。

基础设施编排器的跨服务主体权限

支持转发访问会话（FAS）：否

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

基础设施编排器的服务角色

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会中断基础设施编排器的功能。只有在基础设施编排器提供相关指导时才编辑服务角色。

基础架构编排器的服务相关角色

支持服务相关角色：否

服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅能够与 IAM 搭配使用的AWS 服务。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

合规性验证 AWS 基础架构编辑器

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 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，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。有关您在使用时的合规责任的更多信息 AWS 服务 ，请参阅AWS 安全文档。

韧性在 AWS 基础架构编辑器

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。 AWS 区域 提供多个物理分隔和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。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设施相比，可用区具有更高的可用性、容错能力和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础设施。

您输入到 Infrastructure Composer 的所有数据仅用于在基础架构编排器中提供功能以及生成保存在计算机本地的项目文件和目录。基础架构编排器不保存或存储任何此类数据。

基础架构编辑器的文档历史记录

下表描述了基础架构编排器的重要文档版本。如需有关此文档更新的通知，您可以订阅 RSS 源。

- 文档最新更新时间：2023 年 11 月 30 日

变更	说明	日期
重组并更新了整个开发人员指南中的内容	重新组织并重构了指南，以提高可发现性和可用性。更新和改进了标题。在介绍主题和概念时提供了额外的详细信息。	2024 年 8 月 1 日
添加了有关在 CloudFormation 控制台模式下使用基础设施编排器的文档，并重新调整了《基础设施编排开发者指南》。	AWS 基础架构编辑器 现在可以在 CloudFormation 控制台模式下使用。要了解更多信息，请参阅在 CloudFormation 控制台模式下使用基础架构编排器。此外，用户指南中的许多内容都经过了重组，以创建简化的体验。	2024 年 3 月 28 日
添加了与基础设施编排器集成的文档 CodeWhisperer	AWS 基础架构编辑器 来自 VS Code 的 Toolkit for VS Code 提供了与亚马逊的集成 CodeWhisperer。要了解更多信息，请参阅在 Amazon AWS 基础架构编辑器 上使用 CodeWhisperer。	2023 年 11 月 30 日
添加了有关使用基础架构编排器部署应用程序的文档 AWS Toolkit for Visual Studio Code	使用基础设施编排器画布上的同步按钮将您的应用程序部署到 AWS 云。要了解更多信息，请参阅使用 sam sync 部署应用程序。	2023 年 11 月 30 日

添加了《基础设施编排器》中的文档 AWS Toolkit for Visual Studio Code	现在，您可以将 VS Code 中的基础架构编排器与配合使用 AWS Toolkit for Visual Studio Code。要了解更多信息，请参阅AWS 基础架构编辑器 从中使用 AWS Toolkit for Visual Studio Code。	2023 年 11 月 30 日
新增了 Step Functions 工作流程工作室	从基础设施编辑器画布启动 Step Functions 工作流工作室。要了解更多信息，请参阅 AWS 基础架构编辑器 与一起使用 AWS Step Functions。	2023 年 11 月 27 日
添加了 Lambda 控制台和基础设施编排器集成	从 Lambda 控制台启动基础设施编排器画布。要了解更多信息，请参阅在 AWS Lambda 控制 AWS 基础架构编辑器 台中使用。	2023 年 11 月 14 日
使用基础设施编辑器将 Amazon VPC 添加为特色服务	基础设施编排器引入了 VPC 标签来可视化配置有 VPC 的资源。您也可以使用在外部模板上 VPCs 定义来配置 Lambda 函数。要了解更多信息，请参阅在 Amazon VPC 中使用基础设施编排器。	2023 年 10 月 17 日
通过基础设施编排器将 Amazon RDS 添加为特色服务	将您的 Infrastructure Composer 应用程序连接到 Amazon RDS 数据库集群或在外部模板上定义的实例。要了解更多信息，请参阅在 Amazon RDS 中使用基础设施编排器。	2023 年 10 月 17 日

为使用所有 CloudFormation 资源进行设计添加了基础架构编排支持	从 “ CloudFormation 资源” 选项板中选择任何资源来设计应用程序。要了解更多信息，请参阅使用任何 CloudFormation 资源。	2023 年 9 月 26 日
在基础架构编辑器中添加了卡片文档	Infrastructure Composer 支持多种类型的卡片，您可以使用这些卡片来设计和构建应用程序。要了解更多信息，请参阅基础设施编排器中的使用卡片进行设计。	2023 年 9 月 20 日
添加了有关撤消和重做功能的文档	使用基础设施编排器画布上的撤消和重做按钮。要了解更多信息，请参阅撤消和重做。	2023 年 8 月 1 日
添加了有关本地同步模式的文档	使用本地同步模式自动同步项目并将其保存到本地计算机。要了解更多信息，请参阅本地同步模式。	2023 年 8 月 1 日
添加了导出画布功能的文档	使用导出画布功能将应用程序的画布作为图像导出到本地计算机。要了解更多信息，请参阅导出画布。	2023 年 8 月 1 日
基础架构编排器支持外部文件引用	参考外部文件，了解基础设施编排器中支持的资源。要了解更多信息，请参阅使用引用外部文件的模板。	2023 年 5 月 17 日
有关连接资源的新文档	将资源连接在一起，定义应用程序中资源之间由事件驱动的关系。要了解更多信息，请参阅使用基础设施编排器可视化画布将资源连接在一起。	2023 年 3 月 7 日

全新的变更检查器功能	使用 Change Inspec tor 查看您的模板代码更新，并了解 Infrastructure Composer 正在为您创建什么。要了解更多信息，请参阅使用 Change Inspector 查看代码更新。	2023 年 3 月 7 日
基础架构编排器现已正式上市	AWS 基础架构编辑器 现已正式上市。要了解更多信息，请参阅AWS 基础架构编辑器 现已正式发布——快速直观地构建无服务器应用程序。	2023 年 3 月 7 日
扩展了使用连接模式的好处	在连接模式下使用本地 IDE 的基础架构编排器来加快开发速度。要了解更多信息，请参阅在本地 IDE 中使用基础架构编排器。	2023 年 3 月 7 日
更新了有关使用其他 AWS 服务部署应用程序的主题	使用基础架构编排器设计部署就绪型无服务器应用程序。AWS SAM 用于部署您的无服务器应用程序。要了解更多信息，请参阅将基础架构编排器与 CloudFormation 和配合使用 AWS SAM。	2023 年 3 月 3 日
添加了无服务器概念部分	在使用基础设施编排器之前，请先了解基本的无服务器概念。要了解更多信息，请参阅无服务器概念。	2023 年 3 月 2 日
公开发行	基础设施编排器的首次公开发布。	2022 年 12 月 1 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。