



Guia do Desenvolvedor

# AWS Infrastructure Composer



# AWS Infrastructure Composer: Guia do Desenvolvedor

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

---

# Table of Contents

O que é o Infrastructure Composer? .....	1
Componha sua arquitetura .....	2
Defina seus modelos .....	4
Integre com seus fluxos de trabalho .....	5
Formas de acessar o Infrastructure Composer .....	6
Saiba mais .....	8
Próximas etapas .....	8
Conceitos de tecnologia sem servidor .....	8
Conceitos de tecnologia sem servidor .....	9
Cartões .....	10
Placas de componentes aprimoradas .....	11
Exemplo .....	12
Placas de componentes padrão .....	13
Conexões de cartão .....	17
Conexões entre cartões .....	17
Conexões entre placas de componentes aprimoradas .....	18
Conexões de e para placas de recursos laC padrão .....	20
Conceitos básicos .....	21
Faça um tour pelo console .....	21
Próximas etapas .....	22
Carregar e modificar .....	22
Etapa 1: abrir a demonstração .....	23
Etapa 2: explorar a tela visual .....	23
Etapa 3: Expanda sua arquitetura .....	27
Etapa 4: Salve seu aplicativo .....	28
Próximas etapas .....	29
Compilar .....	29
Propriedades de recursos .....	30
Etapa 1: Crie seu projeto .....	30
Adicionar cartões .....	33
Etapa 3: configure seu REST API .....	34
Etapa 4: configure suas funções .....	35
Etapa 5: Conecte seus cartões .....	36
Etapa 6: organizar a tela .....	37

Adicionar uma tabela do DynamoDB .....	38
Etapa 8: revise seu modelo .....	39
Etapa 9: Integre-se aos seus fluxos de trabalho .....	40
Próximas etapas .....	40
Onde usar o Infrastructure Composer .....	41
Console Infrastructure Composer .....	41
Visão geral visual .....	42
Gerencie seu projeto .....	45
Conecte-se ao seu local IDE .....	48
Permitir acesso à página da web .....	51
Sincronize e salve localmente .....	52
Importar do console Lambda .....	56
Exportar tela .....	56
CloudFormation modo de console .....	58
Por que usar esse modo? .....	58
Acesse este modo .....	59
Visualize uma implantação .....	59
Crie um novo modelo .....	60
Atualizar uma pilha existente .....	61
AWS Toolkit for Visual Studio Code .....	63
Visão geral visual .....	64
Acesso a partir do VS Code .....	65
Sincronizar com Nuvem AWS .....	66
Compositor de infraestrutura com Amazon Q .....	68
Como compor .....	71
Coloque cartões na tela .....	71
Agrupe as cartas .....	72
Agrupamento de placas de componentes aprimoradas .....	72
Agrupando uma placa de componente padrão em outra .....	73
Cartões Connect .....	74
Conectando placas de componentes aprimoradas .....	75
Conectando cartões padrão .....	76
Exemplos .....	78
Placas de desconexão .....	80
Placas de componentes aprimoradas .....	80
Placas de componentes padrão .....	80



Organize os cartões .....	82
Configurar e modificar cartões .....	83
Cartões aprimorados .....	84
Cartões padrão .....	100
Excluir cartões .....	101
Placas de componentes aprimoradas .....	101
Placas de componentes padrão .....	101
Exibir atualizações de código .....	102
Benefícios do Change Inspector .....	103
Procedimento .....	103
Saiba mais .....	105
Arquivos externos de referência .....	106
Práticas recomendadas .....	107
Criar uma referência de arquivo externo .....	107
Carregar um projeto .....	108
Crie um aplicativo usando o AWS SAM CLI .....	109
Referenciar um OpenAPI Especificação .....	113
Integre com a Amazon VPC .....	116
Identifique recursos e informações .....	117
Configurar funções .....	123
Parâmetros em modelos importados .....	123
Adicionar novos parâmetros aos modelos importados .....	126
Configurar uma função Lambda com a VPC em outro modelo .....	127
Implemente na AWS nuvem .....	130
AWS SAM Conceitos importantes .....	130
Próximas etapas .....	130
Configure o AWS SAM CLI .....	131
Instale o AWS CLI .....	131
Instale o AWS SAM CLI .....	131
Acesse o AWS SAM CLI .....	131
Próximas etapas .....	132
Crie e implante .....	132
Excluir uma pilha .....	140
Solução de problemas .....	142
Mensagens de erro .....	142
“Não consigo abrir esta pasta” .....	142

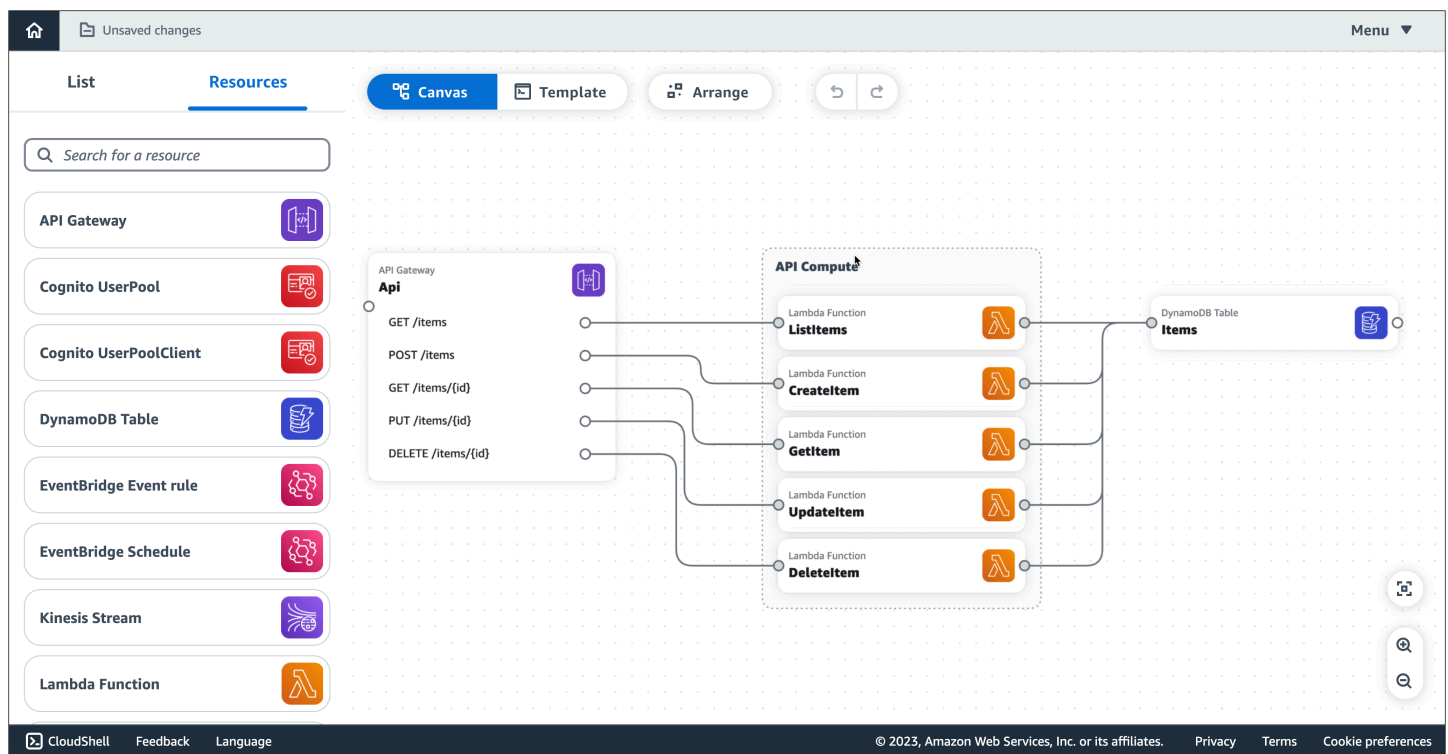
---

“Modelo incompatível” .....	142
“A pasta fornecida contém um template.yaml existente” .....	143
“Seu navegador não tem permissões para salvar seu projeto nessa pasta...” .....	143
Segurança .....	145
Proteção de dados .....	145
Criptografia de dados .....	147
Criptografia em trânsito .....	147
Gerenciamento de chaves .....	147
Privacidade do tráfego entre redes .....	147
AWS Identity and Access Management .....	147
Público .....	148
Autenticação com identidades .....	148
Como gerenciar acesso usando políticas .....	152
Como AWS Infrastructure Composer funciona com IAM .....	155
Validação de conformidade .....	161
Resiliência .....	162
Histórico de documentos .....	164
.....	clxx

# O que é AWS Infrastructure Composer?

AWS Infrastructure Composer permite que você componha visualmente aplicativos modernos em. AWS Mais especificamente, você pode usar o Infrastructure Composer para visualizar, criar e implantar aplicativos modernos de todos os AWS serviços que são suportados por, AWS CloudFormation sem precisar ser um especialista em. AWS CloudFormation

À medida que você compõe sua AWS CloudFormation infraestrutura, por meio de uma drag-and-drop interface agradável, o Infrastructure Composer cria sua infraestrutura como modelos de código (IaC), seguindo as melhores práticas. AWS A imagem a seguir mostra como é fácil arrastar, soltar, configurar e conectar recursos na tela visual do Infrastructure Composer.



O Infrastructure Composer pode ser usado no console do Infrastructure Composer AWS Toolkit for Visual Studio Code, no e no modo CloudFormation console.

## Tópicos

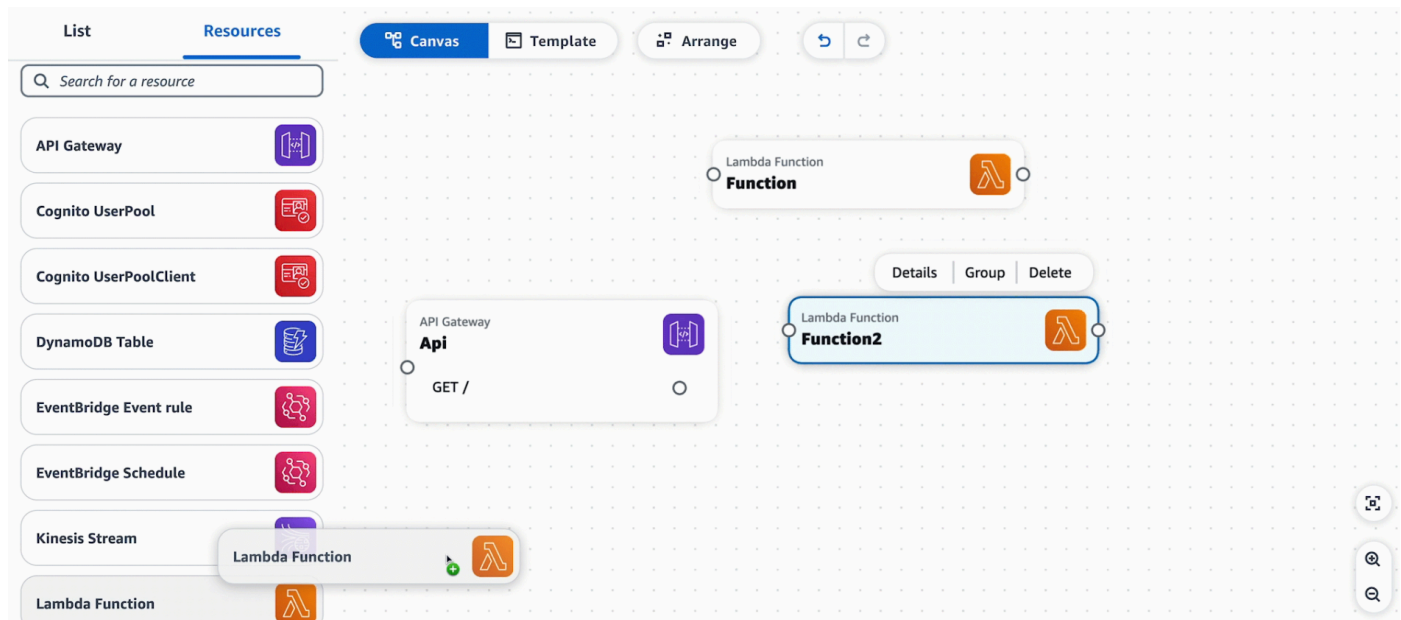
- [Componha sua arquitetura de aplicativos](#)
- [Defina sua infraestrutura como modelos de código \(IaC\)](#)
- [Integre com seus fluxos de trabalho existentes](#)
- [Formas de acessar o Infrastructure Composer](#)

- [Saiba mais](#)
- [Próximas etapas](#)
- [Conceitos sem servidor para AWS Infrastructure Composer](#)

## Componha sua arquitetura de aplicativos

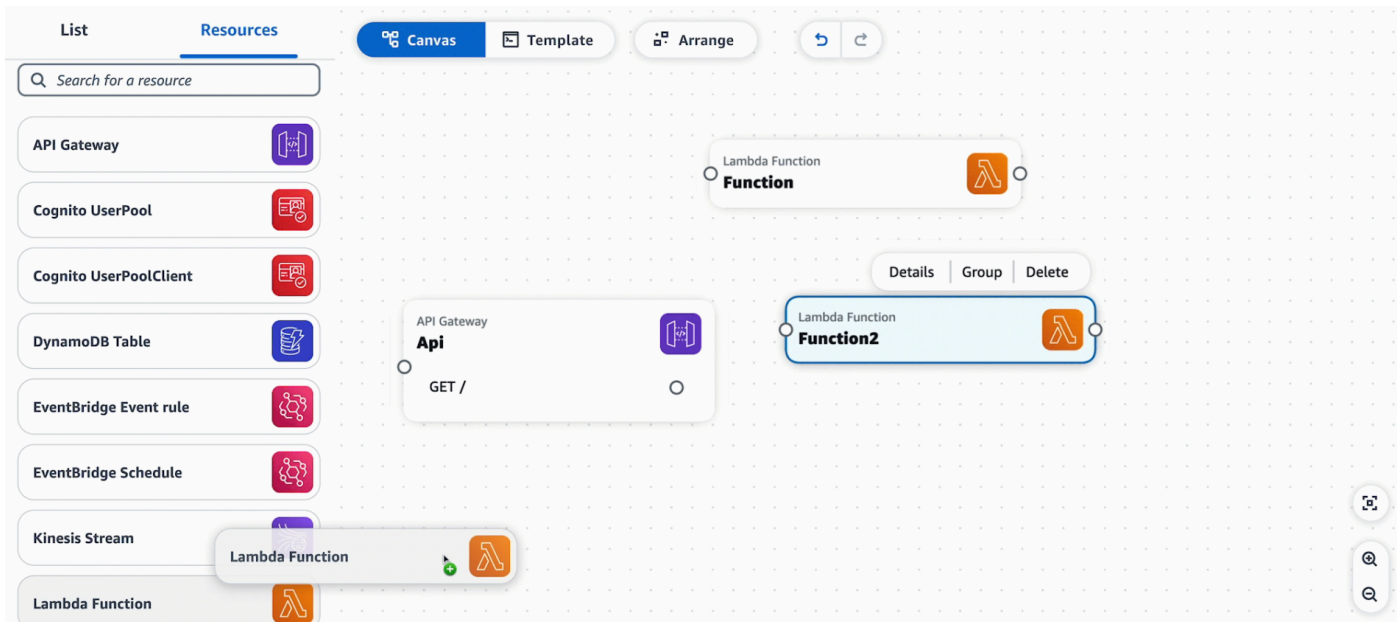
Construa com cartas

Coloque cartões na tela do Infrastructure Composer para visualizar e criar sua arquitetura de aplicativo.



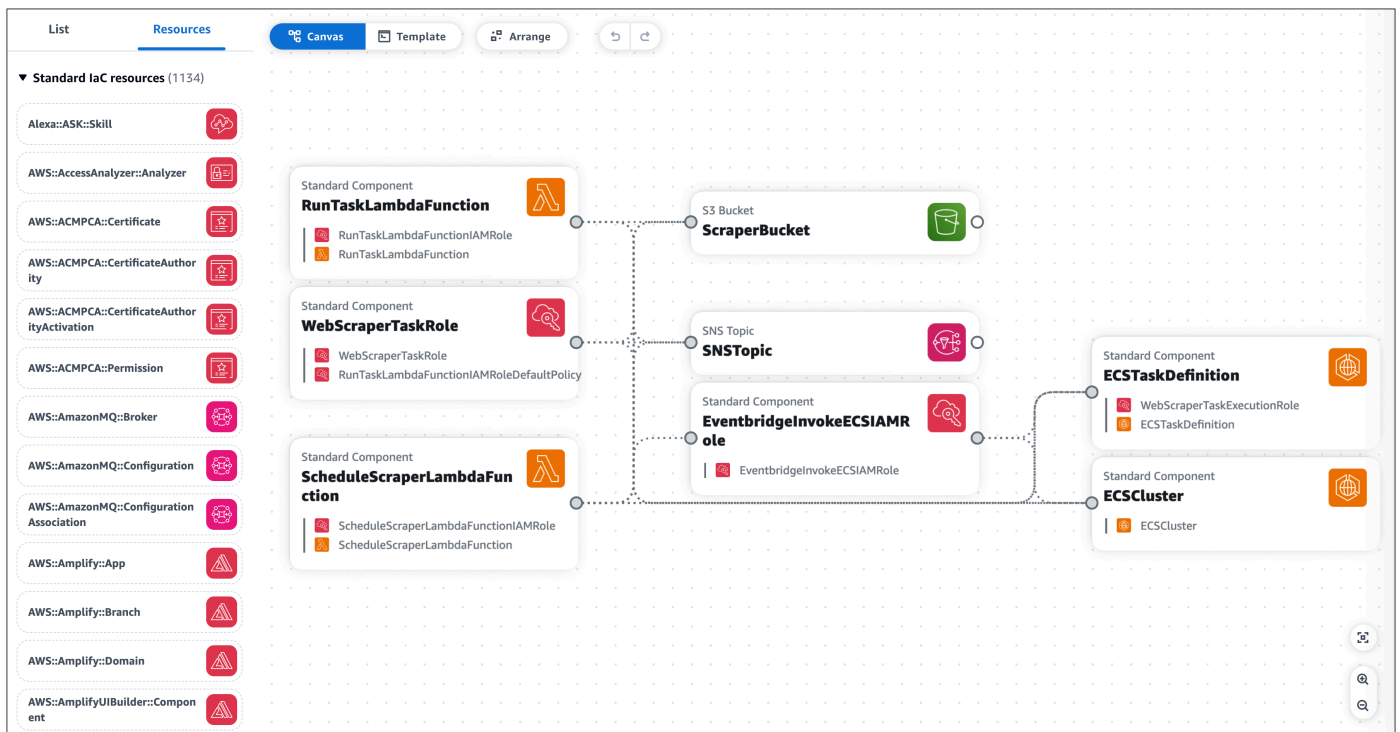
Conecte cartões juntos

Configure como seus recursos interagem entre si conectando-os visualmente. Especifique ainda mais suas propriedades por meio de um painel de propriedades com curadoria.



### Trabalhe com qualquer AWS CloudFormation recurso

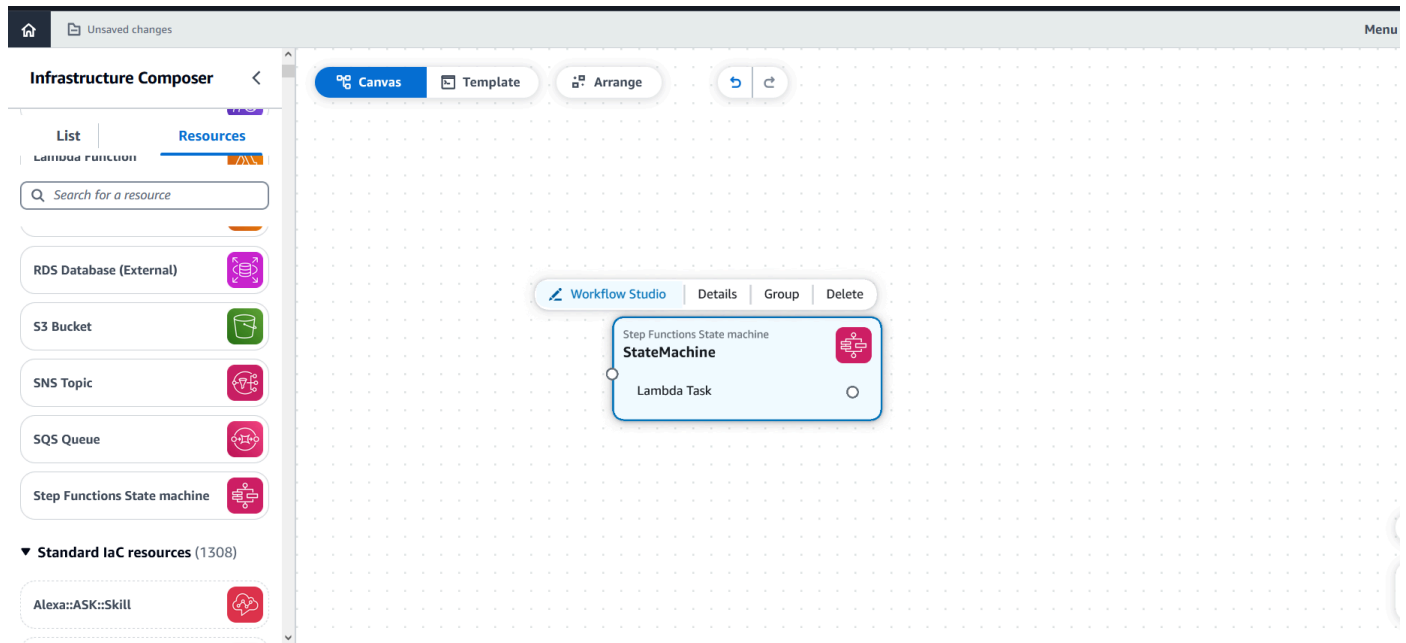
Arraste qualquer AWS CloudFormation recurso para a tela para compor a arquitetura do seu aplicativo. O Infrastructure Composer fornece um modelo inicial de IaC que você pode usar para especificar as propriedades do seu recurso. Para saber mais, consulte [Configurar e modificar cartões no Infrastructure Composer](#).



## Acesse recursos adicionais com recursos Serviços da AWS

Recursos do Infrastructure Composer Serviços da AWS que são comumente usados ou configurados juntos ao criar aplicativos. Para saber mais, consulte [Integre com a Amazon VPC](#).

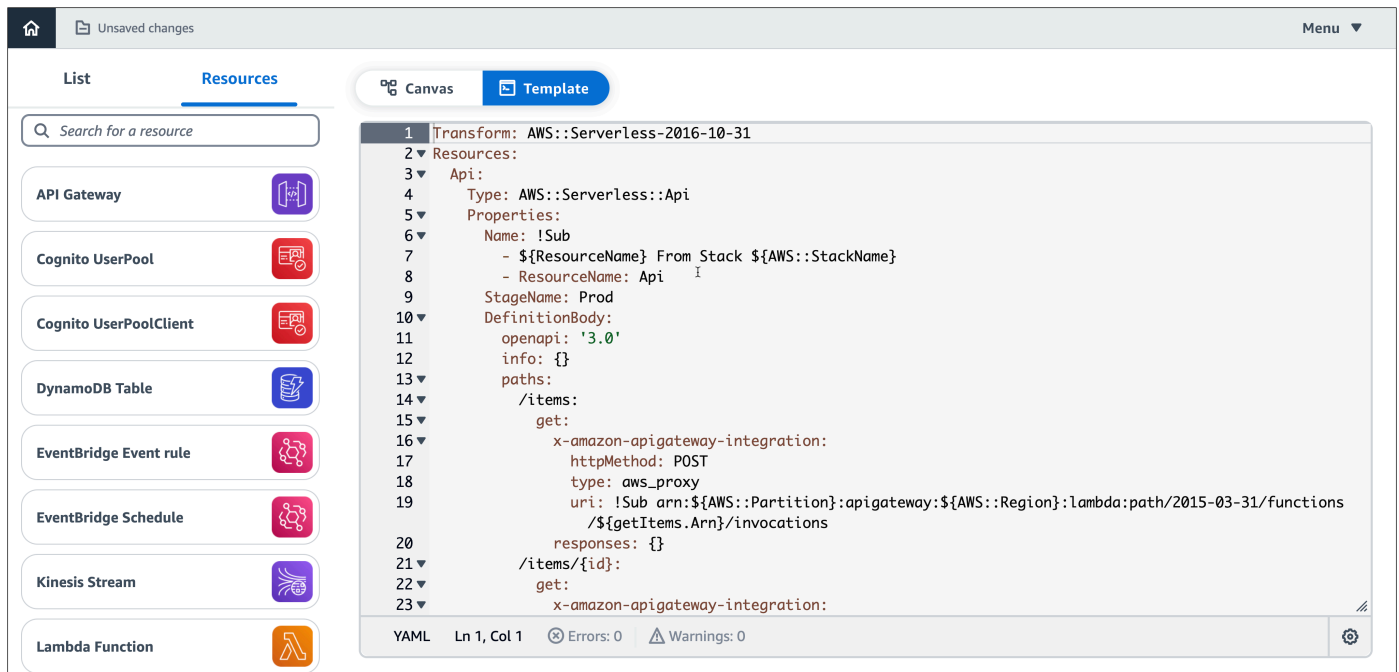
A seguir está um exemplo do AWS Step Functions recurso, que fornece uma integração para iniciar o Step Functions Workflow Studio diretamente na tela do Infrastructure Composer.



## Defina sua infraestrutura como modelos de código (IaC)

O Infrastructure Composer cria seu código de infraestrutura

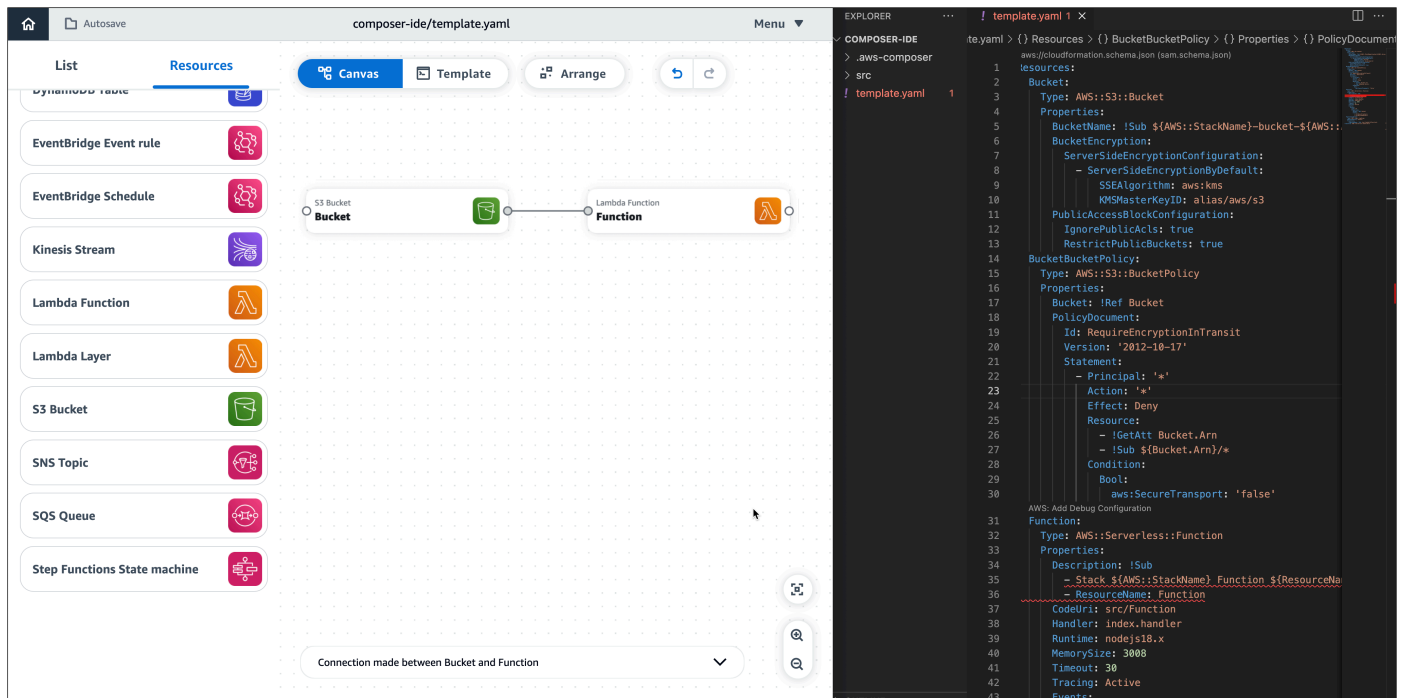
Conforme você compõe, o Infrastructure Composer cria automaticamente seus modelos AWS CloudFormation and AWS Serverless Application Model (AWS SAM), seguindo as AWS melhores práticas. Você pode visualizar e modificar seus modelos diretamente do Infrastructure Composer. O Infrastructure Composer sincroniza automaticamente as alterações entre a tela visual e o código do modelo.



## Integre com seus fluxos de trabalho existentes

### Importe modelos e projetos existentes

Importe AWS SAM modelos AWS CloudFormation e modelos existentes para visualizá-los para melhor entender e modificar seu design. Exporte os modelos que você cria no Infrastructure Composer e integre-os aos seus fluxos de trabalho existentes para implantação.

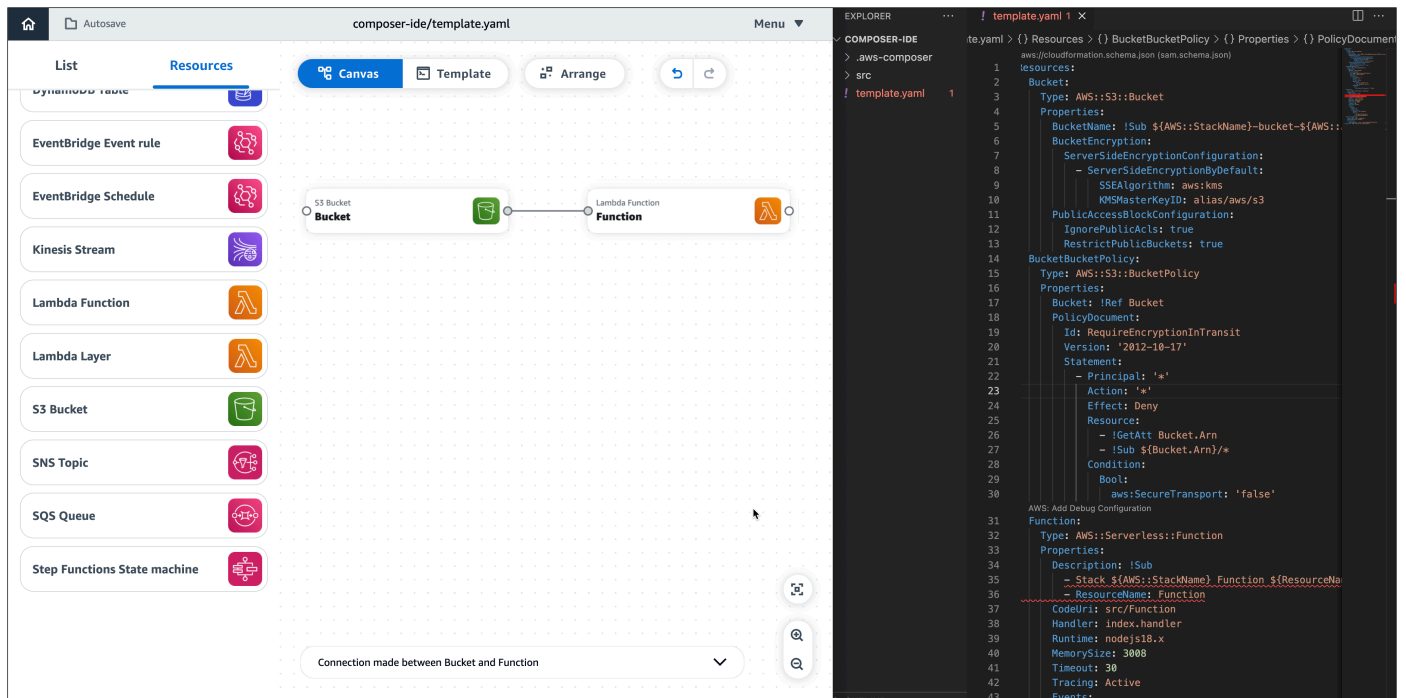


## Formas de acessar o Infrastructure Composer

### Do console do Infrastructure Composer

Acesse o Infrastructure Composer por meio do console do Infrastructure Composer para começar rapidamente. Além disso, você pode usar o modo de sincronização local para sincronizar e salvar automaticamente o Infrastructure Composer com sua máquina local.





## Do AWS CloudFormation console

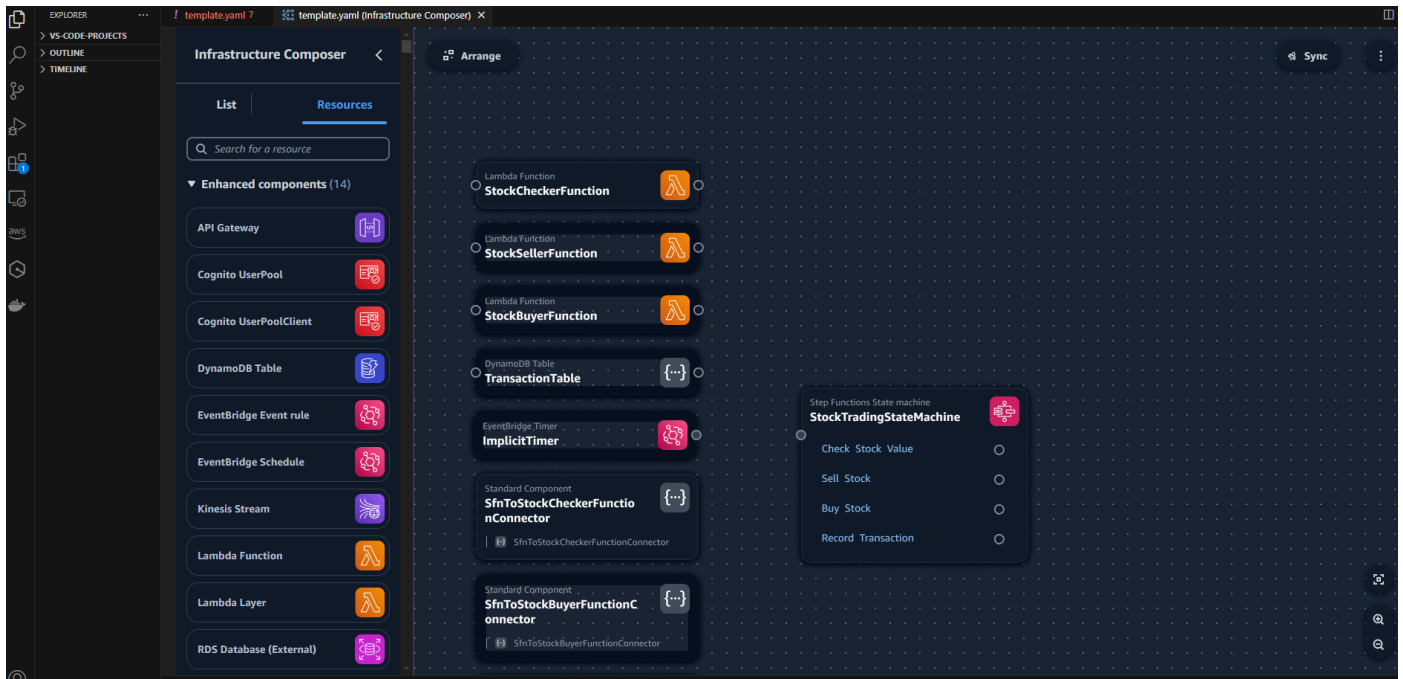
O console do Infrastructure Composer também suporta o [modo CloudFormation console](#), uma melhoria do CloudFormation Designer que é integrada ao fluxo de trabalho da AWS CloudFormation pilha. Essa nova ferramenta agora é a ferramenta recomendada para visualizar seus CloudFormation modelos.

## Do console Lambda

Com o Infrastructure Composer, você também pode importar funções Lambda do console Lambda. Para saber mais, consulte [Importe funções para o Infrastructure Composer a partir do console Lambda](#).

## Do AWS Toolkit for Visual Studio Code

Acesse o Infrastructure Composer por meio da extensão Toolkit for VS Code para levar o Infrastructure Composer ao seu ambiente de desenvolvimento local.



## Saiba mais

Para continuar aprendendo sobre o Infrastructure Composer, consulte os seguintes recursos:

- [Placas Infrastructure Composer](#)
- [Componha e crie visualmente aplicativos sem servidor | Serverless Office Hours](#) — Visão geral e demonstração do Infrastructure Composer.

## Próximas etapas

Para configurar o Infrastructure Composer, consulte [Introdução ao console do Infrastructure Composer](#).

## Conceitos sem servidor para AWS Infrastructure Composer

Saiba mais sobre os conceitos básicos sem servidor antes de usar. [AWS Infrastructure Composer](#)

# Conceitos de tecnologia sem servidor

## Arquitetura orientada a eventos

Um aplicativo sem servidor consiste em AWS serviços individuais, como computação e Amazon DynamoDB AWS Lambda para gerenciamento de banco de dados, cada um desempenhando uma função especializada. Esses serviços são então vagamente integrados entre si por meio de uma arquitetura orientada por eventos. Para saber mais sobre a arquitetura orientada por eventos, consulte [O que é uma arquitetura orientada por eventos?](#).

## infraestrutura como código (IaC)

A infraestrutura como código (IaC) é uma forma de tratar a infraestrutura da mesma forma que os desenvolvedores tratam o código, aplicando o mesmo rigor do desenvolvimento do código do aplicativo ao provisionamento da infraestrutura. Você define sua infraestrutura em um arquivo de modelo AWS, a implanta e AWS cria os recursos para você. ComIaC, você define no código o que deseja AWS provisionar. Para obter mais informações, consulte [Infraestrutura como código](#) na Introdução ao DevOps AWS AWS Whitepaper.

## Tecnologias sem servidor

Com tecnologias AWS sem servidor, você pode criar e executar aplicativos sem precisar gerenciar seus próprios servidores. Todo o gerenciamento do servidor é feito por meio de vários benefícios AWS, como escalabilidade automática e alta disponibilidade incorporada, permitindo que você leve sua ideia à produção rapidamente. Usando tecnologias sem servidor, você pode se concentrar no núcleo do seu produto sem precisar se preocupar com o gerenciamento e a operação de servidores. Para saber mais sobre a tecnologia sem servidor, consulte Sem [servidor](#) ativado. AWS

Para uma introdução básica aos principais serviços sem servidor, consulte AWS Serverless [101: Understanding the serverless services at Serverless](#) Land.

# Placas Infrastructure Composer

O Infrastructure Composer simplifica o processo de escrever infraestrutura como código (IaC) para recursos. AWS CloudFormation [Para usar efetivamente o Infrastructure Composer, há dois conceitos básicos que você deve entender primeiro: placas e conexões de placas do Infrastructure Composer.](#)

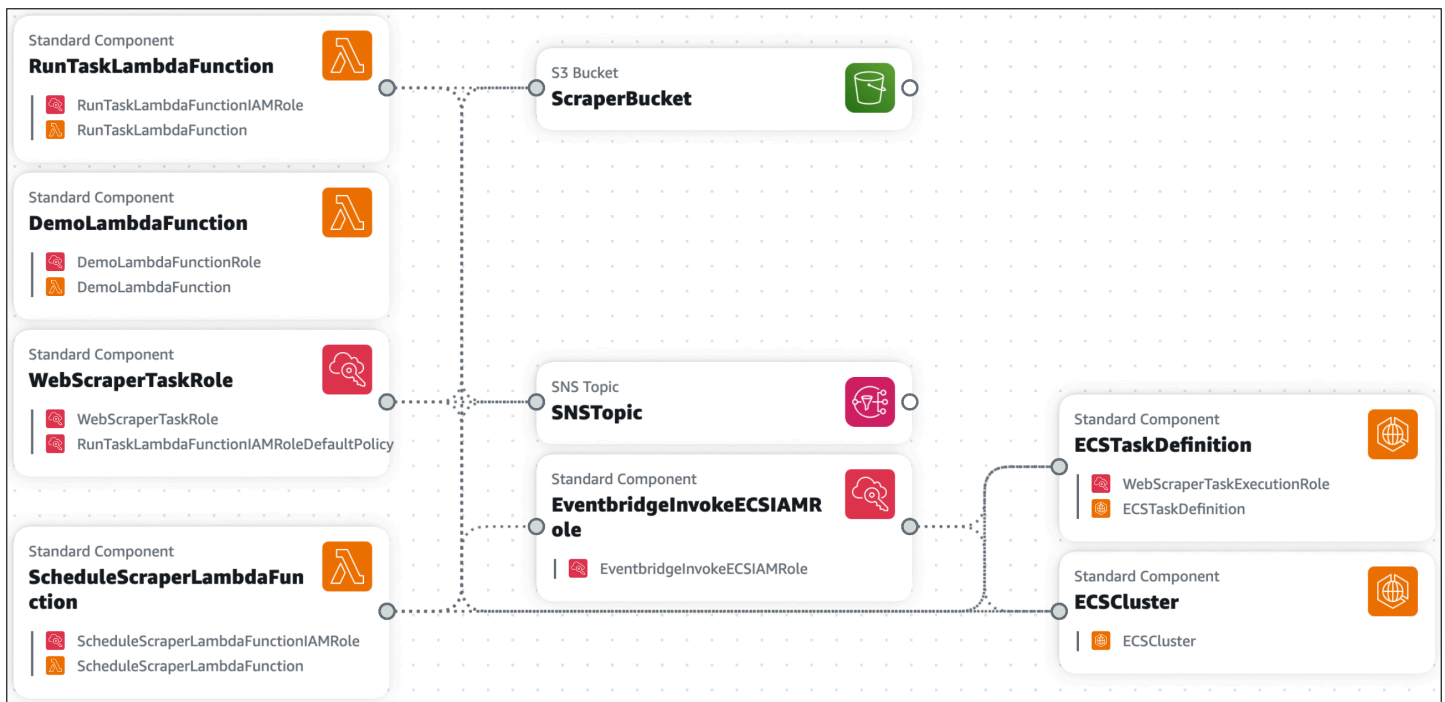
No Infrastructure Composer, os cartões representam AWS CloudFormation recursos. Há duas categorias gerais de cartões:

- [Cartão de componentes aprimorado](#) — Uma coleção de AWS CloudFormation recursos que foram combinados em um único cartão selecionado que aprimora a facilidade de uso e a funcionalidade e foi projetado para uma ampla variedade de casos de uso. As placas de componentes aprimoradas são as primeiras cartas listadas na paleta Resources no Infrastructure Composer.
- [Cartão de recursos padrão do IaC](#) — Um único AWS CloudFormation recurso. Cada cartão de recurso padrão do IaC, uma vez arrastado para a tela, é rotulado como componente padrão e pode ser combinado em vários recursos.

## Note

Dependendo do cartão, um cartão de recursos padrão do IaC pode ser rotulado como um cartão de componente padrão depois de ser arrastado para a tela visual. Isso significa simplesmente que o cartão é uma coleção de um ou mais cartões de recursos IaC padrão.

Embora alguns tipos de cartões estejam disponíveis na paleta Resources, os cartões também podem aparecer na tela quando você importa um modelo existente AWS CloudFormation ou AWS Serverless Application Model (AWS SAM) para o Infrastructure Composer. A imagem a seguir é um exemplo de um aplicativo importado que contém vários tipos de cartão:



## Tópicos

- [Placas de componentes aprimoradas no Infrastructure Composer](#)
- [Placas de componentes padrão no Infrastructure Composer](#)
- [Conexões de cartão no Infrastructure Composer](#)

## Placas de componentes aprimoradas no Infrastructure Composer

As placas de componentes aprimoradas são criadas e gerenciadas pelo Infrastructure Composer. Cada cartão contém AWS CloudFormation recursos que geralmente são usados juntos ao criar aplicativos AWS. Seu código de infraestrutura é criado pelo Infrastructure Composer seguindo as AWS melhores práticas. As placas de componentes aprimoradas são uma ótima maneira de começar a projetar seu aplicativo.

As placas de componentes aprimoradas estão disponíveis na paleta Recursos, na seção Componentes aprimorados.

As placas de componentes aprimoradas podem ser totalmente configuradas e usadas no Infrastructure Composer para projetar e criar seus aplicativos sem servidor. Recomendamos o uso de placas de componentes aprimoradas ao projetar seus aplicativos sem código existente.

Esta tabela mostra nossos componentes aprimorados com links para a especificação do modelo AWS CloudFormation ou AWS Serverless Application Model (AWS SAM) do recurso em destaque do cartão:

Cartão	Referência
Amazon API Gateway	<a href="#">AWS: :Sem servidor:: API</a>
Amazon Cognito UserPool	<a href="#">AWS: :Cognito:: UserPool</a>
Amazon Cognito UserPoolClient	<a href="#">AWS: :Cognito:: UserPoolClient</a>
Tabela do Amazon DynamoDB	<a href="#">AWS: :DynamoDB: :Tabela</a>
Regra de EventBridge eventos da Amazon	<a href="#">AWS: :Eventos: :Regra</a>
EventBridge Cronograma	<a href="#">AWS: :Agenda: :Agenda</a>
Stream do Amazon Kinesis	<a href="#">AWS: :Kinesis: :Stream</a>
AWS Lambda Função	<a href="#">AWS: :Serverless: :Função</a>
Camada Lambda	<a href="#">AWS: :Sem servidor:: LayerVersion</a>
Bucket do Amazon Simple Storage Service (Amazon S3)	<a href="#">AWS: :S3: :Balde</a>
Tópico do Amazon Simple Notification Service (AmazonSNS)	<a href="#">AWS::SNS: :Tópico</a>
Fila do Amazon Simple Queue Service (AmazonSQS)	<a href="#">AWS::SQS: :Fila</a>
AWS Step Functions Máquina de estado	<a href="#">AWS: :Sem servidor:: StateMachine</a>

## Exemplo

Veja a seguir um exemplo de um componente aprimorado do S3 Bucket:



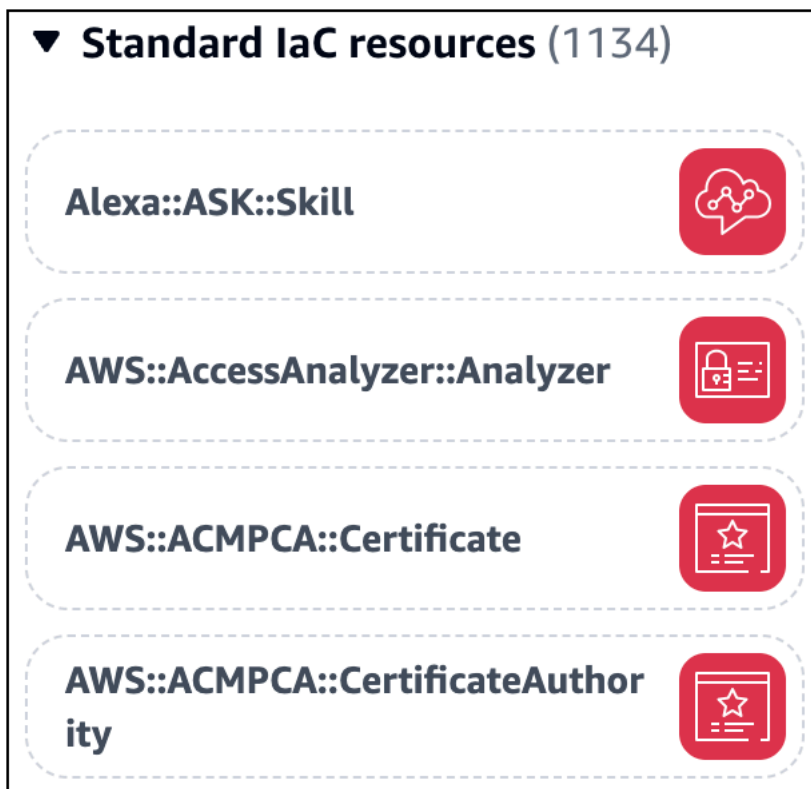
Ao arrastar um cartão de componente do S3 Bucket para a tela e visualizar seu modelo, você verá os dois AWS CloudFormation recursos a seguir adicionados ao seu modelo:

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

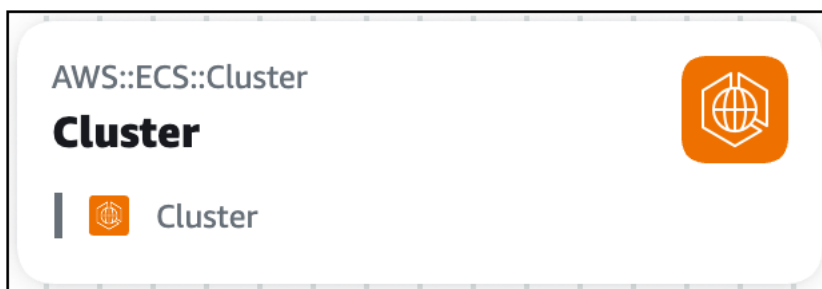
A placa de componente aprimorada do S3 Bucket representa dois AWS CloudFormation recursos que são necessários para que um bucket do Amazon Simple Storage Service (Amazon S3) interaja com outros serviços em seu aplicativo.

## Placas de componentes padrão no Infrastructure Composer

Antes de uma placa de componente padrão ser colocada na tela visual do Infrastructure Composer, ela é listada como uma placa de recurso Standard (IaC) na paleta Resources no Infrastructure Composer. Um cartão de recursos padrão (IaC) representa um único AWS CloudFormation recurso. Cada cartão de recursos padrão do IaC, uma vez colocado na tela visual, se torna um cartão rotulado como componente padrão e pode ser combinado para representar vários AWS CloudFormation recursos.



Cada cartão de recursos padrão do IaC pode ser identificado por seu tipo AWS CloudFormation de recurso. Veja a seguir um exemplo de um cartão de recursos padrão do IaC que representa um tipo de `AWS::ECS::Cluster` AWS CloudFormation recurso:

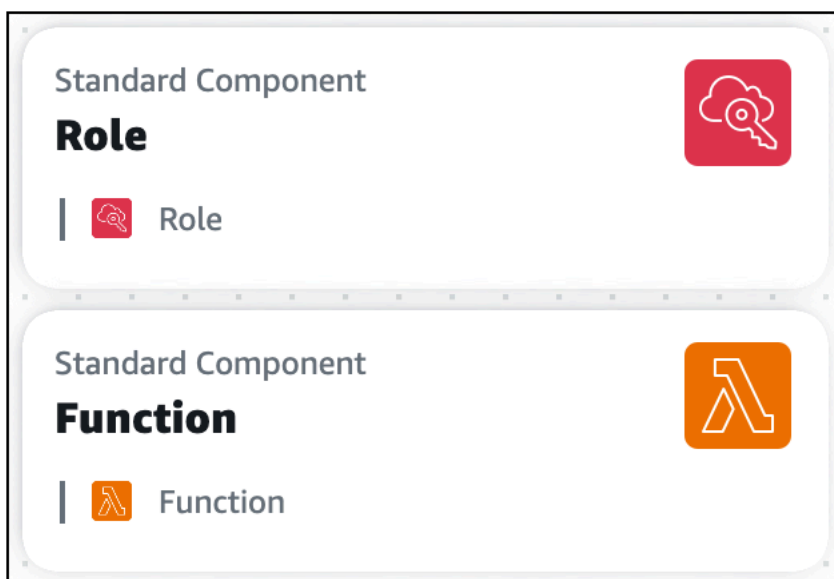


Cada placa de componente padrão visualiza os AWS CloudFormation recursos que ela contém. Veja a seguir um exemplo de uma placa de componente padrão que inclui dois recursos padrão de IaC:






Conforme você configura as propriedades de suas placas de componentes padrão, o Infrastructure Composer pode combinar placas relacionadas. Por exemplo, aqui estão duas placas de componentes padrão:



No painel Propriedades do recurso da placa de componente padrão que representa um `AWS::Lambda::Function` recurso, referenciamos a função AWS Identity and Access Management (IAM) por sua ID lógica:

**Resource properties** ✕

 **AWS::Lambda::Function**  
CFN Resource


**Editing** Function ▾

**Logical ID**  
Updating this value will generate a new resource when this stack is updated.

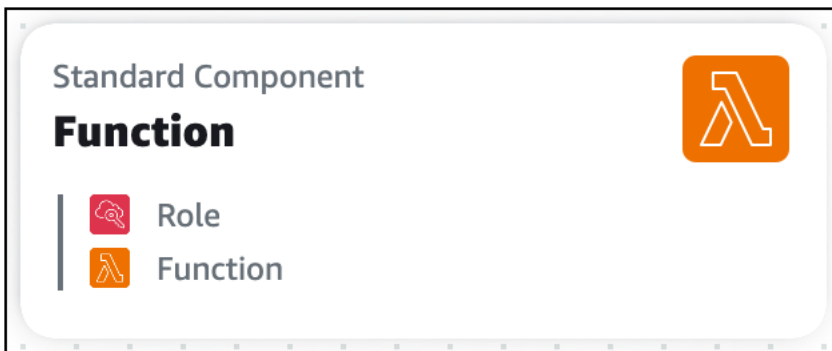
Function

**Resource configuration**  
Updating this value will change the resource's properties. Replace all placeholder values before deploying.

```
Code: {}
Role: !Ref Role
```

[Resource reference](#) 

Depois de salvar nosso modelo, as duas placas de componentes padrão se combinam em uma única placa de componente padrão.



## Conexões de cartão no Infrastructure Composer

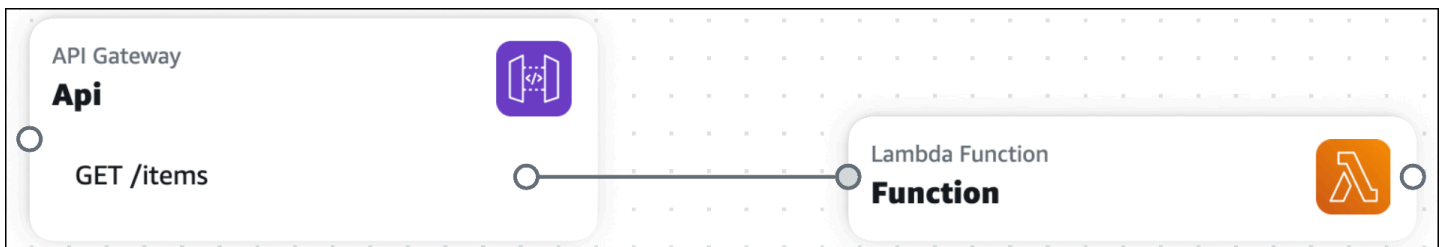
Em AWS Infrastructure Composer, uma conexão entre duas placas é exibida visualmente por uma linha. Essas linhas representam relacionamentos orientados por eventos em seu aplicativo.

Tópicos

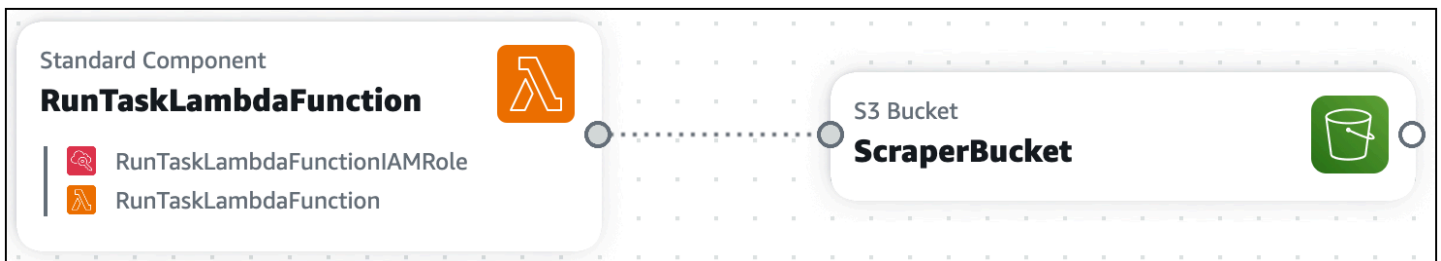
- [Conexões entre cartões](#)
- [Conexões entre placas de componentes aprimoradas](#)
- [Conexões de e para placas de recursos IaC padrão](#)

### Conexões entre cartões

A forma como você conecta as placas varia de acordo com o tipo de placa. Cada placa aprimorada tem pelo menos uma porta de conector. Para conectá-los, basta selecionar uma porta de conector e arrastá-la para a porta de outra placa, e o Infrastructure Composer conectará os dois recursos ou exibirá uma mensagem informando que essa configuração não é suportada.



Como visto acima, as linhas entre as placas de componentes aprimoradas são sólidas. Por outro lado, as placas de recursos IaC padrão (também chamadas de placas de componentes padrão) não têm portas de conector. Para esses cartões, você deve especificar esses relacionamentos orientados por eventos no modelo do seu aplicativo, e o Infrastructure Composer detectará automaticamente suas conexões e as visualizará com uma linha pontilhada entre seus cartões.

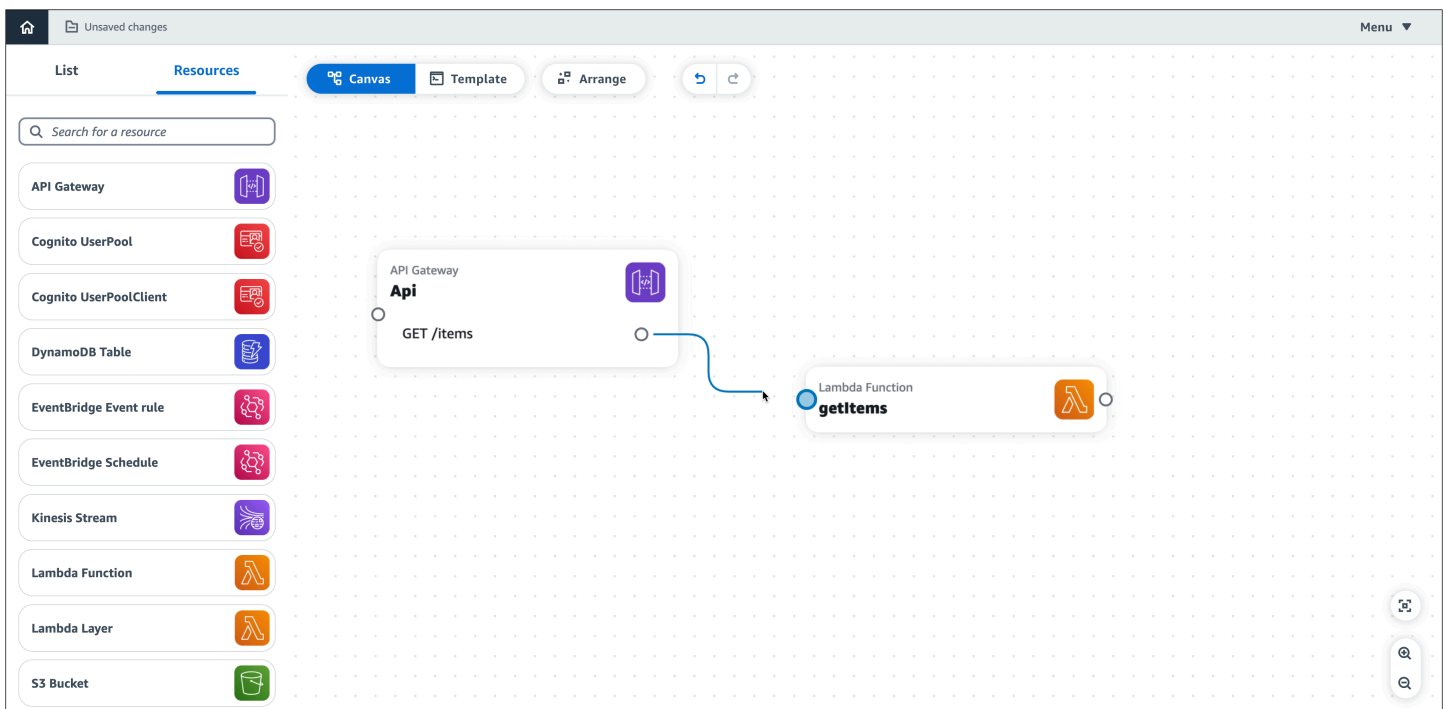


Para saber mais, consulte as seções abaixo.

## Conexões entre placas de componentes aprimoradas

No Infrastructure Composer, uma conexão entre duas placas de componentes aprimoradas é exibida visualmente por uma linha sólida. Essas linhas representam relacionamentos orientados por eventos em seu aplicativo.

Para conectar duas placas, clique em uma porta de uma placa e arraste-a para uma porta em outra placa.



### Note

As placas de recursos IaC padrão não têm portas de conector. Para esses cartões, você deve especificar seus relacionamentos orientados por eventos no modelo do seu aplicativo,

e o Infrastructure Composer detectará automaticamente suas conexões e as visualizará com uma linha pontilhada entre seus cartões.

Para obter mais informações, consulte [Cartões Connect na tela visual do Infrastructure Composer](#).

## O que as placas de componentes aprimoradas fornecem

As conexões entre duas placas, indicadas visualmente por uma linha, fornecem o seguinte quando necessário:

- AWS Identity and Access Management (IAM) políticas
- Variáveis de ambiente
- Eventos

### Políticas do IAM

Quando um recurso precisa de permissão para invocar outro recurso, o Infrastructure Composer provisiona políticas baseadas em recursos usando modelos de política AWS Serverless Application Model (AWS SAM).

- Para saber mais sobre IAM permissões e políticas, consulte [Visão geral do gerenciamento de acesso: permissões e políticas](#) no Guia do IAM usuário.
- Para saber mais sobre os modelos AWS SAM de política, consulte os [modelos de AWS SAM política](#) no Guia do AWS Serverless Application Model desenvolvedor.

### Variáveis de ambiente

As variáveis de ambiente são valores temporários que podem ser alterados para afetar o comportamento de seus recursos. Quando necessário, o Infrastructure Composer define o código da infraestrutura para utilizar variáveis de ambiente entre os recursos.

### Eventos

Os recursos podem invocar outro recurso por meio de diferentes tipos de eventos. Quando necessário, o Infrastructure Composer define o código de infraestrutura necessário para que os recursos interajam por meio de tipos de eventos.

## Conexões de e para placas de recursos IaC padrão

Todos os AWS CloudFormation recursos estão disponíveis para uso como cartões de recursos padrão do IaC na paleta Recursos. Quando você arrasta um cartão de recursos padrão do IaC para a tela, um cartão de recurso padrão do IaC se torna um cartão de componente padrão, e isso solicita que o Infrastructure Composer crie um modelo inicial para seu recurso em seu aplicativo.

Para obter mais informações, consulte [Cartões padrão no Infrastructure Composer](#).

# Introdução ao console do Infrastructure Composer

Use os tópicos desta seção para configurar AWS Infrastructure Composer e aprender como criar um aplicativo usando sua tela visual. O tour e os tutoriais desta seção são mostrados no console do Infrastructure Composer, que é a experiência padrão do usuário. Os tópicos desta seção mostram como preencher os pré-requisitos para usar o Infrastructure Composer, usar o console do Infrastructure Composer, carregar e modificar um projeto e criar seu primeiro aplicativo.

O Infrastructure Composer também está disponível no AWS Toolkit for Visual Studio Code e no modo CloudFormation console. As experiências entre as ferramentas geralmente são as mesmas, mas existem algumas diferenças entre cada uma. Para obter detalhes sobre o uso do Infrastructure Composer em cada uma dessas ferramentas, consulte [Onde você pode usar o Infrastructure Composer](#).

## Tópicos

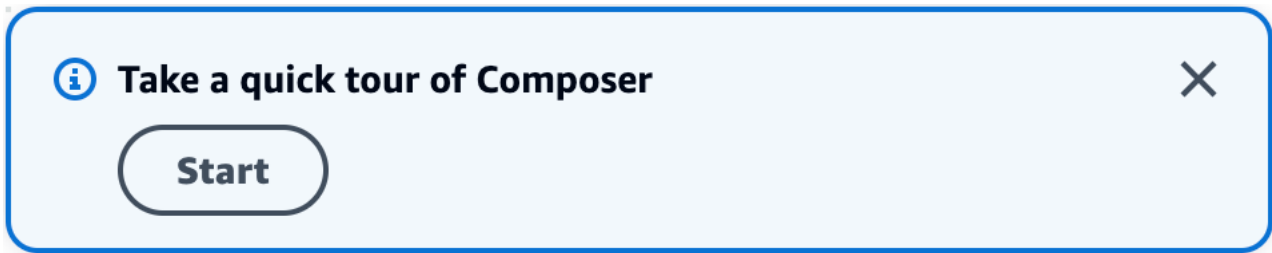
- [Faça um tour no console do Infrastructure Composer](#)
- [Carregue e modifique o projeto de demonstração do Infrastructure Composer](#)
- [Crie seu primeiro aplicativo com o Infrastructure Composer](#)

## Faça um tour no console do Infrastructure Composer

Para ter uma ideia geral de como AWS Infrastructure Composer funciona, faça o tour incorporado ao console do Infrastructure Composer. Para obter uma visão geral do console do Infrastructure Composer, consulte [Faça um tour no console do Infrastructure Composer](#). Para obter orientações detalhadas sobre o uso do Infrastructure Composer, consulte [Como compor em AWS Infrastructure Composer](#).

Para fazer um tour pelo Infrastructure Composer

1. Faça login no [console do Infrastructure Composer](#).
2. Na página inicial, escolha Abrir demonstração.
3. No canto superior direito, na janela Faça um tour rápido pelo Composer, escolha Iniciar.



4. Na janela do tour do Composer, faça o seguinte:
  - Para ir para a próxima etapa, escolha Avançar.
  - Para retornar à etapa anterior, escolha Anterior.
  - Na etapa final, para finalizar o passeio, escolha Encerrar.

O tour fornece uma breve visão geral da funcionalidade básica do Infrastructure Composer, como usar, configurar e conectar placas. Para obter mais informações, consulte [Como compor em AWS Infrastructure Composer](#).

## Próximas etapas

Para carregar e modificar um projeto no Infrastructure Composer, consulte [Carregue e modifique o projeto de demonstração do Infrastructure Composer](#).

## Carregue e modifique o projeto de demonstração do Infrastructure Composer

Use este tutorial para se familiarizar com a interface de usuário do Infrastructure Composer e aprender como carregar, modificar e salvar o projeto de demonstração do Infrastructure Composer.

Este tutorial é feito no console do Infrastructure Composer. Depois de concluído, você estará pronto para começar [Crie seu primeiro aplicativo com o Infrastructure Composer](#).

### Tópicos

- [Etapa 1: abrir a demonstração](#)
- [Etapa 2: Explore a tela visual do Infrastructure Composer](#)
- [Etapa 3: Expandir sua arquitetura de aplicativos](#)
- [Etapa 4: Salve seu aplicativo](#)
- [Próximas etapas](#)



## Etapa 1: abrir a demonstração

Comece a usar o Infrastructure Composer criando um projeto de demonstração.

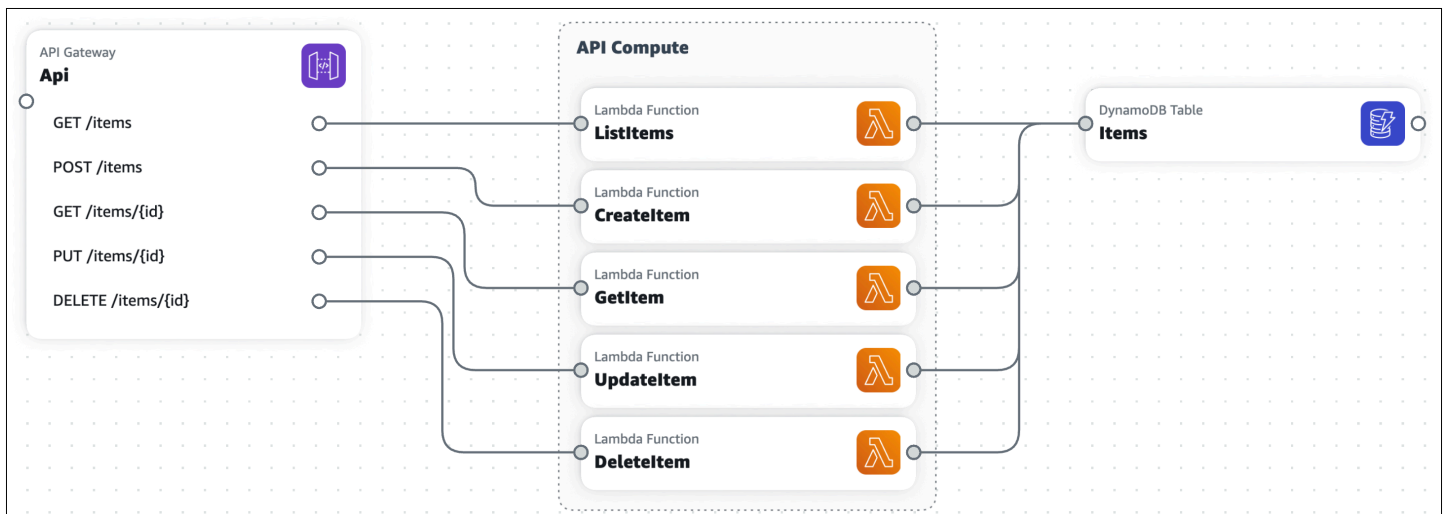
Para criar um projeto de demonstração

1. Faça login no [console do Infrastructure Composer](#).
2. Na página inicial, escolha Abrir demonstração.

O aplicativo de demonstração é um aplicativo básico sem servidor para criar, ler, excluir e atualizar (CRUD) que inclui:

- Um recurso do Amazon API Gateway com cinco rotas.
- Cinco AWS Lambda funções.
- Uma tabela do Amazon DynamoDB.

A imagem a seguir é da demonstração:

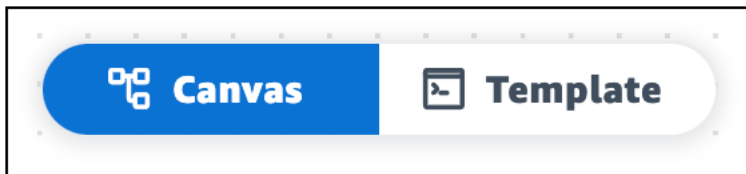


## Etapa 2: Explore a tela visual do Infrastructure Composer

Conheça os recursos da tela visual para criar seu projeto de demonstração do Infrastructure Composer. Para obter uma visão geral do layout da tela visual, consulte [Visão geral visual](#).

## Para explorar as características da tela visual

1. Quando você abre um projeto de aplicativo novo ou existente, o Infrastructure Composer carrega a visualização em tela, conforme indicado acima da área de visualização principal.

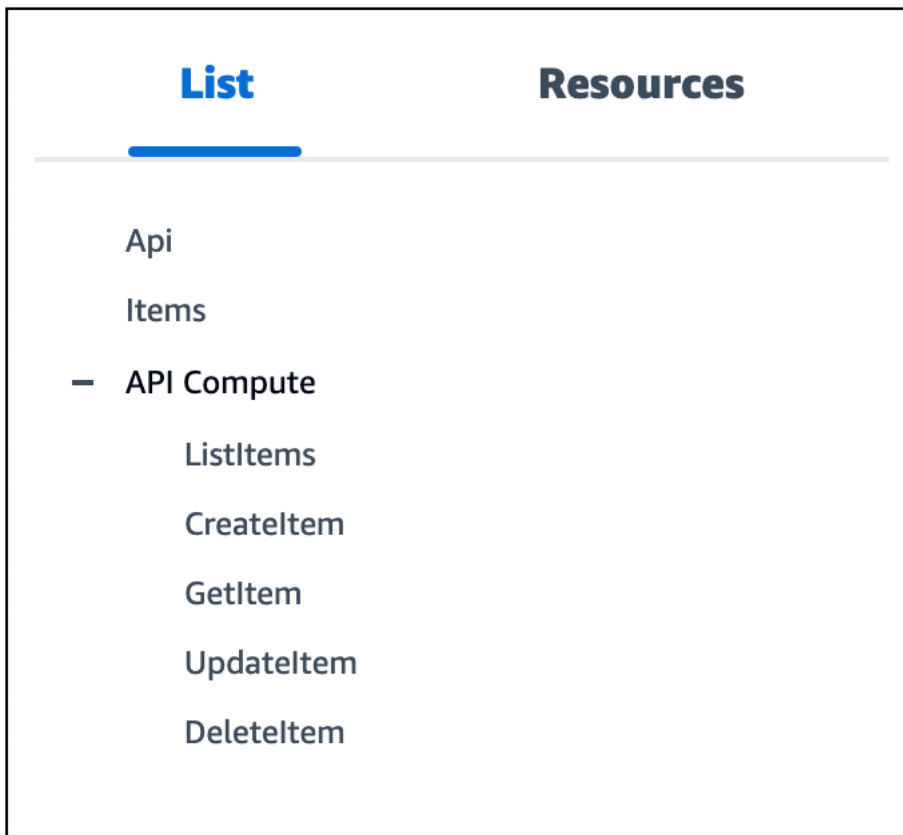


Para mostrar o código de infraestrutura do seu aplicativo na área de exibição principal, escolha Modelo. Por exemplo, aqui está a visualização do modelo AWS Serverless Application Model (AWS SAM) do projeto de demonstração do Infrastructure Composer.

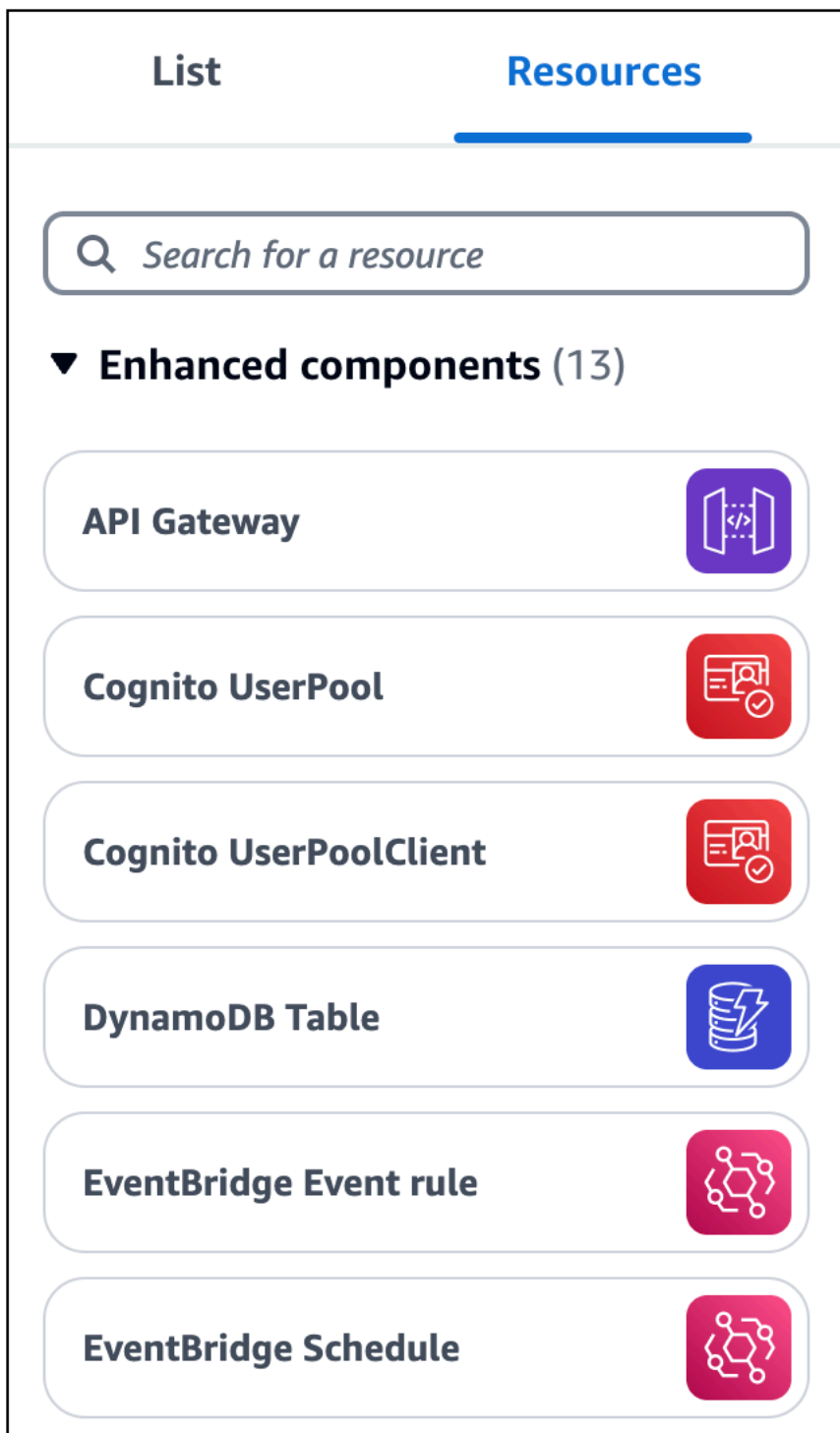
A screenshot of the Infrastructure Composer interface. At the top, there are two buttons: 'Canvas' (blue) and 'Template' (white with blue border). Below the buttons is a code editor showing the AWS SAM JSON template. The code is as follows:

```
1 Transform: AWS::Serverless-2016-10-31
2 Resources:
3   Api:
4     Type: AWS::Serverless::Api
5     Properties:
6       Name: !Sub
7         - ${ResourceName} From Stack ${AWS::StackName}
8         - ResourceName: Api
9     StageName: Prod
10    DefinitionBody:
11      openapi: '3.0'
12      info: {}
13      paths:
14        /items:
15          get:
16            x-amazon-apigateway-integration:
17              httpMethod: POST
18              type: aws_proxy
19              uri: !Sub arn:${AWS::Partition}:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/${ListItems.Arn}/invocations
20              responses: {}
21          post:
22            x-amazon-apigateway-integration:
23              httpMethod: POST
24              type: aws_proxy
25              uri: !Sub arn:${AWS::Partition}:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/${CreateItem.Arn}/invocations
26              responses: {}
27        /items/{id}:
28          get:
29            x-amazon-apigateway-integration:
30              httpMethod: POST
31              type: aws_proxy
32              uri: !Sub arn:${AWS::Partition}:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/${GetItem.Arn}/invocations
33              responses: {}
34          put:
```

2. Para mostrar novamente a visualização em tela do seu aplicativo, escolha Canvas.
3. Para mostrar os recursos do seu aplicativo organizados em uma visualização em árvore, escolha Lista.



4. Para mostrar a paleta de recursos, escolha Recursos. Essa paleta apresenta cartões que você pode usar para expandir a arquitetura do seu aplicativo. Você pode pesquisar cartões ou percorrer a lista.



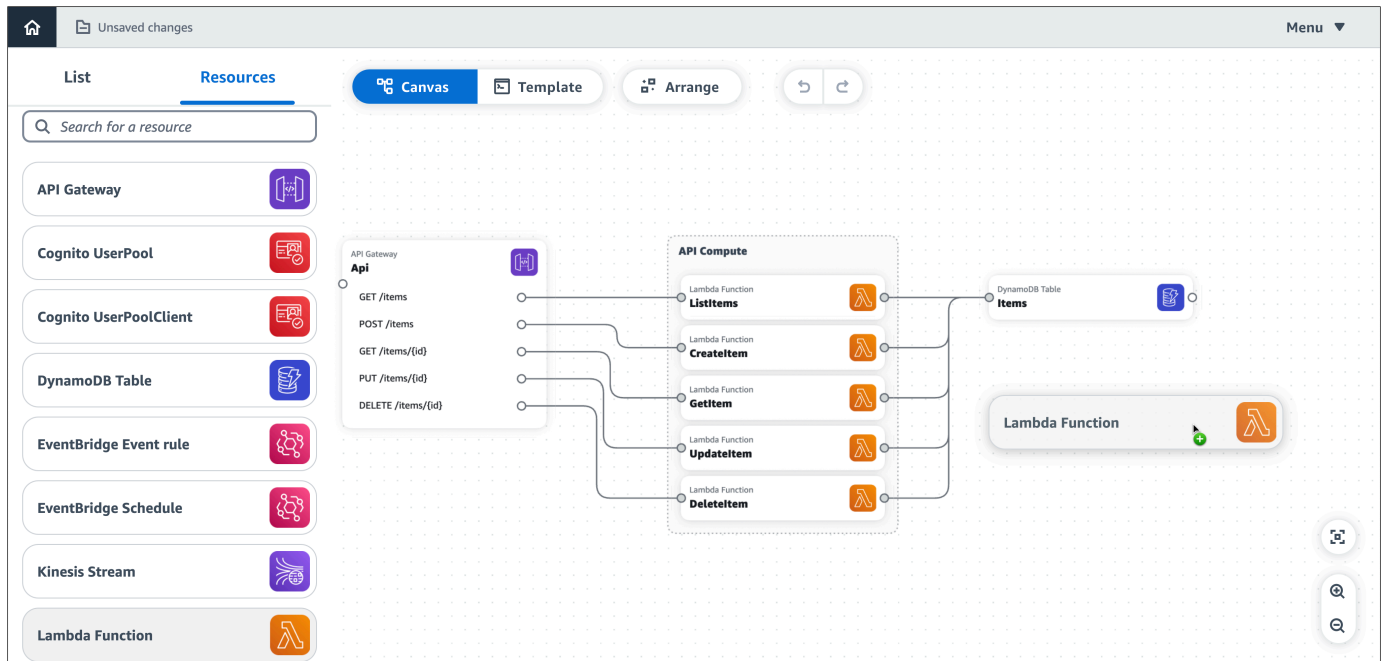
5. Para se mover pela tela visual, use gestos básicos. Para obter mais informações, consulte [Coloque cartões na tela](#).

## Etapa 3: Expandir sua arquitetura de aplicativos

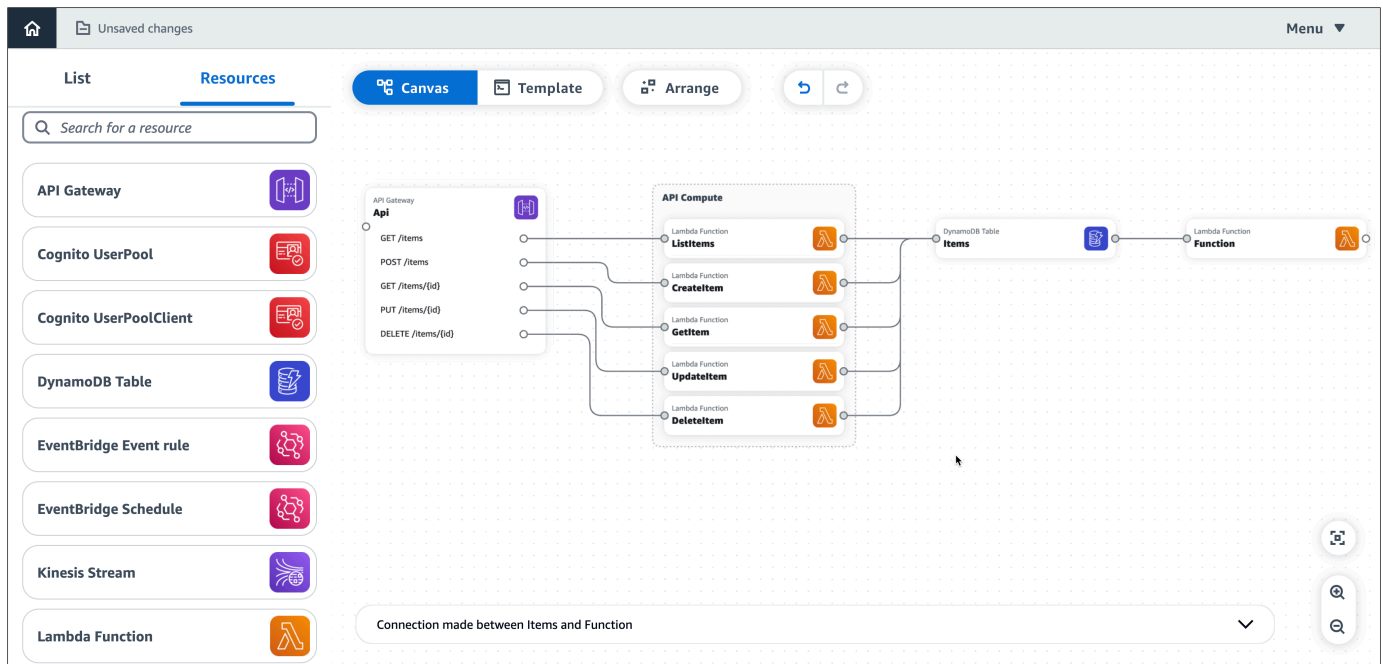
Nesta etapa, você expandirá a arquitetura do seu aplicativo adicionando uma função Lambda à sua tabela do DynamoDB.

Para adicionar uma função Lambda à sua tabela do DynamoDB

1. Na paleta de recursos (Recursos), arraste o cartão do componente aprimorado da Função Lambda até a tela, à direita do cartão de tabela do DynamoDB.



2. Conecte a tabela do DynamoDB à função Lambda. Para conectá-los, clique na porta direita da placa DynamoDB Table e arraste-a até a porta esquerda da placa Lambda Function.
3. Escolha Organizar para organizar os cartões na visualização em tela.



#### 4. Configure sua função Lambda. Para configurá-lo, faça o seguinte:

- Na visualização em tela, modifique as propriedades da função no painel Propriedades do recurso. Para abrir o painel, clique duas vezes no cartão da Função Lambda. Ou selecione o cartão e escolha Detalhes. Para obter mais informações sobre as propriedades configuráveis da função Lambda listadas no painel Propriedades do recurso, consulte o Guia [AWS Lambda do desenvolvedor](#).
- Na visualização do modelo, modifique o código da sua função (`AWS::Serverless::Function`). O Infrastructure Composer sincroniza automaticamente suas alterações na tela. Para obter mais informações sobre o recurso de função em um AWS SAM modelo, consulte [AWS::Serverless::Function](#) na referência de AWS SAM recursos e propriedades.

## Etapa 4: Salve seu aplicativo

Salve seu aplicativo salvando manualmente seu modelo de aplicativo em sua máquina local ou ativando a sincronização local.

Para salvar manualmente seu modelo de aplicativo

1. No menu, selecione Salvar > Salvar arquivo de modelo.
2. Forneça um nome para seu modelo e escolha um local em sua máquina local para salvá-lo. Pressione Salvar.

Para obter instruções sobre como ativar a sincronização local, consulte [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#).

## Próximas etapas

Para começar a criar seu primeiro aplicativo, consulte [Crie seu primeiro aplicativo com o Infrastructure Composer](#).

## Crie seu primeiro aplicativo com o Infrastructure Composer

Neste tutorial, você usa AWS Infrastructure Composer para criar, ler, atualizar e excluir (CRUD) um aplicativo sem servidor que gerencia usuários em um banco de dados.

Para este tutorial, usamos o Infrastructure Composer no AWS Management Console. Recomendamos que você use Google Chrome or Microsoft Edgee uma janela do navegador em tela cheia.

 Você é novato na tecnologia sem servidor?

Recomendamos uma compreensão básica do seguinte:

- [Arquitetura orientada a eventos](#)
- [infraestrutura como código \(IaC\)](#)
- [Tecnologias sem servidor](#)

Para saber mais, consulte [Conceitos sem servidor para AWS Infrastructure Composer](#).

## Tópicos

- [Referência de propriedades do recurso](#)
- [Etapa 1: Crie seu projeto](#)
- [Etapa 2: adicionar cartões à tela](#)
- [Etapa 3: Configurar seu API gateway REST API](#)
- [Etapa 4: Configurar suas funções do Lambda](#)
- [Etapa 5: Conecte seus cartões](#)

- [Etapa 6: organizar a tela](#)
- [Etapa 7: Adicionar e conectar uma tabela do DynamoDB](#)
- [Etapa 8: revise seu AWS CloudFormation modelo](#)
- [Etapa 9: Integre-se aos seus fluxos de trabalho de desenvolvimento](#)
- [Próximas etapas](#)

## Referência de propriedades do recurso

Ao criar seu aplicativo, use essa tabela como referência para configurar as propriedades do Amazon API Gateway e AWS Lambda dos recursos.

Método	Path	Nome da função
GET	/itens	getItems
GET	/itens/ {id}	getItem
PUT	/itens/ {id}	updateItem
POST	/item	addItem
DELETE	/itens/ {id}	deleteItem

## Etapa 1: Crie seu projeto

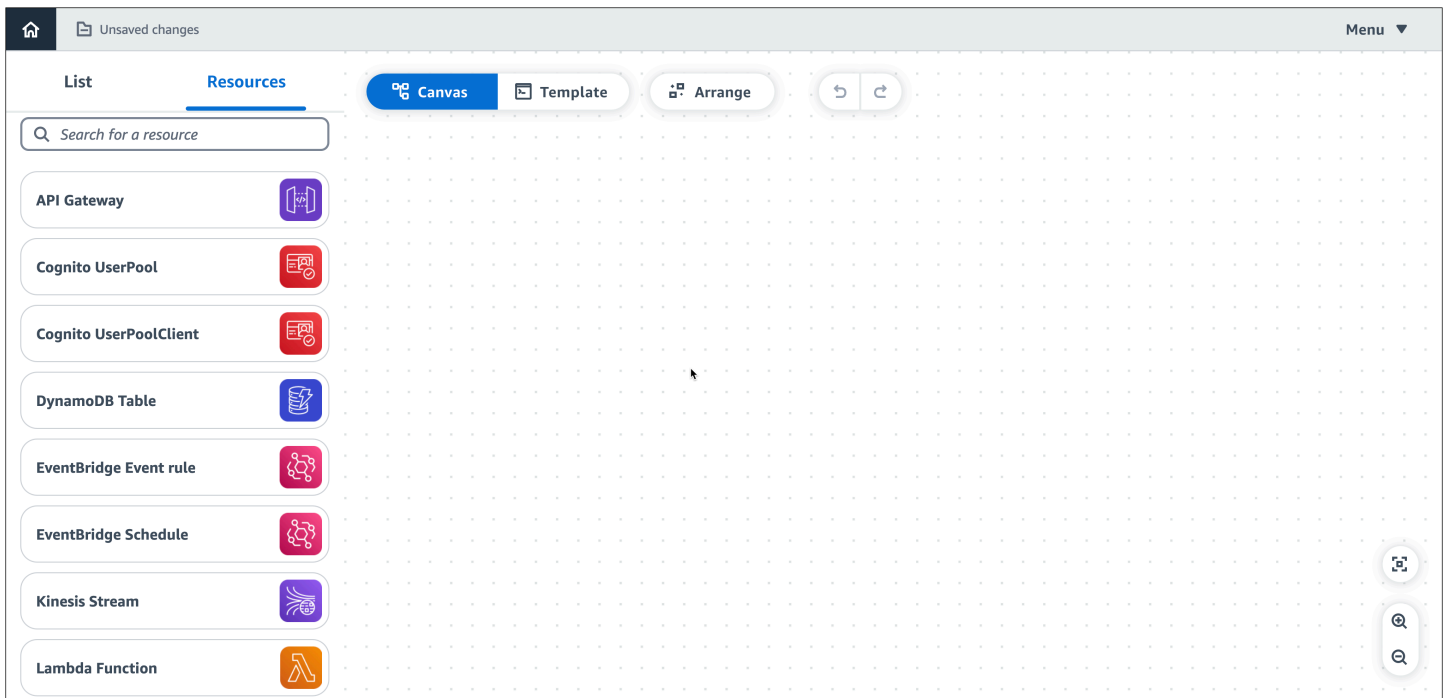
Para começar a usar seu aplicativo CRUD sem servidor, crie um novo projeto no Infrastructure Composer e ative a sincronização local.

Para criar um novo projeto em branco

1. Faça login no [console do Infrastructure Composer](#).
2. Na página inicial, escolha Criar projeto.

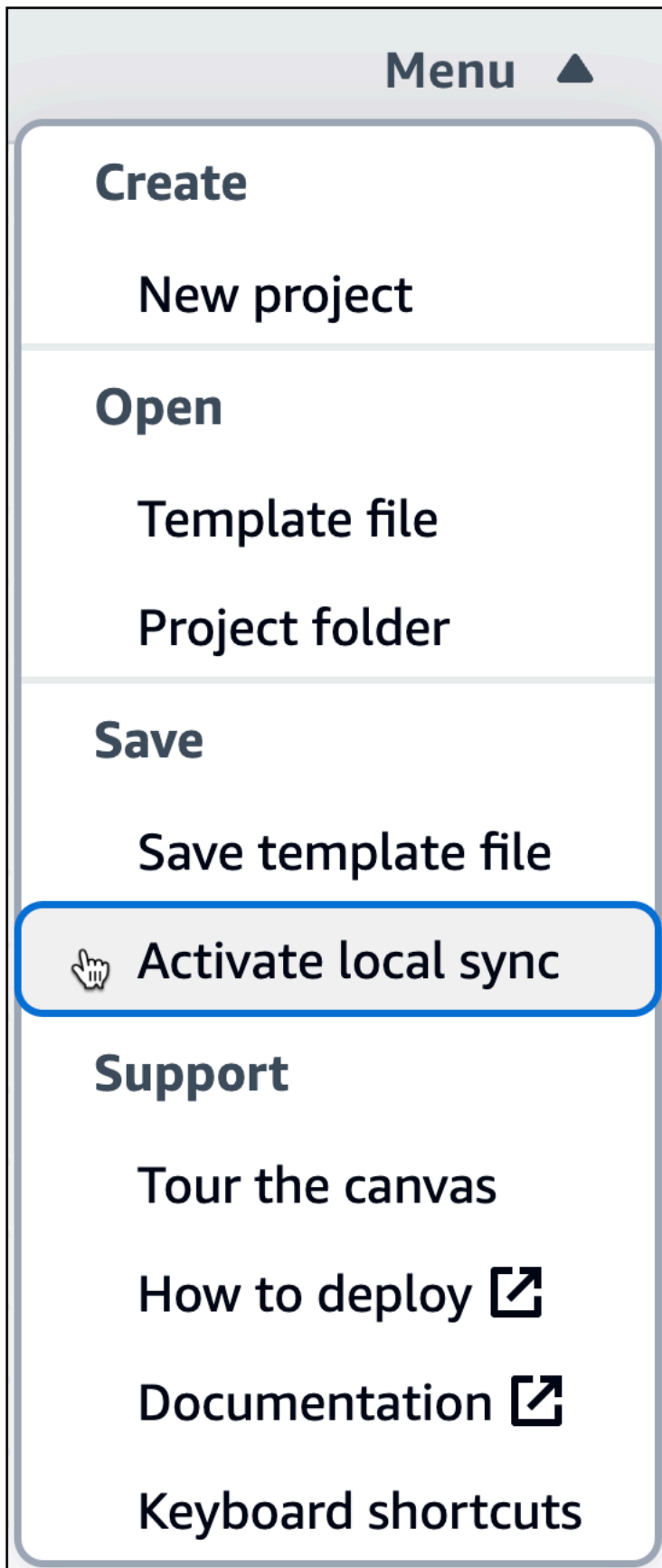
Conforme mostrado na imagem a seguir, o Infrastructure Composer abre a tela visual e carrega um modelo de aplicativo inicial (em branco).






Para ativar a sincronização local

1. No menu Infrastructure Composer, selecione Salvar > Ativar sincronização local.



2. Em Localização do projeto, pressione Selecionar pasta e escolha um diretório. É aqui que o Infrastructure Composer salvará e sincronizará seus arquivos e pastas de modelo conforme você projeta.

O local do projeto não deve conter um modelo de aplicativo existente.

 Note

A sincronização local requer um navegador que ofereça suporte ao acesso ao sistema de arquivosAPI. Para obter mais informações, consulte [O Data Infrastructure Composer obtém acesso a](#).

3. Quando solicitado a permitir o acesso, selecione Exibir arquivos.
4. Pressione Ativar para ativar a sincronização local. Quando solicitado a salvar as alterações, selecione Salvar alterações.

Quando ativado, o indicador de salvamento automático será exibido na área superior esquerda da tela.

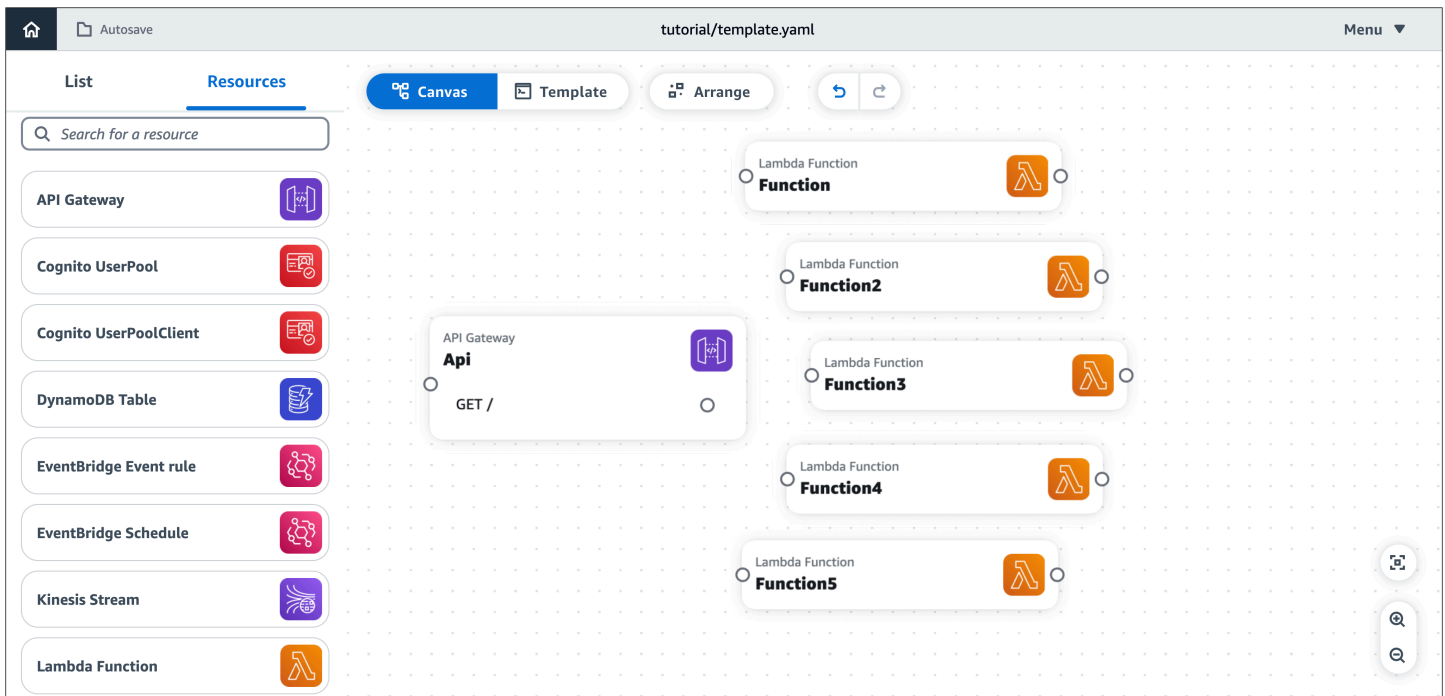
## Etapa 2: adicionar cartões à tela

Comece a projetar sua arquitetura de aplicativo usando placas de componentes aprimoradas, começando com um API Gateway REST API e cinco funções Lambda.

Para adicionar cartões API Gateway e Lambda à tela

Na paleta Recursos, na seção Componentes aprimorados, faça o seguinte:

1. Arraste um cartão APIGateway para a tela.
2. Arraste um cartão da Função Lambda para a tela. Repita até adicionar cinco cartões de Função Lambda à tela.



### Etapa 3: Configurar seu API gateway REST API

Em seguida, adicione cinco rotas em sua placa API Gateway.

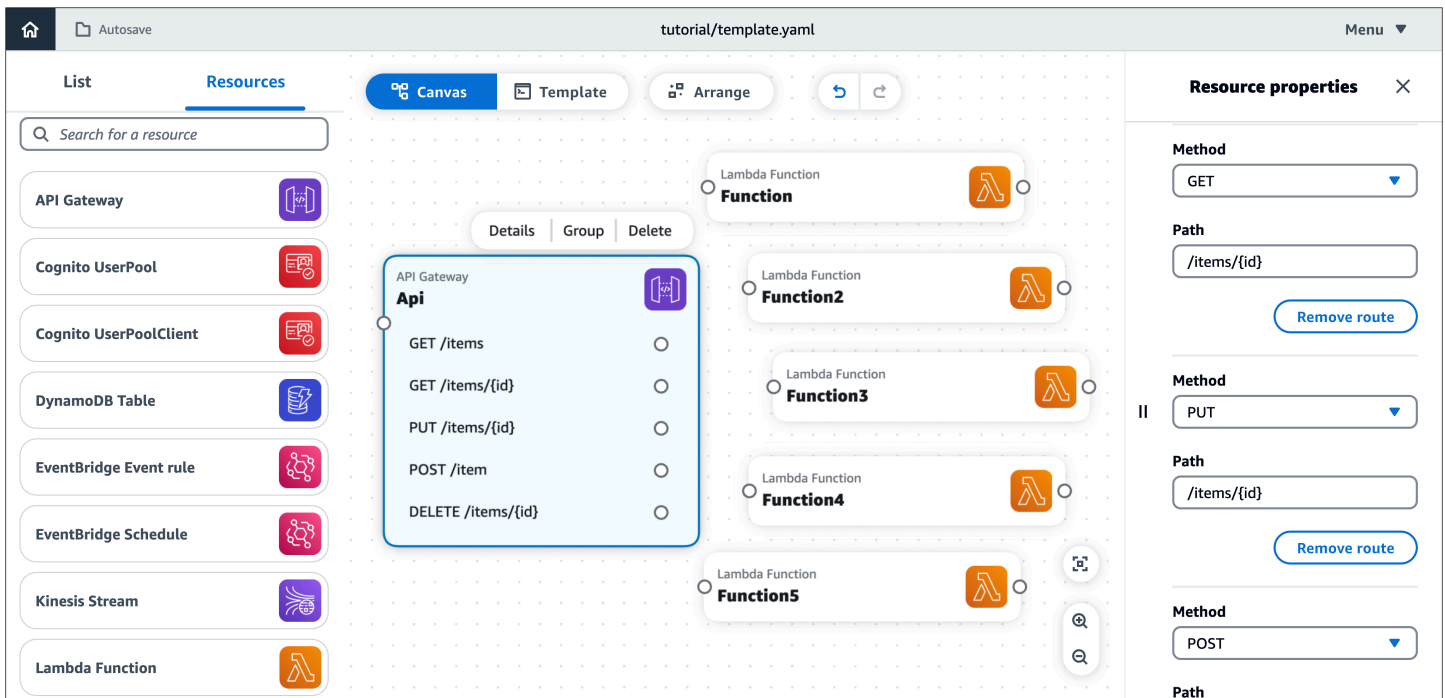
Para adicionar rotas à placa API Gateway

1. Abra o painel de propriedades do recurso para a placa APIGateway. Para abrir o painel, clique duas vezes no cartão. Ou selecione o cartão e, em seguida, escolha Detalhes.
2. No painel Propriedades do recurso, em Rotas, faça o seguinte:

#### Note

Para cada uma das rotas a seguir, use os valores de HTTP método e caminho especificados na [tabela de referência de propriedades do recurso](#).

- a. Em Método, escolha o HTTP método especificado. Por exemplo, GET.
  - b. Em Caminho, insira o caminho especificado. Por exemplo, **/items**.
  - c. Selecione Adicionar rota.
  - d. Repita as etapas anteriores até adicionar todas as cinco rotas especificadas.
3. Escolha Salvar.



## Etapa 4: Configurar suas funções do Lambda

Nomeie cada uma das cinco funções do Lambda conforme especificado na tabela de [referência de propriedades do recurso](#).

Para nomear as funções Lambda

1. Abra o painel de propriedades do recurso de um cartão da Função Lambda. Para abrir o painel, clique duas vezes no cartão. Ou selecione o cartão e, em seguida, escolha Detalhes.
2. No painel Propriedades do recurso, em ID lógica, insira um nome de função especificado. Por exemplo, **getItems**.
3. Escolha Salvar.
4. Repita as etapas anteriores até nomear todas as cinco funções.

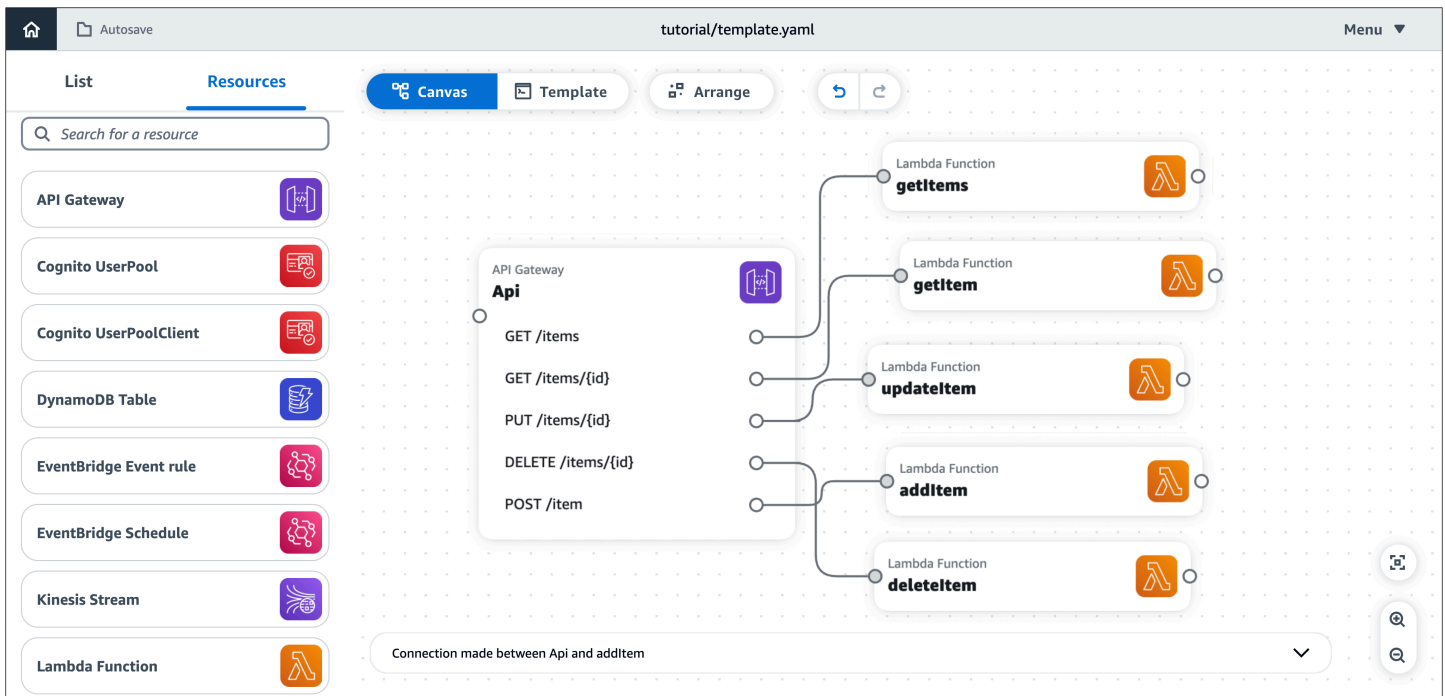
The screenshot displays the AWS Infrastructure Composer interface for a template named 'tutorial/template.yaml'. On the left, a 'Resources' panel lists various AWS services. The central canvas shows an 'API Gateway' resource named 'Api' with five endpoints. Each endpoint is connected to a corresponding 'Lambda Function' resource: 'getItems', 'getItem', 'updateItem', 'addItem', and 'deleteItem'. The 'deleteItem' resource is selected, and its properties are shown in the right-hand pane. The properties pane includes fields for Logical ID (deleteitem), Package type (Zip), and Source path (src/Function5).

## Etapa 5: Conecte seus cartões

Conecte cada rota em sua placa APIGateway à placa de Função Lambda relacionada, conforme especificado na tabela de [referência de propriedades do recurso](#).

Para conectar seus cartões

1. Clique em uma porta direita na placa APIGateway e arraste-a para a porta esquerda da placa Lambda Function especificada. Por exemplo, clique na porta GET/items e arraste-a para a porta esquerda do getItems.
2. Repita a etapa anterior até conectar todas as cinco rotas na placa APIGateway às placas de função Lambda correspondentes.



## Etapa 6: organizar a tela

Organize a tela visual agrupando suas funções do Lambda e organizando todos os cartões.

Para agrupar suas funções

1. Pressione e segure a tecla Shift e selecione cada cartão da Função Lambda na tela.
2. Escolha Grupo.

Para nomear seu grupo

1. Clique duas vezes na parte superior do grupo, perto do nome do grupo (Grupo).

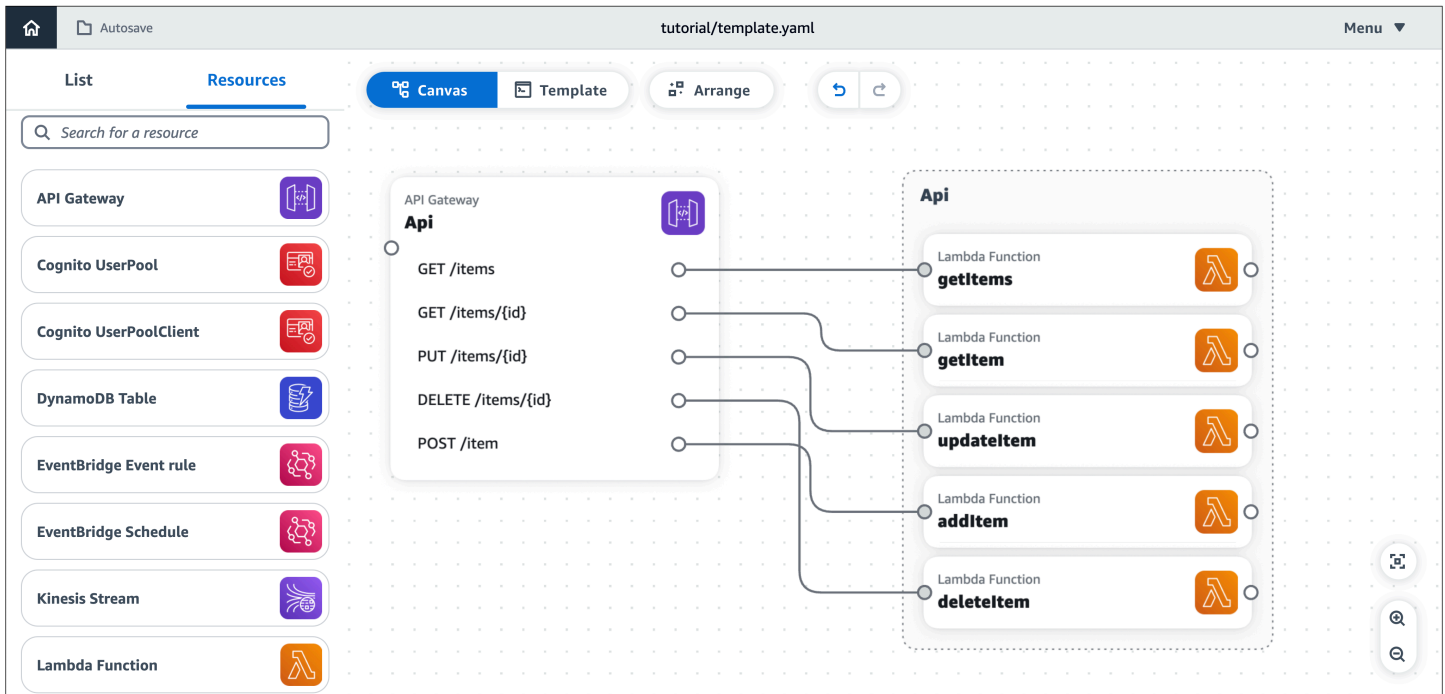
O painel Propriedades do grupo é aberto.

2. No painel Propriedades do grupo, em Nome do grupo, insira **API**.
3. Escolha Salvar.

Para organizar seus cartões

Na tela, acima da área de exibição principal, escolha Organizar.

O Infrastructure Composer organiza e alinha todos os cartões na tela visual, incluindo seu novo grupo (API), conforme mostrado aqui:



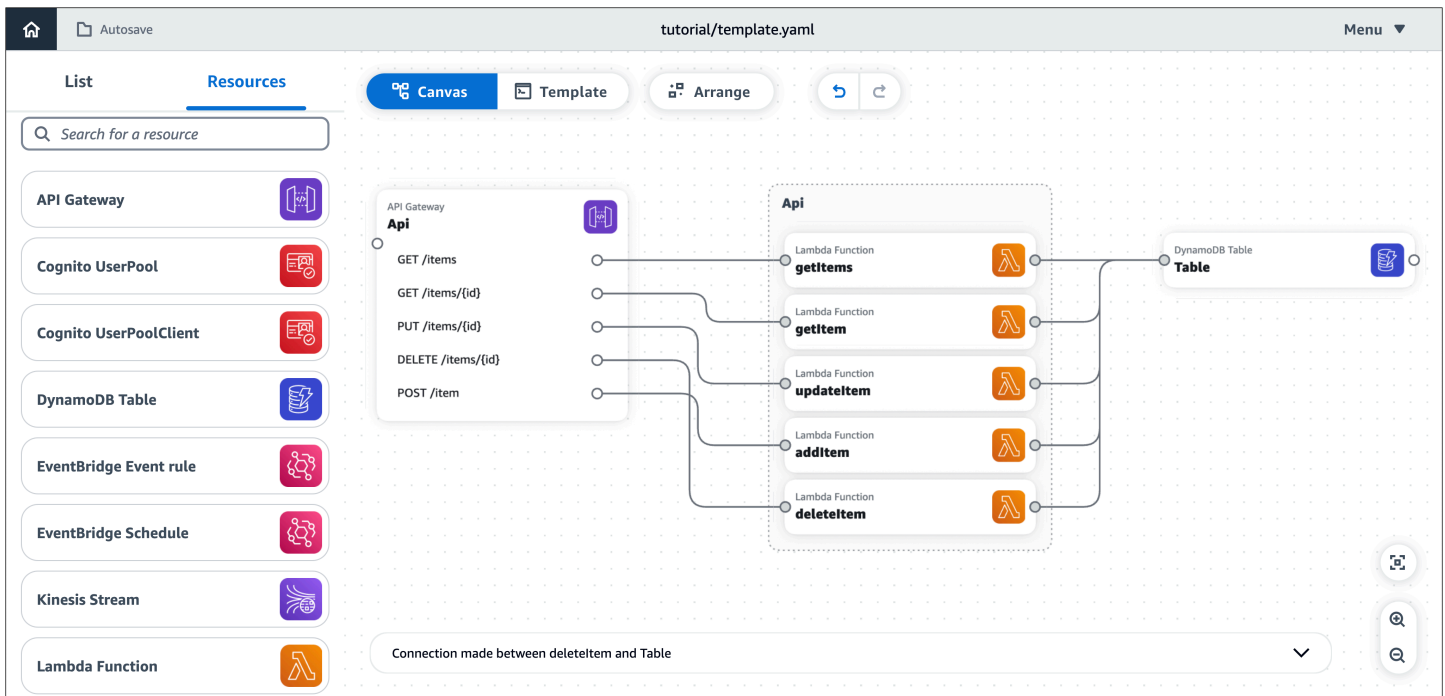
## Etapa 7: Adicionar e conectar uma tabela do DynamoDB

Agora, adicione uma tabela do DynamoDB à arquitetura do seu aplicativo e conecte-a às suas funções do Lambda.

Para adicionar e conectar uma tabela do DynamoDB

1. Na paleta de recursos (Recursos), na seção Componentes aprimorados, arraste um cartão de tabela do DynamoDB para a tela.
2. Clique na porta direita em uma placa de função Lambda e arraste-a até a porta esquerda da placa de tabela do DynamoDB.
3. Repita a etapa anterior até conectar todos os cinco cartões de função Lambda ao cartão de tabela do DynamoDB.
4. (Opcional) Para reorganizar e realinhar os cartões na tela, escolha Organizar.



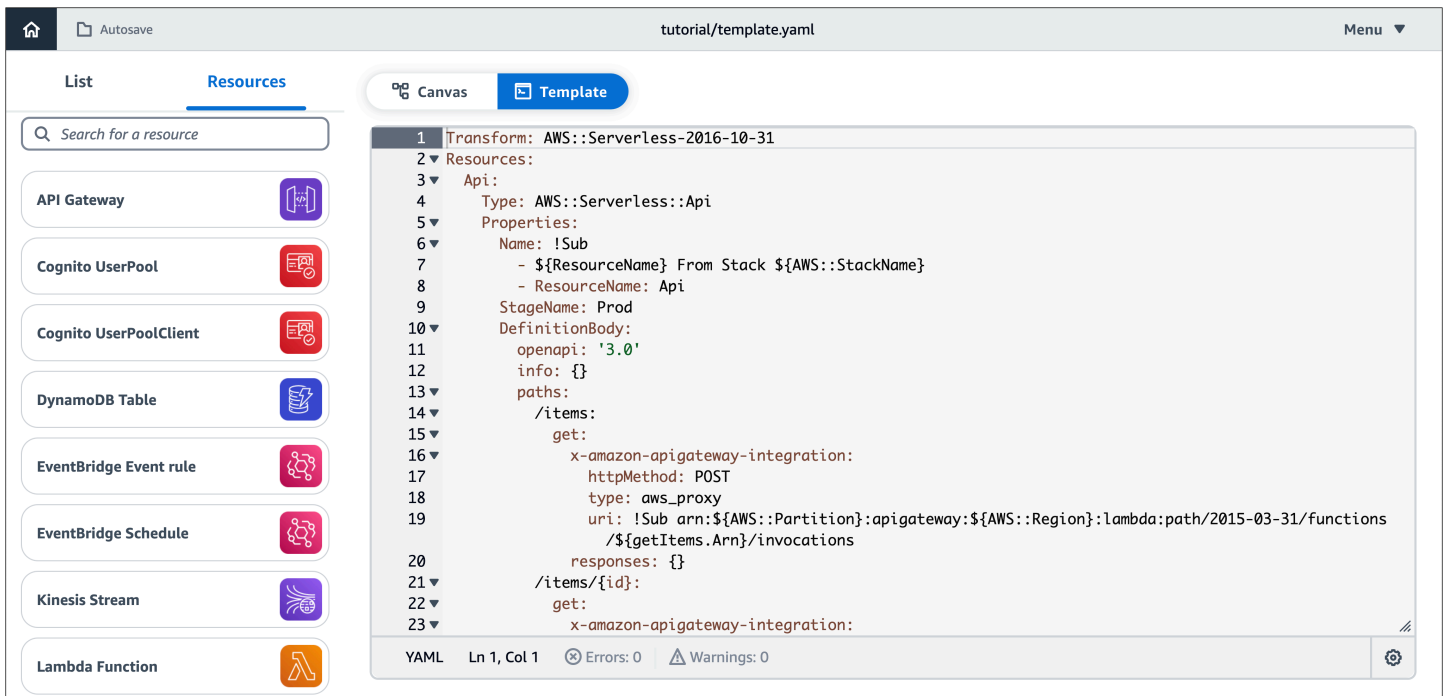


## Etapa 8: revise seu AWS CloudFormation modelo

Parabéns! Você projetou com sucesso um aplicativo sem servidor que está pronto para implantação. Por fim, escolha Modelo para revisar o AWS CloudFormation modelo que o Infrastructure Composer gerou automaticamente para você.

No modelo, o Infrastructure Composer definiu o seguinte:

- A `Transform` declaração, que especifica o modelo como um modelo AWS Serverless Application Model (AWS SAM). Para obter mais informações, consulte [anatomia do AWS SAM modelo](#) no Guia do AWS Serverless Application Model desenvolvedor.
- Um `AWS::Serverless::Api` recurso que especifica seu API Gateway REST API com suas cinco rotas.
- Cinco `AWS::Serverless::Function` recursos, que especificam as configurações de suas funções do Lambda, incluindo suas variáveis de ambiente e políticas de permissões.
- Um `AWS::DynamoDB::Table` recurso que especifica sua tabela do DynamoDB e suas propriedades.
- A `Metadata` seção, que contém informações sobre seu grupo de recursos (API). Para obter mais informações sobre essa seção, consulte [Metadados](#) no Guia do AWS CloudFormation usuário.



## Etapa 9: Integre-se aos seus fluxos de trabalho de desenvolvimento

Use o arquivo de modelo e os diretórios do projeto que o Infrastructure Composer criou para testes e implantação adicionais.

- Com a sincronização local, você pode conectar o Infrastructure Composer IDE à sua máquina local para acelerar o desenvolvimento. Para saber mais, consulte [Conecte o console do Infrastructure Composer ao seu console local IDE](#).
- Com a sincronização local, você pode usar a interface de linha de AWS Serverless Application Model comando (AWS SAM CLI) em sua máquina local para testar e implantar seu aplicativo. Para saber mais, consulte [Implemente seu aplicativo sem servidor Infrastructure Composer na nuvem AWS](#).

## Próximas etapas

Agora você está pronto para criar seus próprios aplicativos com o Infrastructure Composer. Para obter detalhes detalhados sobre o uso do Infrastructure Composer, consulte [Como compor em AWS Infrastructure Composer](#). Quando você estiver pronto para implantar seu aplicativo, consulte [Implemente seu aplicativo sem servidor Infrastructure Composer na nuvem AWS](#).

# Onde você pode usar o Infrastructure Composer

Você pode usar o Infrastructure Composer a partir de seu console AWS Toolkit for Visual Studio Code, de e no Infrastructure Composer no modo CloudFormation console. Embora cada uma varie para casos de uso ligeiramente diferentes, no geral são experiências semelhantes. Esta seção fornece detalhes de cada experiência.

O tópico [Usando o AWS Infrastructure Composer console](#) é uma visão geral abrangente da experiência padrão do console. O tópico [CloudFormation modo de console](#) fornece detalhes sobre uma versão do Infrastructure Composer que está integrada ao fluxo de trabalho da AWS CloudFormation pilha. [AWS Toolkit for Visual Studio Code](#) fornece informações sobre como acessar e usar o Infrastructure Composer no VS Code.

## Tópicos

- [Usando o AWS Infrastructure Composer console](#)
- [Usando o Infrastructure Composer no modo CloudFormation console](#)
- [Usando o Infrastructure Composer do AWS Toolkit for Visual Studio Code](#)

## Usando o AWS Infrastructure Composer console

Esta seção fornece detalhes sobre como acessar e usar a AWS Infrastructure Composer partir do console do Infrastructure Composer. Essa é a experiência padrão do Infrastructure Composer e é uma boa maneira de se familiarizar com o Infrastructure Composer. Você também pode integrar o console do Infrastructure Composer ao seu console local IDE. Para obter detalhes, consulte [Conecte o console do Infrastructure Composer ao seu console local IDE](#).

Você também pode [acessar o Infrastructure Composer a partir do AWS Toolkit no VS Code](#) e usar um [modo do Infrastructure Composer projetado especificamente para ser](#) usado em. AWS CloudFormation

Para obter a documentação geral sobre o uso do Infrastructure Composer, consulte [Como compor](#).

## Tópicos

- [AWS Infrastructure Composer visão geral visual do console](#)
- [Gerencie seu projeto a partir do console do Infrastructure Composer](#)

- [Conecte o console do Infrastructure Composer ao seu console local IDE](#)
- [Permitir acesso à página da web a arquivos locais no Infrastructure Composer](#)
- [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#)
- [Importe funções para o Infrastructure Composer a partir do console Lambda](#)
- [Exportar uma imagem da tela visual do Infrastructure Composer](#)

## AWS Infrastructure Composer visão geral visual do console

Esta seção fornece uma visão geral visual do AWS Infrastructure Composer console.

### Tópicos

- [Home page \(Página inicial\)](#)
- [Designer visual e tela visual](#)

### Home page (Página inicial)

A imagem a seguir é da página inicial no console do Infrastructure Composer:

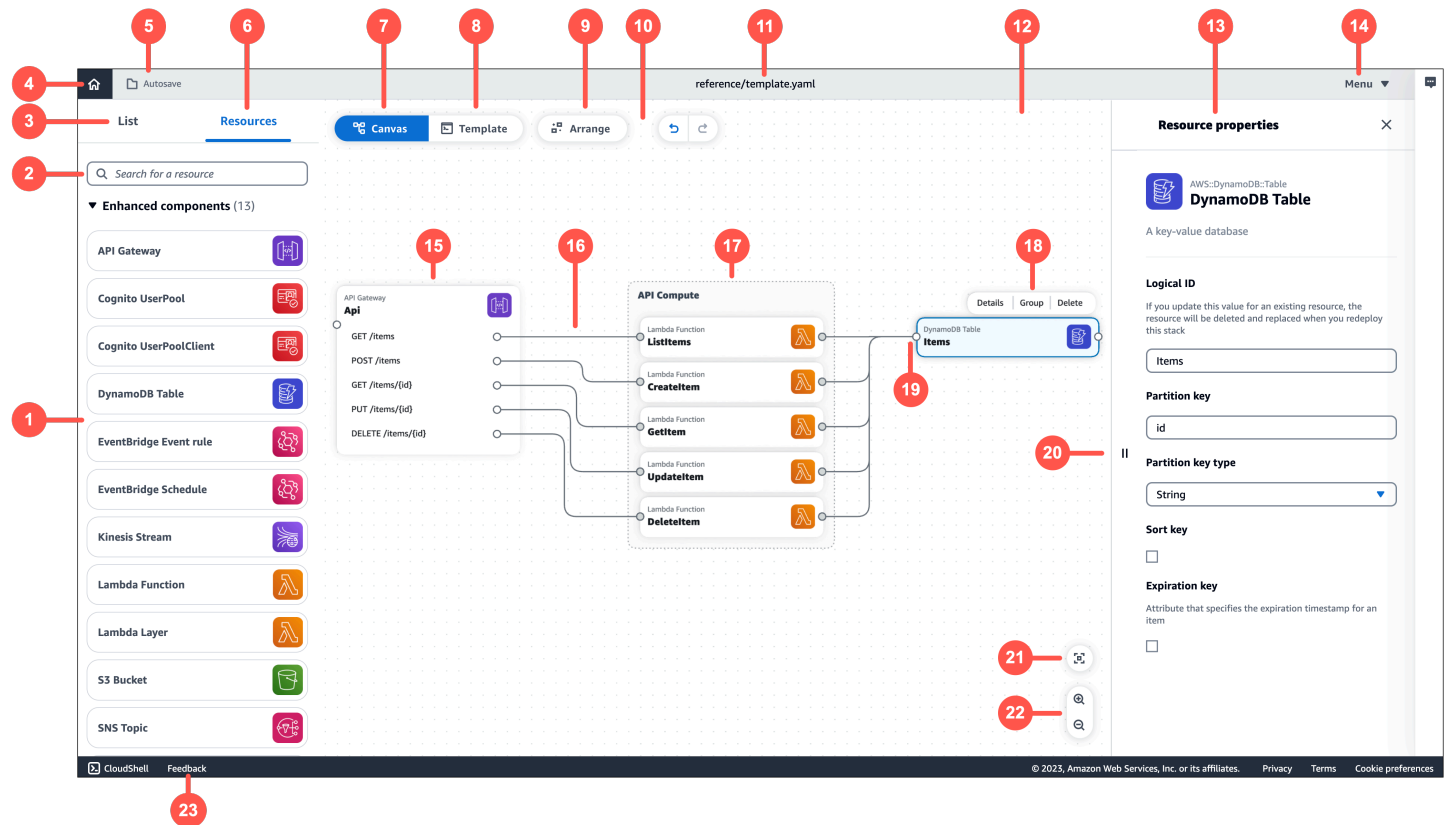
The screenshot displays the AWS Infrastructure Composer console. On the left, a sidebar contains the 'Infrastructure Composer' logo and navigation options: 'Canvas', 'Documentation', 'AWS Toolkit for VS Code', and 'Import a Lambda Function'. The main content area features a breadcrumb 'Infrastructure Composer > Home' and a 'New!' notification stating that the service is now part of the AWS Toolkit for Visual Studio Code. Below the notification are three buttons: 'Download VS Code Extension', 'Open demo', and 'Create project'. The 'About' section explains that Infrastructure Composer helps visually compose modern applications and iterate on their architecture design. It lists 'How it works' with four bullet points: dragging and dropping CloudFormation resources, connecting enhanced components for automatic IaC build, transitioning between visual workflows and Step Functions, and integrating with local sync or VS Code. To the right, a visual diagram shows an 'API Gateway' connected to a 'Lambda' function, which triggers an 'Amazon S3' bucket. The Lambda function also triggers an 'Amazon DynamoDB' database. The database is connected to an 'Amazon ElastiCache' instance. The ElastiCache instance triggers an 'Amazon SES' service, which is connected to an 'Amazon SNS' topic. The SNS topic triggers an 'Amazon SQS' queue, which is connected to a 'Send Message' action.

1. Documentação — Acesse a documentação do Infrastructure Composer.
2. Tela — Vá até a tela e crie ou carregue um projeto.
3. Demonstração — Abra o aplicativo de demonstração do Infrastructure Composer.

4. Criar projeto — Crie ou carregue um projeto.
5. Comece a criar — Links rápidos para começar a criar um aplicativo.
6. Feedback — Acesse aqui para enviar feedback.

## Designer visual e tela visual

A imagem a seguir é do designer visual e da tela visual do Infrastructure Composer:



1. Paleta de recursos — Exibe cartões com os quais você pode criar designs.
2. Barra de pesquisa de recursos — Pesquise cartões que você pode adicionar à tela.
3. Lista — Exibe uma visualização em árvore dos recursos do seu aplicativo.
4. Início — Selecione aqui para acessar a página inicial do Infrastructure Composer.
5. Status de salvamento — Indica se as alterações do Infrastructure Composer foram salvas em sua máquina local. Os estados são:
  - Salvamento automático — A sincronização local é ativada e seu projeto está sendo sincronizado e salvo automaticamente.
  - Alterações salvas — Seu modelo de aplicativo é salvo em sua máquina local.

- Alterações não salvas — Seu modelo de aplicativo tem alterações que não são salvas em sua máquina local.
6. Recursos — Exibe a paleta de recursos.
  7. Tela — Exibe a visualização em tela do seu aplicativo na área de exibição principal.
  8. Modelo — Exibe a visualização do modelo do seu aplicativo na área de exibição principal.
  9. Organizar — Organiza a arquitetura do seu aplicativo na tela.
  10. Desfazer e refazer — Execute ações de desfazer e refazer quando houver suporte.
  11. Nome do modelo — Indica o nome do modelo que você está criando.
  12. Área de visualização principal — Exibe a tela ou o modelo com base na sua seleção.
  13. Painel de propriedades do recurso — Exibe as propriedades relevantes do cartão que foi selecionado na tela. Esse painel é dinâmico. As propriedades exibidas mudarão conforme você configura seu cartão.
  14. Menu — Fornece opções gerais, como as seguintes:
    - Criar um projeto
    - Abra um arquivo de modelo ou projeto
    - Salvar um arquivo de modelo
    - [Ativar sincronização local](#)
    - [Exportar tela](#)
    - Obtenha suporte
    - Atalhos de teclado
  15. Cartão — Exibe uma visualização do seu cartão na tela.
  16. Linha — Representa uma conexão entre cartões.
  17. Grupo — Agrupa cartões selecionados para organização visual.
  18. Ações do cartão — Fornece ações que você pode realizar no seu cartão.
    - a. Detalhes — Abre o painel de propriedades do recurso.
    - b. Grupo — Agrupe as cartas selecionadas.
    - c. Excluir — Exclui o cartão da sua tela.
  19. Porta — Pontos de conexão com outras placas.
  20. Campos de propriedades de recursos — Um conjunto selecionado de campos de propriedades para configurar para seus cartões.
  21. Centralizar novamente — centralize novamente o diagrama do aplicativo na tela visual.

22Zoom — Aumente e diminua o zoom em sua tela.

23Feedback — Acesse aqui para enviar feedback.

## Gerencie seu projeto a partir do console do Infrastructure Composer

Este tópico fornece orientação sobre as tarefas básicas que você executa para gerenciar seu projeto a partir do console do Infrastructure Composer. Isso inclui tarefas comuns, como criar um novo projeto, salvar um projeto e importar um projeto ou modelo. Você também pode carregar um projeto existente se ativar o [modo de sincronização local](#). Depois de ativar o modo de sincronização local, você pode fazer o seguinte:

- Crie um novo projeto que consiste em um modelo inicial e uma estrutura de pastas.
- Carregue um projeto existente escolhendo uma pasta principal que contenha o modelo e os arquivos do seu projeto.
- Use o Infrastructure Composer para gerenciar seus modelos e pastas

Com o modo de sincronização local, o Infrastructure Composer salva automaticamente as alterações de modelo e pasta do seu projeto em sua máquina local. Se o seu navegador não suportar o modo de sincronização local ou se você preferir usar o Infrastructure Composer sem o modo de sincronização local ativado, você pode criar um novo modelo ou carregar um modelo existente. Para salvar as alterações, você deve exportar o modelo para sua máquina local.

### Note

O Infrastructure Composer oferece suporte a aplicativos que consistem no seguinte:

- Um AWS Serverless Application Model modelo AWS CloudFormation or que define seu código de infraestrutura.
- Uma estrutura de pastas que organiza seus arquivos de projeto, como o código da função Lambda, arquivos de configuração e pastas de criação.

### Tópicos

- [Crie um novo projeto no console do Infrastructure Composer](#)
- [Importar uma pasta de projeto existente no console do Infrastructure Composer](#)
- [Importar um modelo de projeto existente no console do Infrastructure Composer](#)

- [Salve um modelo de projeto existente no console do Infrastructure Composer](#)

## Crie um novo projeto no console do Infrastructure Composer

Quando você cria um novo projeto, o Infrastructure Composer gera um modelo inicial. Conforme você projeta seu aplicativo na tela, seu modelo é modificado. Para salvar seu trabalho, você deve exportar seu modelo ou ativar o modo de sincronização local.

### Como criar um projeto

1. Faça login no [console do Infrastructure Composer](#).
2. Na página inicial, escolha Criar projeto.

#### Note

Você também pode carregar um existente no Infrastructure Composer, mas primeiro você deve [ativar o modo de sincronização local](#). Depois de ativado, veja [Carregue um projeto existente do Infrastructure Composer com a sincronização local ativada](#) para carregar um projeto existente.

## Importar uma pasta de projeto existente no console do Infrastructure Composer

Usando o modo de sincronização local, você pode importar a pasta principal de um projeto existente. Se o seu projeto contiver vários modelos, você poderá escolher o modelo a ser carregado.

### Para importar um projeto existente da página inicial

1. Faça login no [console do Infrastructure Composer](#).
2. Na página inicial, escolha Carregar um CloudFormation modelo.
3. Em Local do projeto, escolha Seleccionar pasta. Selecione a pasta principal do seu projeto e escolha Seleccionar.

#### Note

Se você não receber esse aviso, seu navegador pode não suportar o acesso ao sistema de arquivosAPI, que é necessário para o modo de sincronização local. Para obter




mais informações, consulte [Permitir acesso à página da web a arquivos locais no Infrastructure Composer](#).

4. Quando solicitado pelo seu navegador, selecione Exibir arquivos.
5. Para Arquivo de modelo, escolha seu modelo na lista suspensa. Se o seu projeto contiver um único modelo, o Infrastructure Composer o selecionará automaticamente para você.
6. Escolha Criar.

Para importar um projeto existente da tela

1. Na tela, escolha Menu para abrir o menu.
2. Na seção Abrir, escolha Pasta do projeto.

 Note

Se a opção de pasta do projeto não estiver disponível, seu navegador pode não suportar o acesso ao sistema de arquivosAPI, que é necessário para o modo de sincronização local. Para obter mais informações, consulte [Permitir acesso à página da web a arquivos locais no Infrastructure Composer](#).

3. Em Local do projeto, escolha Selecionar pasta. Selecione a pasta principal do seu projeto e escolha Selecionar.
4. Quando solicitado pelo seu navegador, selecione Exibir arquivos.
5. Para Arquivo de modelo, escolha seu modelo na lista suspensa. Se o seu projeto contiver um único modelo, o Infrastructure Composer o selecionará automaticamente para você.
6. Escolha Criar.

Quando você importa uma pasta de projeto existente, o Infrastructure Composer ativa o modo de sincronização local. As alterações feitas no modelo ou nos arquivos do seu projeto são salvas automaticamente na sua máquina local.

### Importar um modelo de projeto existente no console do Infrastructure Composer

Quando você importa um modelo existente AWS CloudFormation ou um AWS SAM modelo, o Infrastructure Composer gera automaticamente uma visualização da arquitetura do seu aplicativo na tela.

Você pode importar um modelo de projeto da sua máquina local.

Para importar um modelo de projeto existente

1. Faça login no [console do Infrastructure Composer](#).
2. Escolha Criar projeto para abrir uma tela em branco.
3. Escolha Menu para abrir o menu.
4. Na seção Abrir, escolha Arquivo de modelo.
5. Selecione seu modelo e escolha Abrir.

Para salvar as alterações em seu modelo, você deve exportar seu modelo ou ativar o modo de sincronização local.

### Salve um modelo de projeto existente no console do Infrastructure Composer

Se você não usa o modo de sincronização local, você deve exportar seu modelo para salvar suas alterações. Se você tiver o modo de sincronização local ativado, não será necessário salvar manualmente seu modelo. As alterações são salvas automaticamente em sua máquina local.

Para salvar um modelo de projeto existente

1. Na tela do Infrastructure Composer, escolha Menu para abrir o menu.
2. Na seção Salvar, escolha Salvar arquivo de modelo.
3. Forneça um nome para seu modelo.
4. Selecione um local para salvar seu modelo.
5. Escolha Salvar.

### Conecte o console do Infrastructure Composer ao seu console local IDE

Para conectar o console do Infrastructure Composer ao seu ambiente de desenvolvimento integrado local (IDE), use o modo de sincronização local. Esse modo sincroniza e salva automaticamente os dados em sua máquina local. Para obter mais informações sobre o modo de sincronização local, consulte [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#). Para obter instruções sobre como usar o modo de sincronização local, consulte [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#).

## Note

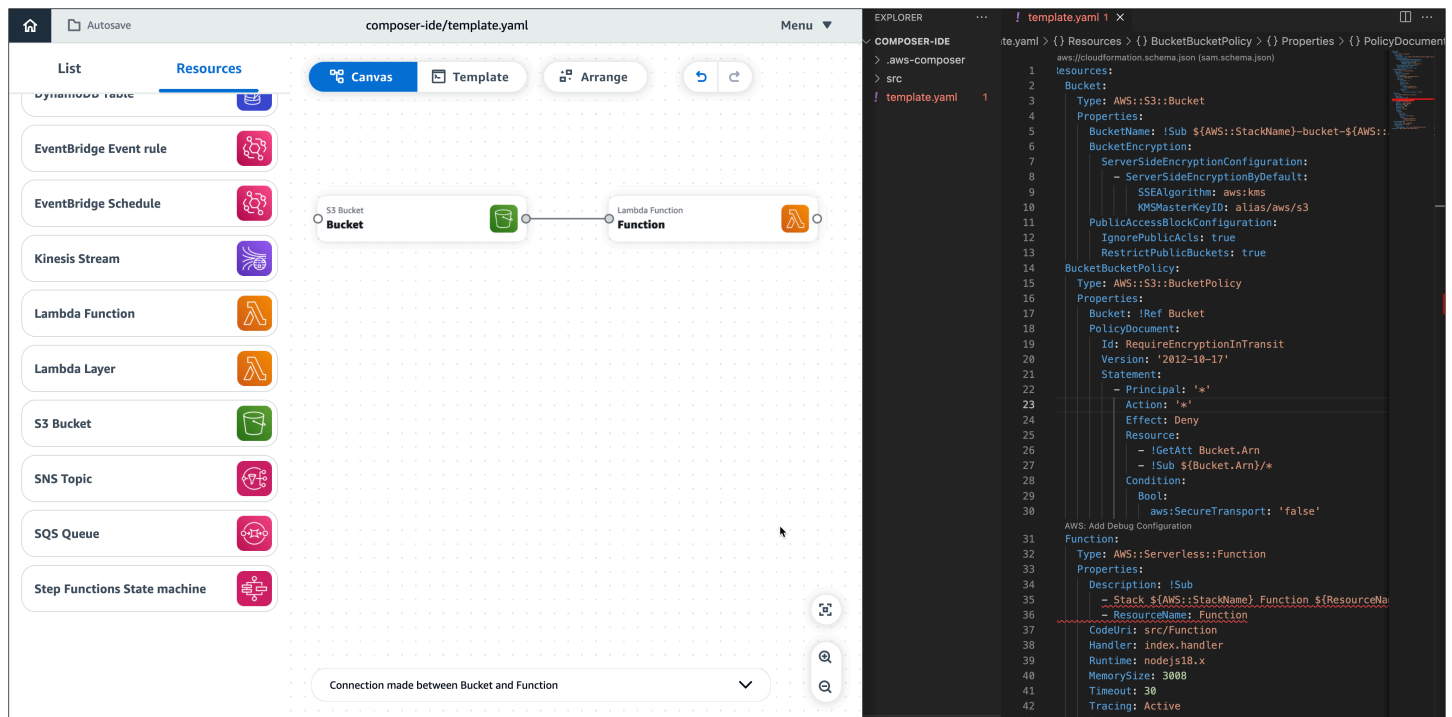
A opção Ativar sincronização local não está disponível em todos os navegadores. Ele está disponível no Google Chrome e no Microsoft Edge.

## Benefícios de usar o Infrastructure Composer com seu local IDE

Conforme você projeta no Infrastructure Composer, seu modelo local e o diretório do projeto são automaticamente sincronizados e salvos.

Você pode usar seu local IDE para visualizar as alterações e modificar seus modelos. As alterações feitas localmente são sincronizadas automaticamente com o Infrastructure Composer.

Você pode usar ferramentas locais, como a AWS Serverless Application Model Command Line Interface (AWS SAM CLI), para criar, testar, implantar seu aplicativo e muito mais. O exemplo a seguir mostra como você pode arrastar e soltar recursos na tela visual do Infrastructure Composer que, por sua vez, cria marcações em seu AWS SAM modelo em seu local. IDE



The screenshot displays the Infrastructure Composer interface. On the left, a 'Resources' panel lists various AWS services like DynamoDB Table, EventBridge Event rule, S3 Bucket, and Lambda Function. The central 'Canvas' shows a visual diagram with an 'S3 Bucket' resource connected to a 'Lambda Function' resource. On the right, the 'EXPLORER' view shows the corresponding AWS SAM template snippet for the 'BucketBucketPolicy' resource, including properties for encryption and public access control.

```
1 resources:
2   Bucket:
3     Type: AWS::S3::Bucket
4     Properties:
5       BucketName: !Sub ${AWS::StackName}-bucket-${AWS::
6         BucketEncryption:
7           ServerSideEncryptionConfiguration:
8             - ServerSideEncryptionByDefault:
9                 SSEAlgorithm: aws:kms
10                KMSMasterKeyId: alias/aws/s3
11            PublicAccessBlockConfiguration:
12              IgnorePublicAcls: true
13              RestrictPublicBuckets: true
14      BucketBucketPolicy:
15        Type: AWS::S3::BucketPolicy
16        Properties:
17          Bucket: !Ref Bucket
18        PolicyDocument:
19          Id: RequireEncryptionInTransit
20          Version: '2012-10-17'
21          Statement:
22            - Principal: '*'
23              Action: '*'
24              Effect: Deny
25              Resource:
26                - !GetAtt Bucket.Arn
27                - !Sub ${Bucket.Arn}/*
28              Condition:
29                Bool:
30                  aws:SecureTransport: 'false'
31      AWS: Add Debug Configuration
32      Function:
33        Type: AWS::Serverless::Function
34        Properties:
35          Description: !Sub
36            - Stack ${AWS::StackName} Function ${ResourceName}
37            - ResourceName: Function
38          CodeUri: src/Function
39          Handler: index.handler
40          Runtime: nodejs18.x
41          MemorySize: 3008
42          Timeout: 30
43          Tracing: Active
44          Events:
```

## Integre o Infrastructure Composer com seu local IDE

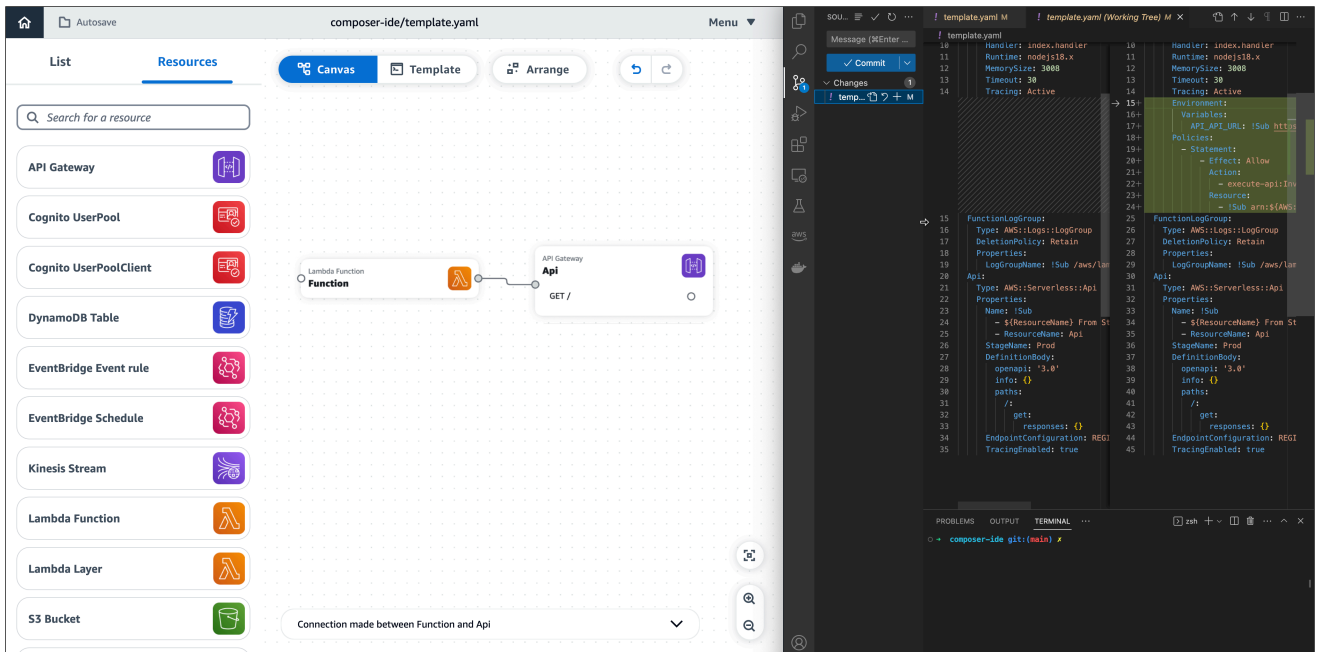
Para integrar o Infrastructure Composer ao seu local IDE

1. No Infrastructure Composer, crie ou carregue um projeto e ative a sincronização local selecionando o botão Menu no canto superior direito da tela e escolhendo Ativar sincronização local.

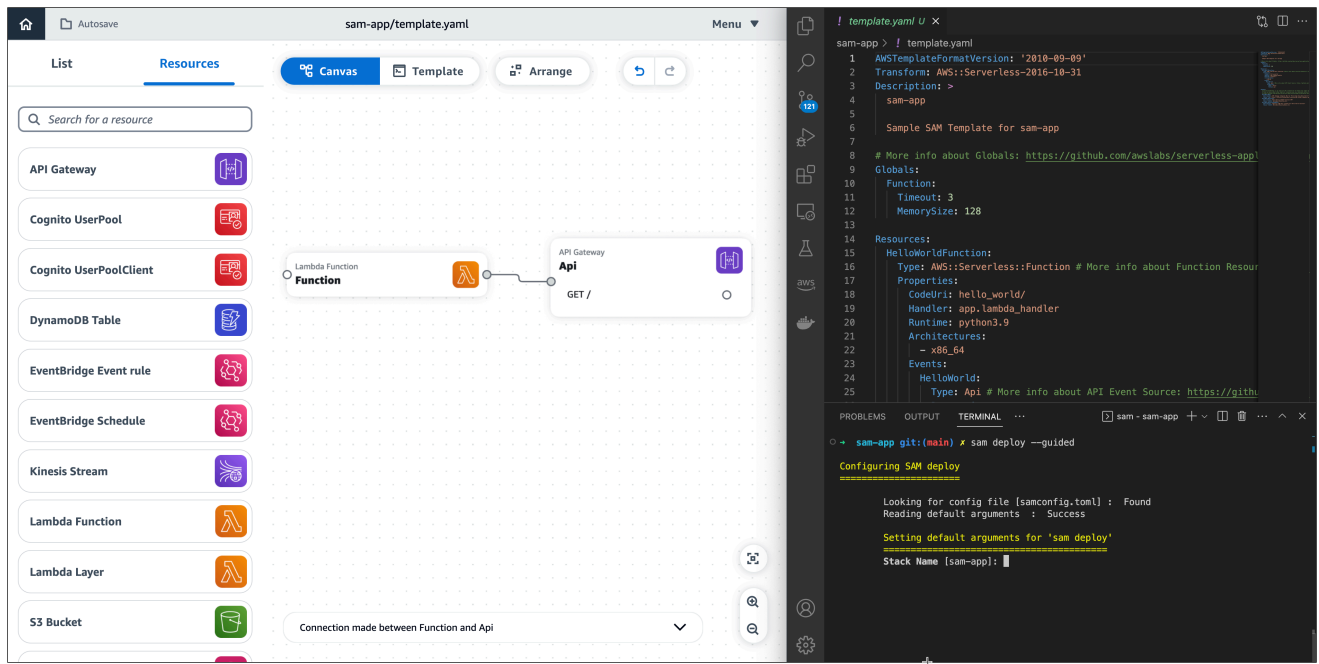
### Note

A opção Ativar sincronização local não está disponível em todos os navegadores. Ele está disponível no Google Chrome e no Microsoft Edge.

2. Em seu local IDE, abra a mesma pasta de projeto do Infrastructure Composer.
3. Use o Infrastructure Composer com seu local IDE. As atualizações feitas no Infrastructure Composer serão sincronizadas automaticamente com sua máquina local. Aqui estão alguns exemplos do que você pode fazer:
  - a. Use o sistema de controle de versão de sua escolha para rastrear as atualizações que estão sendo realizadas pelo Infrastructure Composer.



- b. Use o AWS SAM CLI localmente para criar, testar, implantar seu aplicativo e muito mais. Para saber mais, consulte [Implemente seu aplicativo sem servidor Infrastructure Composer na nuvem AWS](#).



## Permitir acesso à página da web a arquivos locais no Infrastructure Composer

O console do Infrastructure Composer oferece suporte ao [modo de sincronização local](#) e [às funções de importação do console Lambda](#). Para usar esses recursos, é necessário um navegador da Web que ofereça suporte ao acesso ao API sistema de arquivos. Qualquer versão recente do Google Chrome e do Microsoft Edge oferece suporte a todos os recursos do File System Access API e pode ser usada com o modo de sincronização local no Infrastructure Composer.

O Acesso ao Sistema de Arquivos API permite que páginas da Web tenham acesso ao seu sistema de arquivos local para ler, gravar ou salvar arquivos. Esse recurso está desativado por padrão e requer sua permissão por meio de um aviso visual para permitir isso. Uma vez concedido, esse acesso permanece durante a sessão do navegador da sua página da web.

Para saber mais sobre o acesso ao sistema de arquivosAPI, consulte:

- [Acesso ao sistema de arquivos API](#) nos documentos web do mdn.
- [O acesso ao sistema de arquivosAPI: simplificando o acesso aos arquivos locais](#) no site web.dev.

## modo de sincronização local

O modo de sincronização local permite que você sincronize e salve automaticamente seus arquivos de modelo e pastas de projetos localmente à medida que você cria no Infrastructure Composer. Para usar esse recurso, é necessário um navegador da Web que ofereça suporte ao acesso ao API sistema de arquivos.

### O Data Infrastructure Composer obtém acesso a

O Infrastructure Composer obtém acesso de leitura e gravação à pasta do projeto que você permite, junto com todas as pastas secundárias dessa pasta do projeto. Esse acesso é usado para criar, atualizar e salvar quaisquer arquivos de modelo, pastas de projeto e diretórios de backup que são gerados à medida que você projeta. Os dados acessados pelo Infrastructure Composer não são usados para nenhuma outra finalidade e não são armazenados em nenhum lugar além do sistema de arquivos local.

#### Acesso a dados confidenciais

O acesso ao sistema de arquivos API exclui ou limita o acesso a diretórios específicos que podem conter dados confidenciais. Ocorrerá um erro se você selecionar um desses diretórios para usar com o modo de sincronização local do Infrastructure Composer. Você pode escolher outro diretório local para se conectar ou usar o Infrastructure Composer em seu modo padrão com a sincronização local desativada.

Para obter mais informações, incluindo exemplos de diretórios confidenciais, consulte [Usuários dando acesso a mais ou mais arquivos confidenciais do que pretendiam](#) no Relatório preliminar do Grupo Comunitário do W3C de Acesso ao Sistema de Arquivos.

Se você usa Windows Subsystem for Linux (WSL), o acesso ao sistema de arquivos API exclui o acesso a todo Linux diretório devido à sua localização dentro do seu Windows sistema. Você pode usar o Infrastructure Composer com a sincronização local desativada ou configurar uma solução para sincronizar arquivos de projeto a partir do seu WSL diretório para um diretório de trabalho em Windows. Em seguida, use o modo de sincronização local do Infrastructure Composer com seu Windows diretório.

## Sincronize e salve localmente seu projeto no console do Infrastructure Composer

Esta seção fornece informações sobre como usar o modo de sincronização local do Infrastructure Composer para sincronizar e salvar automaticamente seu projeto em sua máquina local.

Recomendamos que você use a sincronização local pelos seguintes motivos:

Você pode ativar a sincronização local para um novo projeto ou carregar um projeto existente com a sincronização local ativada.

- Por padrão, você precisa salvar manualmente seu modelo de aplicativo à medida que projeta. Use a sincronização local para salvar automaticamente seu modelo de aplicativo em sua máquina local à medida que você faz alterações.
- A sincronização local gerencia e sincroniza automaticamente as pastas do projeto, a pasta de backup e [os arquivos externos compatíveis](#) com sua máquina local.
- Ao usar a sincronização local, você pode conectar o Infrastructure Composer ao seu local IDE para acelerar o desenvolvimento. Para saber mais, consulte [Conecte o console do Infrastructure Composer ao seu console local IDE](#).

## O que o modo de sincronização local salva

O modo de sincronização local sincroniza automaticamente e salva o seguinte em sua máquina local:

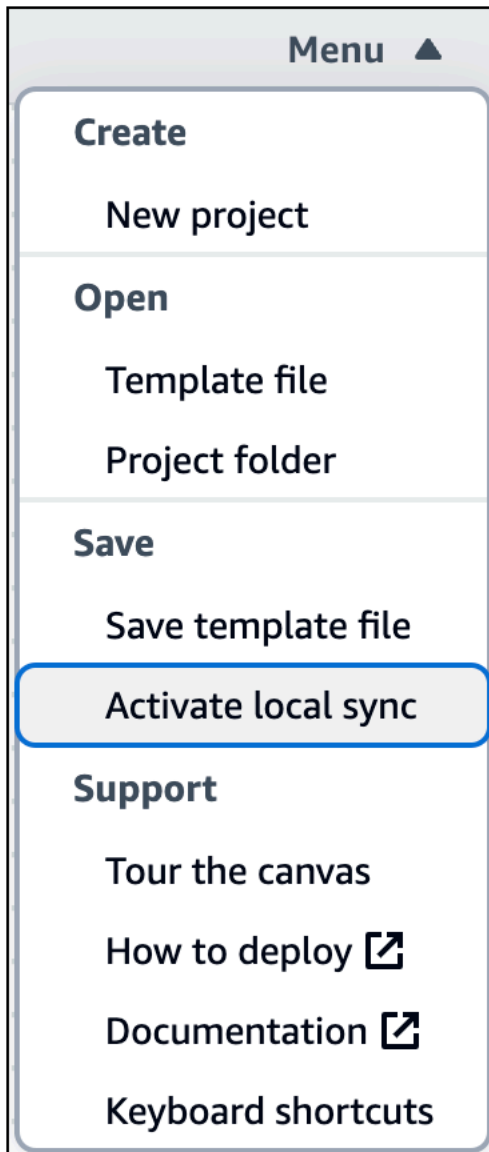
- Arquivo de modelo de aplicativo — O modelo AWS CloudFormation or AWS Serverless Application Model (AWS SAM) que contém sua infraestrutura como código (IaC).
- Pastas do projeto — Uma estrutura geral de diretórios que organiza suas AWS Lambda funções.
- Diretório de backup — Um diretório de backup chamado `.aws-composer`, criado na raiz da localização do seu projeto. Esse diretório contém uma cópia de backup do arquivo de modelo do aplicativo e das pastas do projeto.
- Arquivos externos — Arquivos externos compatíveis que você pode usar no Infrastructure Composer. Para saber mais, consulte [Faça referência a arquivos externos no Infrastructure Composer](#).

## Requisitos do navegador

O modo de sincronização local requer um navegador que ofereça suporte ao acesso ao sistema de arquivosAPI. Para obter mais informações, consulte [Permitir acesso à página da web a arquivos locais no Infrastructure Composer](#).

## Ativando o modo de sincronização local

O modo de sincronização local está desativado por padrão. Você pode ativar o modo de sincronização local por meio do menu Infrastructure Composer.



Para obter instruções sobre como ativar a sincronização local e os projetos de carregamento existentes, veja os tópicos a seguir:.

- [Ative a sincronização local no Infrastructure Composer](#)
- [Carregue um projeto existente do Infrastructure Composer com a sincronização local ativada](#)



## Ative a sincronização local no Infrastructure Composer

Para ativar a sincronização local, conclua as seguintes etapas:

1. Na página [inicial](#) do Infrastructure Composer, selecione Criar projeto.
2. No menu Infrastructure Composer, selecione Ativar sincronização local.
3. Em Localização do projeto, pressione Seleccionar pasta e escolha um diretório. É aqui que o Infrastructure Composer salvará e sincronizará seus arquivos e pastas de modelo conforme você projeta.

### Note

O local do projeto não deve conter um modelo de aplicativo existente.

4. Quando solicitado a permitir o acesso, selecione Exibir arquivos.
5. Pressione Ativar. Quando solicitado a salvar as alterações, selecione Salvar alterações.

Quando ativado, o indicador de salvamento automático será exibido na área superior esquerda da tela.

## Carregue um projeto existente do Infrastructure Composer com a sincronização local ativada

Para carregar um projeto existente com a sincronização local ativada, conclua as seguintes etapas:

1. Na página [inicial](#) do Infrastructure Composer, selecione Carregar um AWS CloudFormation modelo.
2. No menu Infrastructure Composer, selecione Abrir > Pasta do projeto.
3. Em Localização do projeto, pressione Seleccionar pasta e escolha a pasta raiz do seu projeto.
4. Quando solicitado a permitir o acesso, selecione Exibir arquivos.
5. Em Arquivo de modelo, selecione seu modelo de aplicativo e pressione Criar.
6. Quando solicitado a salvar as alterações, selecione Salvar alterações.

Quando ativado, o indicador de salvamento automático será exibido na área superior esquerda da tela.

## Importe funções para o Infrastructure Composer a partir do console Lambda

O Infrastructure Composer fornece uma integração com o AWS Lambda console. Você pode importar uma função Lambda do console Lambda para o console do Infrastructure Composer. Em seguida, use a tela do Infrastructure Composer para projetar ainda mais a arquitetura do seu aplicativo.

- Essa integração requer um navegador que ofereça suporte ao acesso ao sistema de arquivos API. Para obter mais informações, consulte [Permitir acesso à página da web a arquivos locais no Infrastructure Composer](#).
- Ao importar sua função Lambda para o Infrastructure Composer, você deve ativar o modo de sincronização local para salvar as alterações. Para obter mais informações, consulte [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#).

Para começar a usar essa integração, consulte [Usando AWS Lambda com AWS Infrastructure Composer](#) no Guia do AWS Lambda desenvolvedor.

## Exportar uma imagem da tela visual do Infrastructure Composer

Este tópico descreve o recurso de tela de exportação do AWS Infrastructure Composer console.

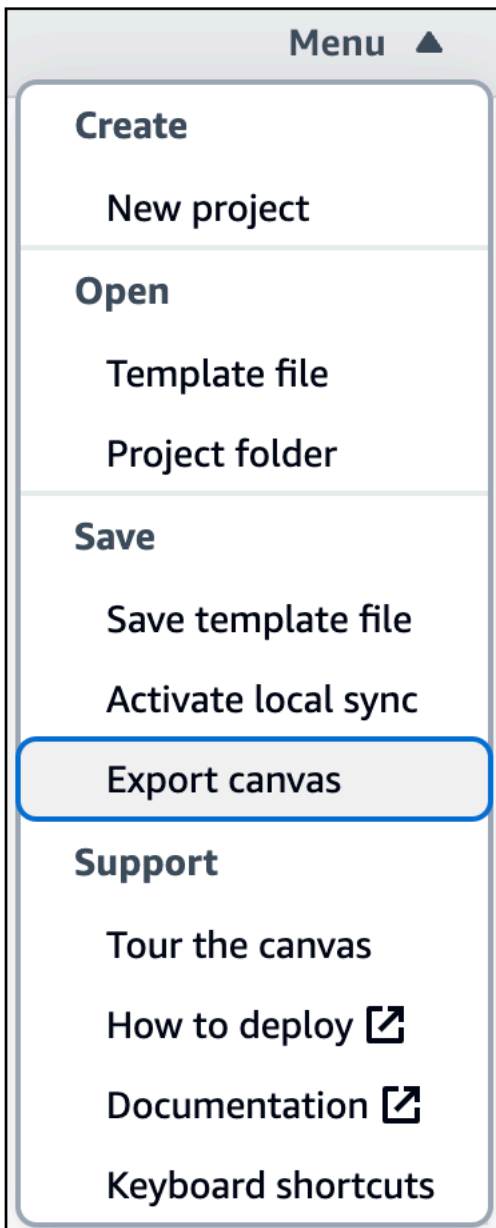
Para obter uma visão geral visual de todos os recursos do Infrastructure Composer, consulte [AWS Infrastructure Composer visão geral visual do console](#).

### Sobre a tela de exportação

O recurso de exportação de tela exporta a tela do seu aplicativo como uma imagem para sua máquina local.

- O Infrastructure Composer remove os elementos da interface do usuário do designer visual e exporta somente o diagrama do seu aplicativo.
- O formato de arquivo de imagem padrão é png.
- O arquivo é exportado para o local de download padrão da sua máquina local.

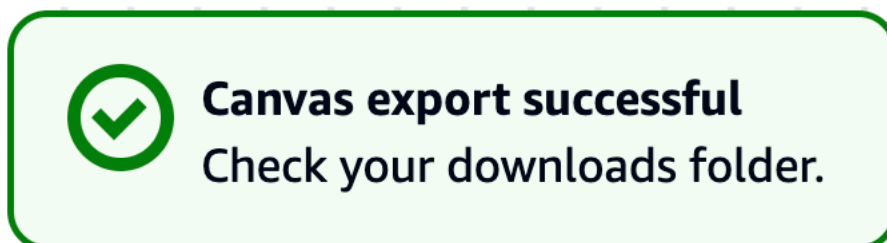
Você pode acessar o recurso de exportação da tela no Menu.



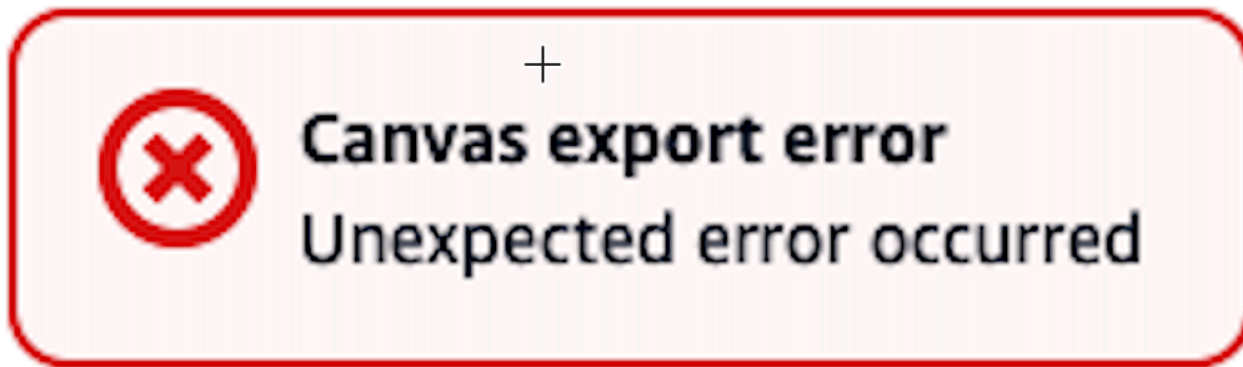
## Exportando tela

Quando você exporta sua tela, o Infrastructure Composer exibe uma mensagem de status.

Se a exportação for bem-sucedida, você verá a seguinte mensagem:



Se a exportação não tiver sido bem-sucedida, você verá uma mensagem de erro. Se você receber uma mensagem de erro, tente exportar novamente.



## Usando o Infrastructure Composer no modo CloudFormation console

O Infrastructure Composer no modo CloudFormation console é a ferramenta recomendada para visualizar seus AWS CloudFormation modelos. Você também pode usar essa ferramenta para criar e editar AWS CloudFormation modelos.

### Em que esse modo difere do console padrão do Infrastructure Composer?

O Infrastructure Composer no modo CloudFormation console geralmente tem a mesma funcionalidade do [console padrão do Infrastructure Composer](#), mas há algumas diferenças a serem observadas.

- Esse modo é integrado ao fluxo de trabalho da pilha no AWS CloudFormation console. Isso permite que você use o Infrastructure Composer diretamente no AWS CloudFormation.
- [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#), um recurso que sincroniza e salva dados automaticamente em sua máquina local, não é suportado.
- Os cartões relacionados ao Lambda (Lambda Function e Lambda Layer) exigem compilações de código e soluções de empacotamento que não estão disponíveis nesse modo.

#### Note

Esses cartões e a sincronização local podem ser usados no [console do Infrastructure Composer](#) ou no AWS Toolkit for Visual Studio Code.

Quando você abre o Infrastructure Composer a partir do AWS CloudFormation console, o Infrastructure Composer é aberto no modo CloudFormation console. Nesse modo, você pode usar o Infrastructure Composer para visualizar, criar e atualizar seus modelos.

## Como acessar o Infrastructure Composer no modo CloudFormation console

O Infrastructure Composer no modo CloudFormation console é uma atualização do AWS CloudFormation Designer. Recomendamos usar o Infrastructure Composer para visualizar seus AWS CloudFormation modelos. Você também pode usar essa ferramenta para criar e editar AWS CloudFormation modelos.

1. Acesse o [console do Cloudformation](#) e faça login.
2. Selecione Infrastructure Composer no menu de navegação do lado esquerdo. Isso o levará ao Infrastructure Composer no modo CloudFormation console.

### Note

Para obter informações sobre como usar o Infrastructure Composer no modo CloudFormation console, consulte [Usando o Infrastructure Composer no modo CloudFormation console](#).

## Visualize uma implantação no Infrastructure Composer no modo console CloudFormation

Siga as instruções neste tópico para visualizar um modelo implantado do AWS CloudFormation Stack/Infrastructure Composer.

1. Vá para o [AWS CloudFormation console](#) e faça o login.
2. Selecione a pilha que você deseja editar.
3. Selecione a guia Modelo.
4. Selecione Infrastructure Composer.

O Infrastructure Composer visualizará sua pilha/modelo. Mudanças também podem ser feitas aqui.

# Crie um novo modelo no Infrastructure Composer no modo CloudFormation console

Siga as instruções neste tópico para criar um novo modelo.

1. Vá para o [AWS CloudFormation console](#) e faça o login.
2. Selecione Infrastructure Composer no menu de navegação do lado esquerdo. Isso abrirá o Infrastructure Composer no modo CloudFormation console.
3. Arraste, solte, configure e conecte os recursos ([cartões](#)) necessários na paleta Recursos.

## Note

Consulte [Como compor](#) para obter detalhes sobre o uso do Infrastructure Composer e observe que os cartões relacionados ao Lambda (Lambda Function e Lambda Layer) exigem compilações de código e soluções de empacotamento que não estão disponíveis no Infrastructure Composer no modo console. CloudFormation Essas placas podem ser usadas no [console do Infrastructure Composer](#) ou no AWS Toolkit for Visual Studio Code. Para obter informações sobre o uso dessas ferramentas, consulte [Onde você pode usar o Infrastructure Composer](#).

4. Clique duas vezes nos cartões para usar o painel Propriedades do recurso para especificar como os cartões são configurados.
5. [Conecte seus cartões](#) para especificar o fluxo de trabalho orientado por eventos do seu aplicativo.
6. Selecione Modelo para visualizar e editar seu código de infraestrutura. As alterações são sincronizadas automaticamente com sua visualização em tela.
7. Quando seu modelo estiver pronto para ser exportado para uma pilha, selecione Criar modelo.
8. Selecione o CloudFormation botão Confirmar e exportar para. Isso o levará de volta ao fluxo de trabalho de criação da pilha com uma mensagem confirmando que seu modelo foi importado com sucesso.

## Note

Somente modelos com recursos podem ser exportados.

9. No fluxo de trabalho Criar pilha, selecione Avançar.


10. Forneça um nome de pilha, revise todos os parâmetros listados e selecione Avançar.

 Note

O nome da pilha deve começar com uma letra e conter somente letras, números e traços.

11. Selecione Avançar depois de fornecer as seguintes informações:

- Tags associadas à pilha
- Permissões de pilha
- As opções de falha da pilha

 Note


Para obter orientação sobre o gerenciamento de pilhas, consulte [as AWS CloudFormation melhores práticas](#) no Guia do AWS CloudFormation usuário.

12. Confirme se os detalhes da pilha estão corretos, verifique as confirmações na parte inferior da página e selecione o botão Enviar.

AWS CloudFormation começará a criar a pilha com base nos dados do seu modelo.

## Atualize uma pilha existente no Infrastructure Composer no modo console CloudFormation


Siga as instruções neste tópico para atualizar uma AWS CloudFormation pilha existente.

 Note

Se seu arquivo for salvo localmente, recomendamos o uso [AWS Toolkit for Visual Studio Code](#).


1. Vá para o [AWS CloudFormation console](#) e faça o login.
2. Selecione a pilha que você deseja editar.
3. Selecione o botão Atualizar. Isso levará você ao assistente da pilha de atualizações.

4. À direita, selecione Editar no Infrastructure Composer.
5. Selecione o botão abaixo chamado Editar no Infrastructure Composer. Isso o levará ao Infrastructure Composer no modo CloudFormation console.
6. Aqui, você pode arrastar, soltar, configurar e conectar recursos ([cartões](#)) da paleta Recursos.

 Note

Consulte [Como compor](#) para obter detalhes sobre o uso do Infrastructure Composer e observe que os cartões relacionados ao Lambda (Lambda Function e Lambda Layer) exigem compilações de código e soluções de empacotamento que não estão disponíveis no Infrastructure Composer no modo console. CloudFormation Essas placas podem ser usadas no [console do Infrastructure Composer](#) ou no AWS Toolkit for Visual Studio Code. Para obter informações sobre o uso dessas ferramentas, consulte [Onde você pode usar o Infrastructure Composer](#).

7. Quando você estiver pronto para exportar as alterações para AWS CloudFormation, selecione Atualizar modelo.
8. Selecione Confirmar e continue CloudFormation. Isso o levará de volta ao fluxo de trabalho da pilha de atualizações com uma mensagem confirmando que seu modelo foi importado com sucesso.

 Note

Somente modelos com recursos podem ser exportados.

9. No fluxo de trabalho Atualizar pilha, selecione Avançar.
10. Revise todos os parâmetros listados e selecione Avançar.
11. Selecione Avançar depois de fornecer as seguintes informações:
  - Tags associadas à pilha
  - Permissões de pilha
  - As opções de falha da pilha



**Note**

Para obter orientação sobre o gerenciamento de pilhas, consulte [as AWS CloudFormation melhores práticas](#) no Guia do AWS CloudFormation usuário.

12. Confirme se os detalhes da pilha estão corretos, verifique as confirmações na parte inferior da página e selecione o botão Enviar.

AWS CloudFormation começará a atualizar a pilha com base nas atualizações que você fez no seu modelo.

## Usando o Infrastructure Composer do AWS Toolkit for Visual Studio Code

Esta seção descreve como você pode usar AWS Infrastructure Composer a partir do [AWS Toolkit for Visual Studio Code](#). Isso inclui uma visão geral visual do Infrastructure Composer a AWS Toolkit for Visual Studio Code partir do. Também inclui instruções que mostram como você pode acessar essa experiência e sincronizar seu projeto do VS Code com a AWS nuvem. Para sincronizar, você usa o sam sync comando do AWS SAM CLI. Esta seção também fornece orientação sobre o uso Amazon Q enquanto estiver no Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code.

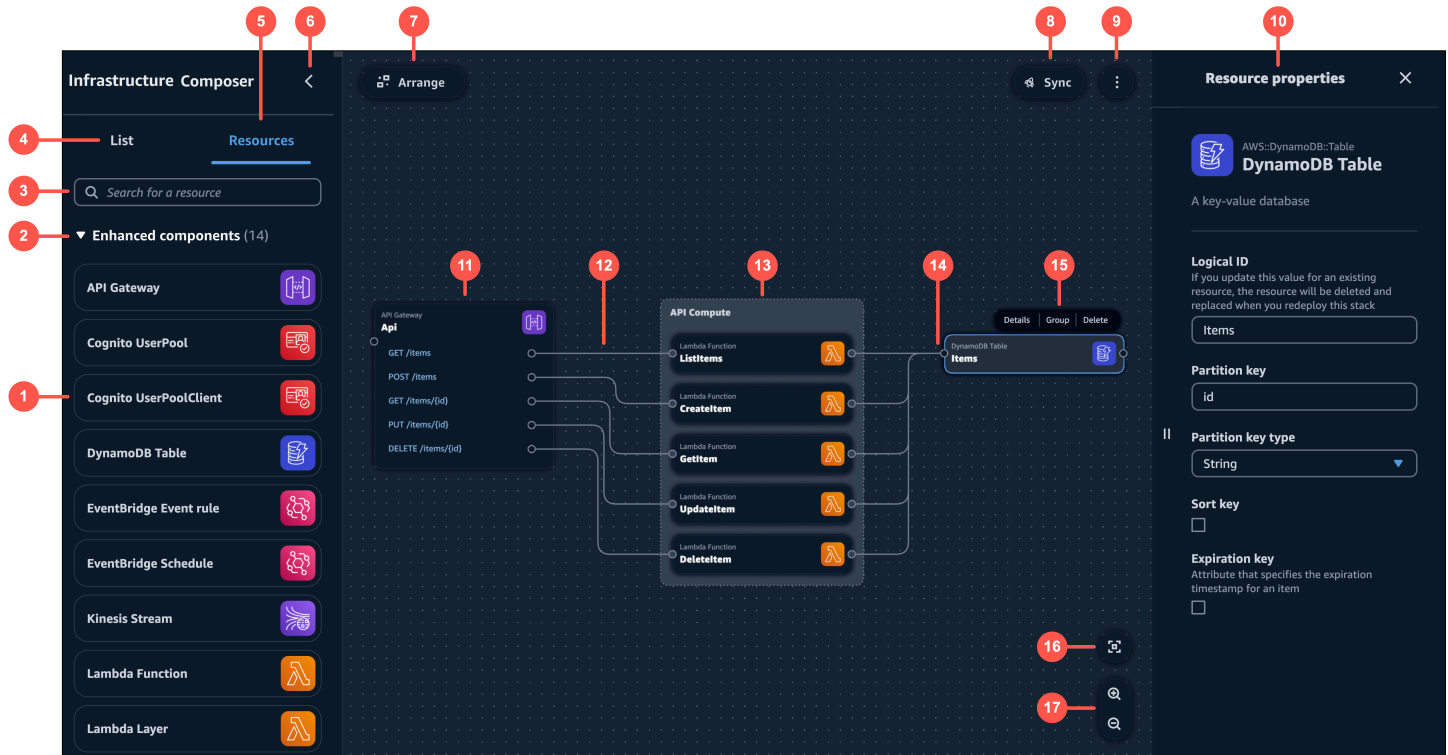
Para obter orientações adicionais sobre o uso do Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code, consulte. [Como compor](#) O conteúdo desta seção se aplica a essa experiência, bem como à experiência do console do Infrastructure Composer.

### Tópicos

- [Visão geral visual do Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code](#)
- [Acesse o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code](#)
- [Sincronize o Infrastructure Composer para implantar no Nuvem AWS](#)
- [Usando AWS Infrastructure Composer com Amazon Q Developer](#)

## Visão geral visual do Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code

O designer visual do Infrastructure Composer no AWS Toolkit for Visual Studio Code inclui uma tela visual, que inclui componentes numerados na imagem a seguir e listados abaixo.



1. Paleta de recursos — Exibe cartões com os quais você pode criar designs.
2. Categorias de cartas — As cartas são organizadas por categorias exclusivas do Infrastructure Composer.
3. Barra de pesquisa de recursos — Pesquise cartões que você pode adicionar à tela.
4. Lista — Exibe uma visualização em árvore dos recursos do seu aplicativo.
5. Recursos — Exibe a paleta de recursos.
6. Alternar no painel esquerdo — Ocultar ou mostrar o painel esquerdo.
7. Organizar — Organiza a arquitetura do seu aplicativo na tela.
8. Sincronizar — inicia o AWS Serverless Application Model (AWS SAMCLI `aws sam sync` comando para implantar seu aplicativo.
9. Menu — Fornece opções gerais, como as seguintes:
  - Exportar tela

- Visite a tela
- Links para a documentação
- Atalhos de teclado

10 Painel de propriedades do recurso — Exibe as propriedades relevantes do cartão que foi selecionado na tela. Esse painel é dinâmico. As propriedades exibidas mudarão conforme você configura seu cartão.

11 Cartão — Exibe uma visualização do seu cartão na tela.

12 Linha — Representa uma conexão entre cartões.

13 Grupo — Um grupo de cartas. Você pode agrupar cartões para organização visual.

14 Porta — Pontos de conexão com outras placas.

15 Ações do cartão — Fornece ações que você pode realizar no seu cartão.

- Detalhes — Abre o painel de propriedades do recurso.
- Grupo — Agrupe as cartas selecionadas.
- Excluir — Exclui o cartão da tela e do modelo.

16 Centralizar novamente — centralize novamente o diagrama do aplicativo na tela visual.

17 Zoom — Aumente e diminua o zoom em sua tela.

## Acesse o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code

Siga as instruções neste tópico para acessar o Infrastructure Composer a AWS Toolkit for Visual Studio Code partir do.

### Note

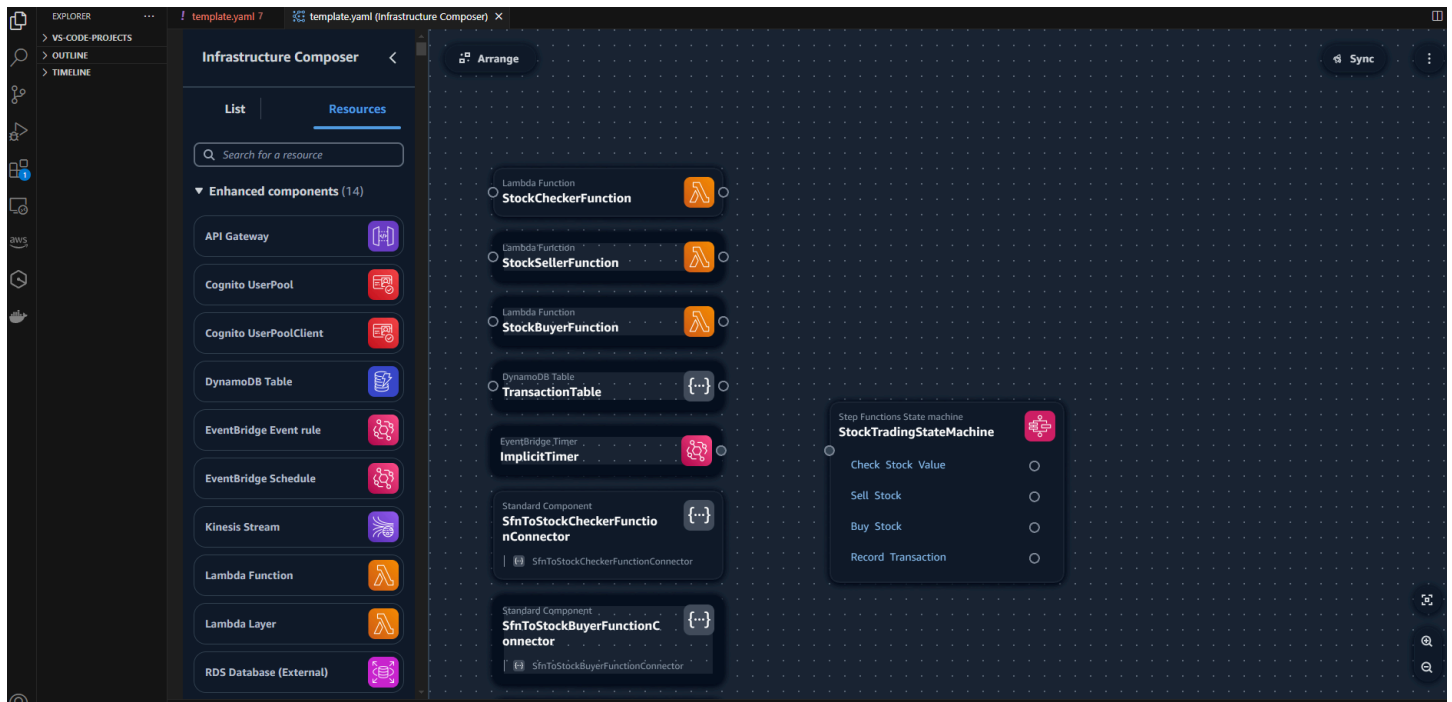
Antes de acessar o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code, você deve primeiro baixar e instalar o Toolkit for VS Code. Para obter instruções, consulte Como [baixar o kit de ferramentas para VS Code](#).

Para acessar o Infrastructure Composer a partir do Toolkit for VS Code

Você pode acessar o Infrastructure Composer de qualquer uma das seguintes formas:

1. Ao selecionar o botão Infrastructure Composer de AWS CloudFormation qualquer AWS SAM modelo.
2. Acesse o menu de contexto clicando com o botão direito do mouse em seu AWS SAM modelo AWS CloudFormation ou modelo.
3. Da paleta de comandos do VS Code.

Veja a seguir um exemplo de como acessar o Infrastructure Composer a partir do botão Infrastructure Composer:



Para obter mais informações sobre como acessar o Infrastructure Composer, consulte [Acessando a AWS Infrastructure Composer partir do kit de ferramentas](#).

## Sincronize o Infrastructure Composer para implantar no Nuvem AWS

Use o botão de sincronização AWS Infrastructure Composer do AWS Toolkit for Visual Studio Code para implantar seu aplicativo no Nuvem AWS.

O botão de sincronização inicia o `sam sync` comando a partir da interface de linha de AWS SAM comando (CLI).

O `sam sync` comando pode implantar novos aplicativos ou sincronizar rapidamente as alterações feitas localmente no Nuvem AWS. A execução `sam sync` pode incluir o seguinte:

- Construindo seu aplicativo com `sam build` o objetivo de preparar seus arquivos de aplicativos locais para implantação, criando ou atualizando um `.aws-sam` diretório local.
- Para recursos que oferecem suporte ao AWS serviço APIs, o AWS SAM CLI usará o APIs para implantar suas alterações. O AWS SAM CLI faz isso para atualizar rapidamente seus recursos na nuvem.
- Se necessário, o AWS SAM CLI executa uma AWS CloudFormation implantação para atualizar toda a pilha por meio de um conjunto de alterações.

O `sam sync` comando é mais adequado para ambientes de desenvolvimento rápido, pois a atualização rápida de seus recursos de nuvem pode beneficiar seus fluxos de trabalho de desenvolvimento e teste.

Para saber mais sobre `sam sync`, consulte Como [usar o sam sync](#) no Guia do AWS Serverless Application Model desenvolvedor.

## Configurar

Para usar o recurso de sincronização no Infrastructure Composer, você deve ter o AWS SAM CLI instalado em sua máquina local. Para obter instruções, consulte [Instalando o AWS SAM CLI](#) no Guia do desenvolvedor do AWS Serverless Application Model .

Quando você usa o recurso de sincronização no Infrastructure Composer, o AWS SAM CLI faz referência ao seu arquivo de configuração para obter as informações necessárias para sincronizar seu aplicativo com Nuvem AWS o. Para obter instruções sobre como criar, modificar e usar arquivos de configuração, consulte Definir [as configurações do projeto](#) no Guia do AWS Serverless Application Model desenvolvedor.

## Sincronize e implante seu aplicativo

Para sincronizar seu aplicativo com o Nuvem AWS

1. Selecione o botão de sincronização na tela do Infrastructure Composer.
2. Você pode receber uma solicitação para confirmar que está trabalhando com uma pilha de desenvolvimento. Selecione OK para continuar.
3. O Infrastructure Composer pode solicitar que você configure as seguintes opções:
  - Região da AWS— A região com a qual sincronizar seu aplicativo.

- AWS CloudFormation nome da pilha — O nome da sua AWS CloudFormation pilha. Você pode selecionar um nome de pilha existente ou criar um novo.
- Bucket Amazon Simple Storage Service (Amazon S3) — O nome do seu bucket Amazon S3. O AWS SAM CLI empacotará e armazenará os arquivos do aplicativo e o código da função aqui. Você pode selecionar um bucket existente ou criar um novo.

O Infrastructure Composer iniciará o AWS SAM CLI `sam sync` e abra uma janela de terminal em seu IDE para exibir seu progresso.

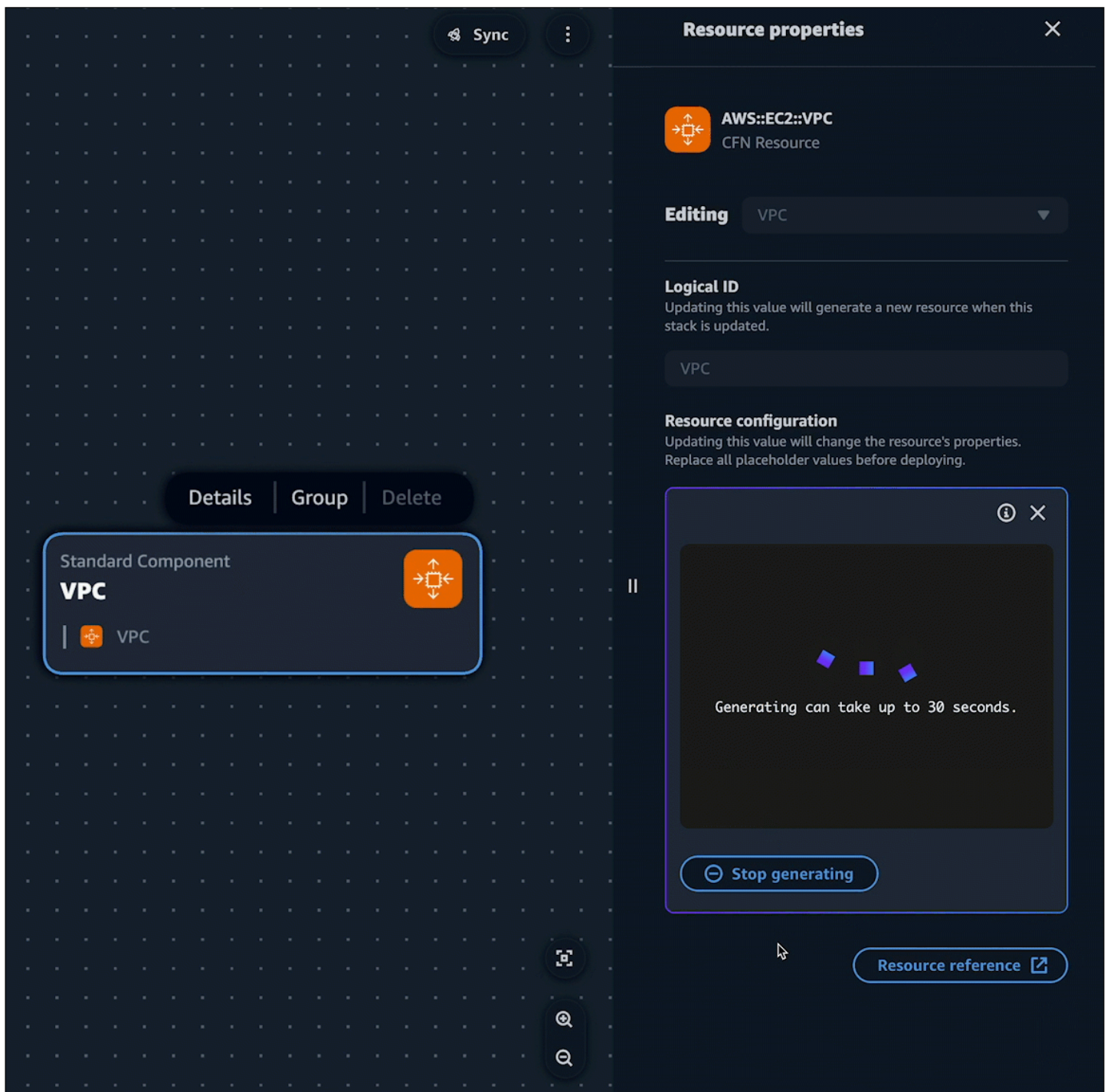
## Usando AWS Infrastructure Composer com Amazon Q Developer

AWS Infrastructure Composer do AWS Toolkit for Visual Studio Code fornece uma integração com Amazon Q. Você pode usar Amazon Q no Infrastructure Composer para gerar o código de infraestrutura para seus AWS recursos à medida que você projeta seu aplicativo.

Amazon Q é um gerador de código de uso geral baseado em aprendizado de máquina. Para saber mais, consulte [O que é Amazon Q?](#) no Amazon Q Developer Guia do usuário.

Para placas padrão de recursos e componentes padrão, você pode usar Amazon Q para gerar sugestões de código de infraestrutura para seus recursos.





As placas padrão de recursos e componentes padrão podem representar um AWS CloudFormation recurso ou uma coleção de AWS CloudFormation recursos. Para saber mais, consulte [Configurar e modificar cartões no Infrastructure Composer](#).

## Configuração

Para usar Amazon Q no Infrastructure Composer, você deve se autenticar com Amazon Q no kit de ferramentas. Para obter instruções, consulte [Introdução ao Amazon Q no VS Code e JetBrains](#) no Amazon Q Developer Guia do usuário.

### O uso do Amazon Q Developer no Infrastructure Composer

Você pode usar: Amazon Q Developer no painel Propriedades do recurso de qualquer recurso padrão ou placa de componente padrão.

Para usar Amazon Q no Infrastructure Composer

1. Em um recurso padrão ou cartão de componente padrão, abra o painel Propriedades do recurso.
2. Localize o campo Configuração do recurso. Esse campo contém o código de infraestrutura do cartão.
3. Selecione o botão Gerar sugestões. Amazon Q gerará uma sugestão.

#### Note

O código gerado nesse estágio não substituirá o código de infraestrutura existente do seu modelo.

4. Para gerar mais sugestões, selecione Regenerar. Você pode alternar entre as amostras para comparar os resultados.
5. Para selecionar uma opção, escolha Selecionar. Você pode modificar o código aqui antes de salvá-lo em seu aplicativo. Para sair sem salvar, selecione o ícone de saída (X).
6. Para salvar o código no seu modelo de aplicativo, selecione Salvar no painel Propriedades do recurso.

### Saiba mais

Para saber mais a respeito Amazon Q, veja [O que é Amazon Q?](#) no Amazon Q Developer Guia do usuário.



# Como compor em AWS Infrastructure Composer

Esta seção aborda os conceitos básicos do uso do Infrastructure Composer a partir do [Console Infrastructure Composer](#), [CloudFormation modo de console](#), e o. [AWS Toolkit for Visual Studio Code](#). Mais especificamente, os tópicos desta seção fornecem detalhes importantes sobre como compor um aplicativo com o Infrastructure Composer e incluem detalhes sobre recursos e atalhos adicionais. Há algumas variações na funcionalidade entre as experiências do console e do VS Code, e os tópicos desta seção identificam e descrevem essas variações onde elas ocorrem.

Depois de compor seu aplicativo, você estará pronto [Implemente seu aplicativo sem servidor Infrastructure Composer na nuvem AWS](#) para revisar as informações sobre como implantar seu aplicativo.

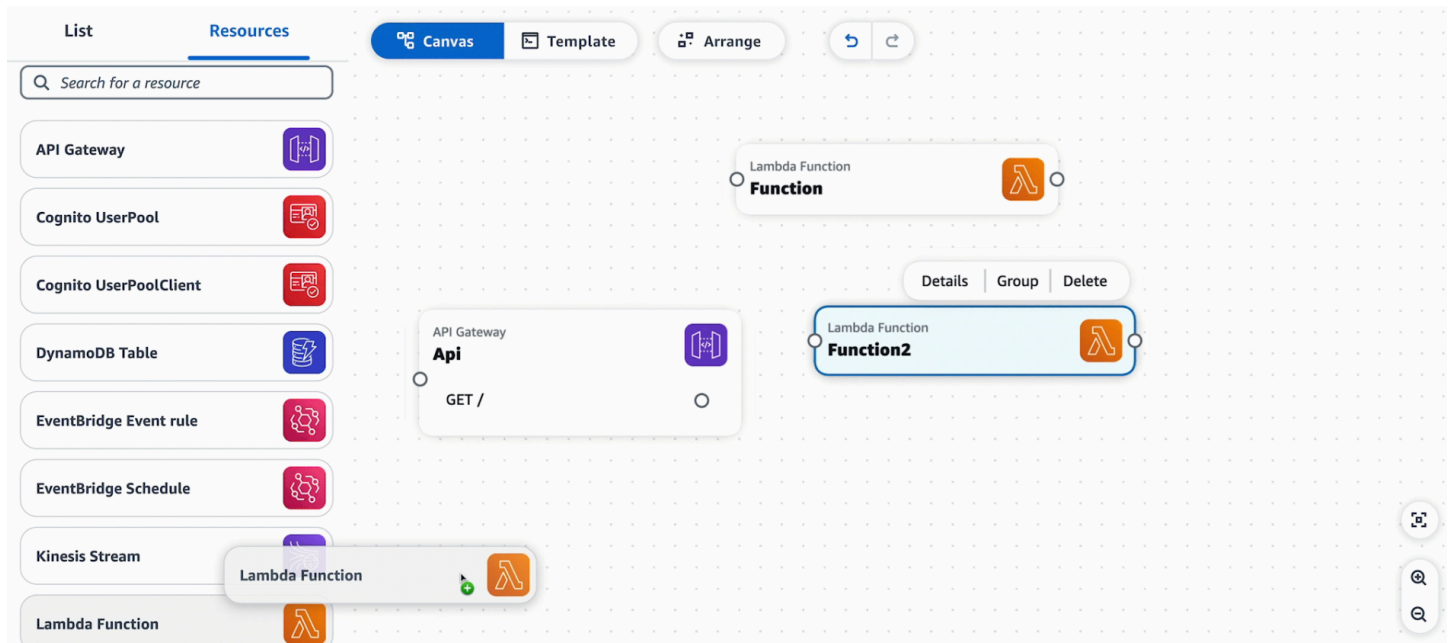
## Tópicos

- [Coloque cartões na tela visual do Infrastructure Composer](#)
- [Agrupe cartões na tela visual do Infrastructure Composer](#)
- [Cartões Connect na tela visual do Infrastructure Composer](#)
- [Desconecte as placas no Infrastructure Composer](#)
- [Organize os cartões na tela visual do Infrastructure Composer](#)
- [Configurar e modificar cartões no Infrastructure Composer](#)
- [Excluir cartões no Infrastructure Composer](#)
- [Veja as atualizações de código com o Change Inspector no Infrastructure Composer](#)
- [Faça referência a arquivos externos no Infrastructure Composer](#)
- [Integre o Infrastructure Composer com a Amazon Virtual Private Cloud \(AmazonVPC\)](#)

## Coloque cartões na tela visual do Infrastructure Composer

Esta seção descreve como você seleciona e arrasta os [cartões](#) do Infrastructure Composer em sua tela visual. Antes de começar, identifique quais recursos seu aplicativo precisa e como eles precisam interagir. Para obter dicas sobre como fazer isso, consulte [Crie seu primeiro aplicativo com o Infrastructure Composer](#).

Para adicionar um cartão ao seu aplicativo, arraste-o da paleta de recursos e solte-o na tela visual.



Você pode escolher entre dois tipos de placas: placas de [componentes aprimoradas e placas de recursos IaC padrão](#).

Depois de colocar seus cartões na tela visual, você estará pronto para agrupar, conectar, organizar e configurar seus cartões. Consulte os tópicos a seguir para obter informações sobre como fazer isso:

- [Agrupe cartões na tela visual do Infrastructure Composer](#)
- [Cartões Connect na tela visual do Infrastructure Composer](#)
- [Organize os cartões na tela visual do Infrastructure Composer](#)
- [Configurar e modificar cartões no Infrastructure Composer](#)

## Agrupe cartões na tela visual do Infrastructure Composer

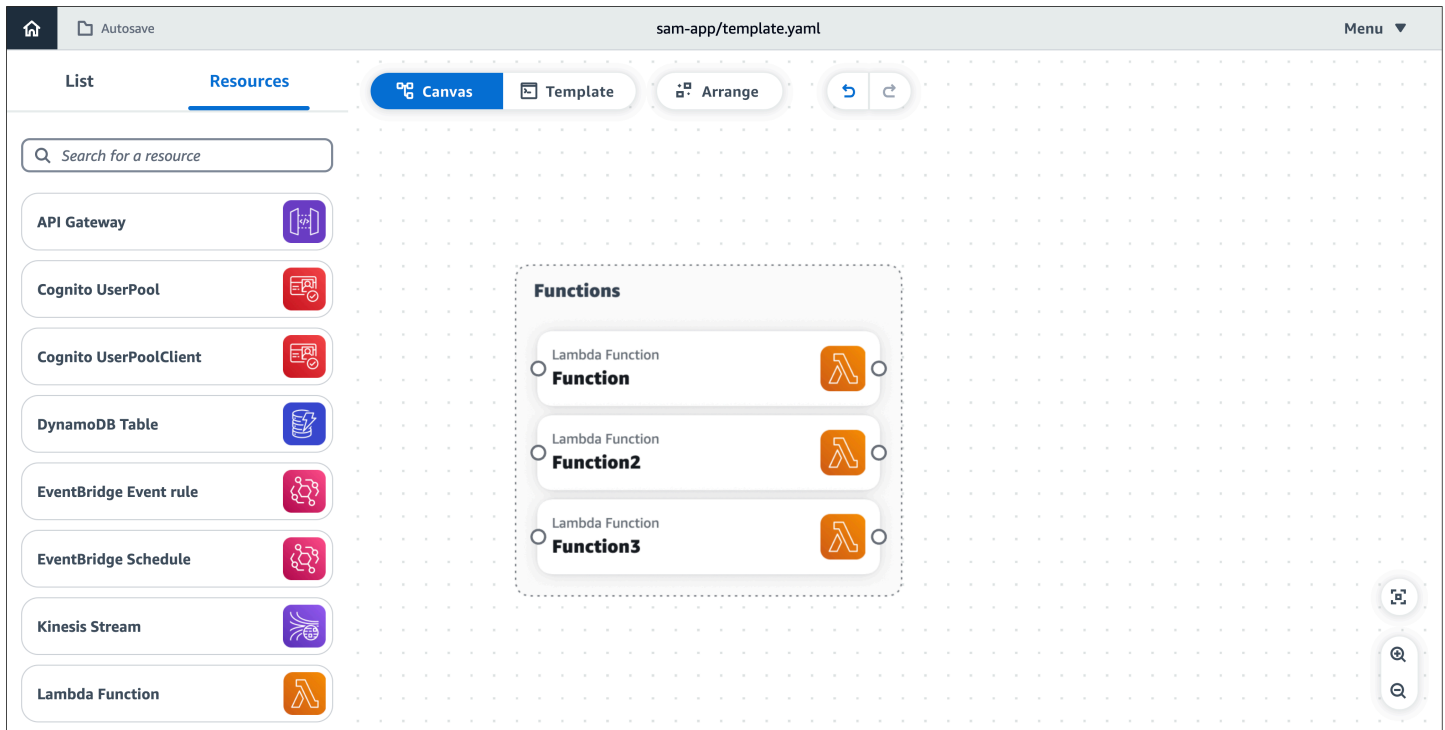
Este tópico contém detalhes sobre o agrupamento de placas de componentes aprimoradas e placas de componentes padrão. Os cartões de agrupamento ajudam você a categorizar e organizar seus recursos sem precisar pensar no código ou na marcação que você precisa escrever.

### Agrupamento de placas de componentes aprimoradas

Há duas maneiras de agrupar placas de componentes aprimoradas:

- Ao pressionar Shift, selecione os cartões para agrupar. Em seguida, escolha Grupo no menu de ações do recurso.

- selecione um cartão que você deseja em um grupo. No menu exibido, selecione Grupo. Isso criará um grupo no qual você poderá arrastar e soltar outras cartas.



## Agrupando uma placa de componente padrão em outra

O exemplo a seguir mostra uma forma pela qual uma placa de componente padrão pode ser agrupada em outra placa a partir do painel de propriedades do recurso:

The screenshot displays the AWS Infrastructure Composer interface. On the left, a grid contains a 'Standard Component' card for 'Function'. The card has a blue border and includes a 'Role' icon and a 'Function' icon. Above the card are buttons for 'Details', 'Group', and 'Delete'. On the right, the 'Resource properties' panel is open, showing the 'AWS::Lambda::Function' resource. The panel includes an 'Editing' section with a dropdown menu set to 'Function', a 'Role' dropdown, and a 'Logical ID' dropdown set to 'Function'. Below these is a text input field containing 'Function'. The 'Resource configuration' section contains a code editor with the following content:

```
Code: {}
Role: !Ref Role
```

At the bottom right of the panel is a 'Resource reference' button with an external link icon.

No campo Configuração do recurso no painel Propriedades do recurso, o Role foi referenciado na função Lambda. Isso faz com que o cartão de função seja agrupado no cartão de função na tela.

## Cartões Connect na tela visual do Infrastructure Composer

Use este tópico para entender como conectar placas no Infrastructure Composer. Esta seção inclui detalhes sobre como conectar placas de componentes aprimoradas e placas de componentes

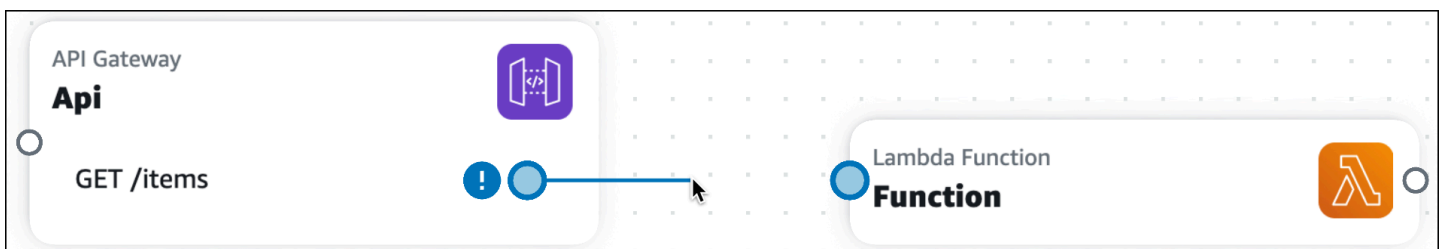
padrão. Ele também fornece alguns exemplos que ilustram as diferentes maneiras pelas quais os cartões podem ser conectados.

## Conectando placas de componentes aprimoradas

Nas placas de componentes aprimoradas, as portas identificam visualmente onde as conexões podem ser feitas.

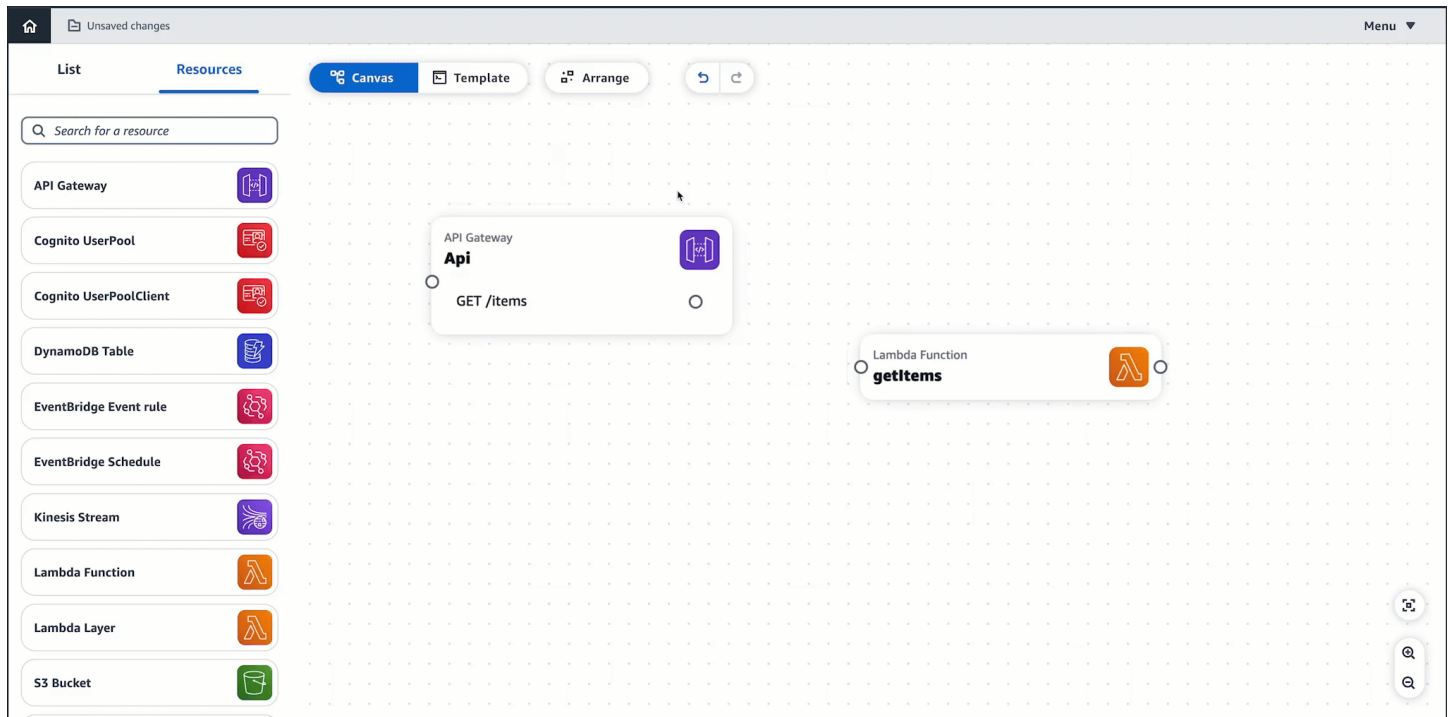
- Uma porta no lado direito de um cartão indica uma oportunidade para o cartão invocar outro cartão.
- Uma porta no lado esquerdo de um cartão indica uma oportunidade para o cartão ser invocado por outro cartão.

Conecte as placas clicando na porta direita de uma placa e arrastando-a para a porta esquerda de outra placa.



Quando você cria uma conexão, uma mensagem é exibida informando se a conexão foi feita com sucesso. Selecione a mensagem para ver o que o Infrastructure Composer alterou para provisionar uma conexão. Se a conexão não foi bem-sucedida, você pode selecionar Visualização de modelo para atualizar manualmente seu código de infraestrutura para provisionar a conexão.

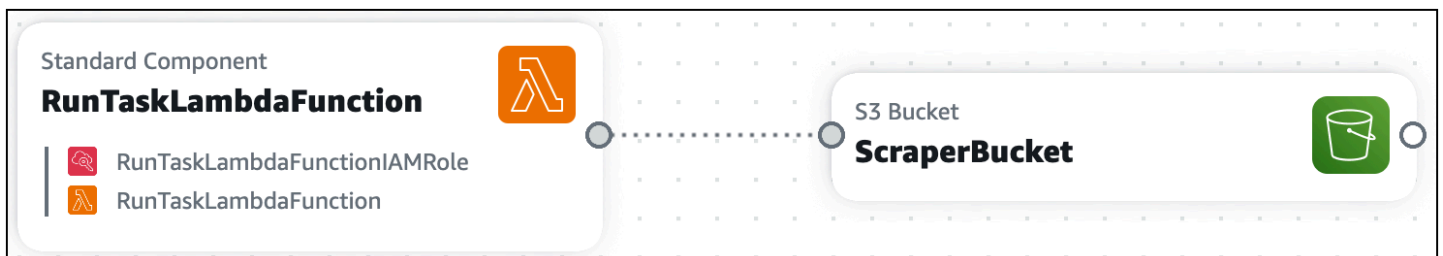
- Quando for bem-sucedido, clique na mensagem para ver o Inspetor de alterações. Aqui, você pode ver o que o Infrastructure Composer modificou para provisionar sua conexão.
- Quando não for bem-sucedida, uma mensagem será exibida. Você pode selecionar a visualização Modelo e atualizar manualmente seu código de infraestrutura para provisionar a conexão.



Quando você conecta placas de componentes aprimoradas, o Infrastructure Composer cria automaticamente o código de infraestrutura em seu modelo para provisionar o relacionamento orientado por eventos entre seus recursos.

## Conectando placas de componentes padrão (placas de recursos IaC padrão)

As placas de recursos IaC padrão não incluem portas para criar conexões com outros recursos. Durante a [configuração da placa](#), você especifica relacionamentos orientados por eventos no modelo de seu aplicativo. O Infrastructure Composer detectará automaticamente essas conexões e as visualizará com uma linha pontilhada entre suas placas. Veja a seguir um exemplo de conexão entre uma placa de componente padrão e uma placa de componente aprimorada:



O exemplo a seguir mostra como uma função Lambda pode ser conectada a um Amazon API Gateway rest: API

```
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
```

No exemplo acima, o trecho de código listado `ApiGatewayMethod`: abaixo `Integration`: especifica a relação orientada por eventos que conecta os dois cartões.

## Exemplos para conectar placas no Infrastructure Composer

Use os exemplos nesta seção para entender como as placas podem ser conectadas no Infrastructure Composer.

### Invoque uma AWS Lambda função quando um item é colocado em um bucket do Amazon Simple Storage Service (Amazon S3)

Neste exemplo, um cartão bucket do Amazon S3 está conectado a um cartão de função Lambda. Quando um item é colocado no bucket do Amazon S3, a função é invocada. A função pode então ser usada para processar o item ou acionar outros eventos em seu aplicativo.



Essa interação exige que um evento seja definido para a função. Aqui está o que o Infrastructure Composer fornece:

```
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

```

## Invocar um bucket do Amazon S3 a partir de uma função Lambda

Neste exemplo, um cartão de função Lambda invoca um cartão bucket do Amazon S3. A função Lambda pode ser usada para realizar CRUD operações em itens no bucket do Amazon S3.



Essa interação requer o seguinte, que é provisionado pelo Infrastructure Composer:

- IAM políticas que permitem que a função Lambda interaja com o bucket do Amazon S3.
- Variáveis de ambiente que influenciam o comportamento da função Lambda.

```

Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    ...

```

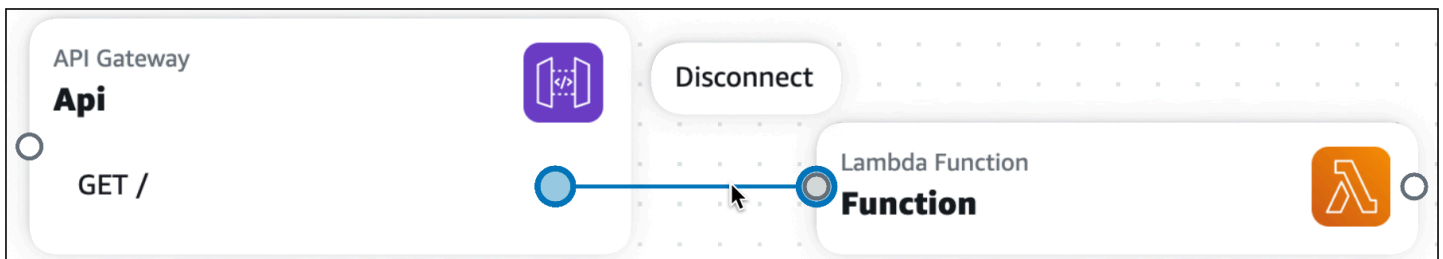
```
MyBucketBucketPolicy:
  Type: AWS::S3::BucketPolicy
  ...
MyFunction:
  Type: AWS::Serverless::Function
  Properties:
    ...
  Environment:
    Variables:
      BUCKET_NAME: !Ref MyBucket
      BUCKET_ARN: !GetAtt MyBucket.Arn
  Policies:
    - S3CrudPolicy:
      BucketName: !Ref MyBucket
```

## Desconecte as placas no Infrastructure Composer

No Infrastructure Composer, você conecta e desconecta AWS recursos usando placas de componentes aprimoradas e placas de componentes padrão. Esta seção descreve como desconectar os dois tipos de cartões.

### Placas de componentes aprimoradas

Para desconectar as placas de componentes aprimoradas, selecione a linha e escolha Desconectar.



O Infrastructure Composer modificará automaticamente seu modelo para remover o relacionamento orientado por eventos do seu aplicativo.

### Placas de componentes padrão

As placas de componentes padrão não incluem portas para criar conexões com outros recursos. Durante a [configuração da placa](#), você especifica relacionamentos orientados por eventos no modelo de seu aplicativo. O Infrastructure Composer detectará automaticamente essas conexões e as

visualizará com uma linha pontilhada entre suas placas. Para desconectar uma placa de componente padrão, remova a relação orientada por eventos no modelo do seu aplicativo.

O exemplo a seguir mostra uma função Lambda conectada a um Amazon API Gateway rest: API

```

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:

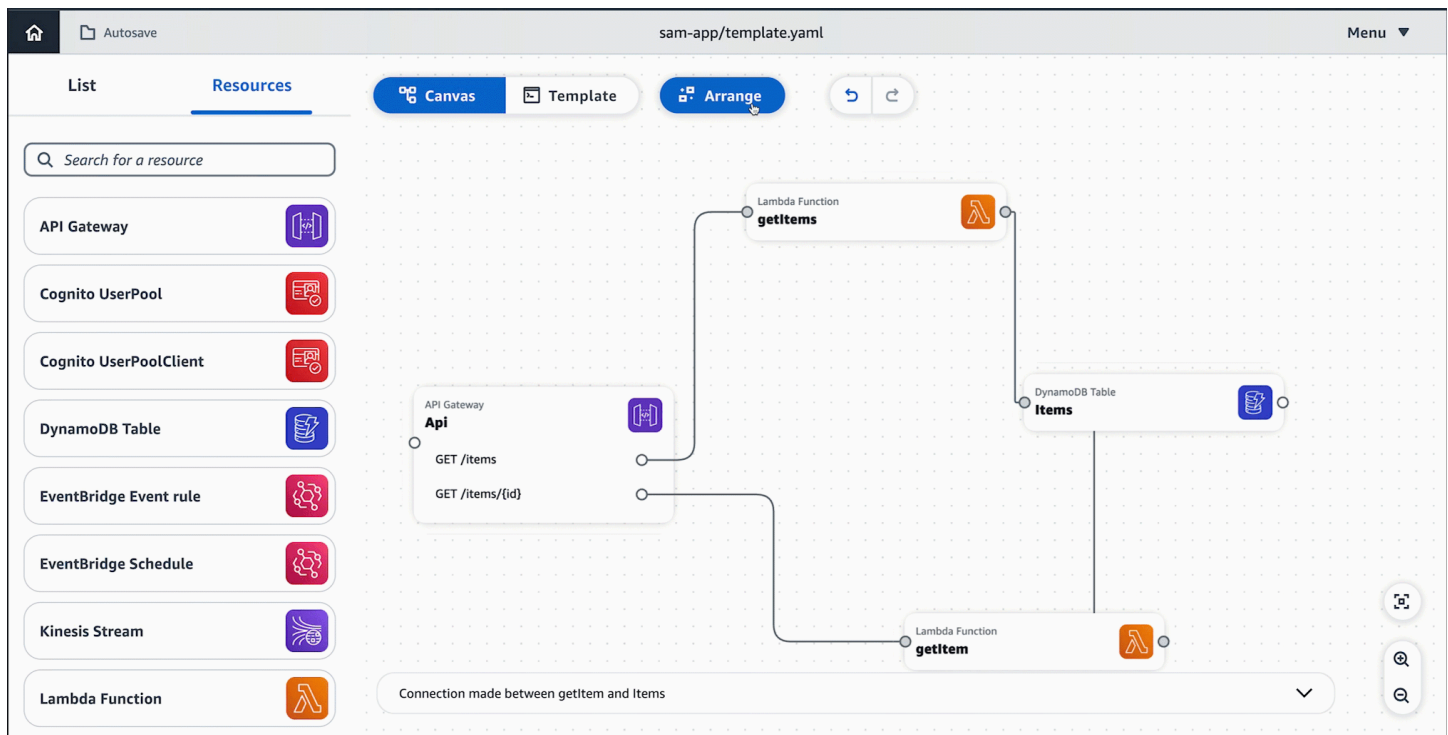
```

```
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
```

Para remover a conexão entre os dois cartões, remova as referências `MyLambdaFunction` listadas `ApiGatewayMethod`: abaixo. `Integration`

## Organize os cartões na tela visual do Infrastructure Composer

Selecione `Organizar` para organizar e organizar visualmente os cartões na tela. Usar o botão `Organizar` é particularmente útil quando há muitos cartões e conexões na tela.



## Configurar e modificar cartões no Infrastructure Composer

No Infrastructure Composer, os cartões representam os recursos que você usa para projetar sua arquitetura de aplicativo. Ao configurar uma placa no Infrastructure Composer, você define os detalhes dos recursos em seu aplicativo. Isso inclui detalhes como a ID lógica e a chave de partição do cartão. A forma como essas informações são definidas varia entre as placas de componentes aprimoradas e as placas padrão.

Um cartão de componente aprimorado é uma coleção de AWS CloudFormation recursos que foram combinados em um único cartão selecionado que aprimora a facilidade de uso e a funcionalidade e foi projetado para uma ampla variedade de casos de uso. Um cartão de recursos padrão do IaC representa um único AWS CloudFormation recurso. Cada cartão de recurso padrão do IaC, uma vez arrastado para a tela, é rotulado como componente padrão.

Este tópico fornece detalhes sobre a configuração das placas de componentes aprimoradas e das placas de componentes padrão.

### Note

Este tópico se aplica ao uso de cartões do console do Infrastructure Composer, da AWS Toolkit for Visual Studio Code extensão e enquanto estiver no Infrastructure Composer no

modo CloudFormation console. Os cartões relacionados ao Lambda (Lambda Function e Lambda Layer) exigem compilações de código e soluções de empacotamento que não estão disponíveis no Infrastructure Composer no modo console. CloudFormation Para obter mais informações, consulte [Usando o Infrastructure Composer no modo CloudFormation console](#).

## Tópicos

- [Placas de componentes aprimoradas no Infrastructure Composer](#)
- [Cartões padrão no Infrastructure Composer](#)

## Placas de componentes aprimoradas no Infrastructure Composer

Para configurar placas de componentes aprimoradas, o Infrastructure Composer fornece um formulário no painel de propriedades do recurso. Este formulário foi criado exclusivamente para orientá-lo na configuração de cada placa de componente aprimorada. Conforme você preenche o formulário, o Infrastructure Composer modifica seu código de infraestrutura.

Algumas placas de componentes aprimoradas têm recursos adicionais. Esta seção analisa os conceitos básicos do uso de placas de componentes aprimoradas e oferece detalhes sobre placas com recursos adicionais.

Para obter mais informações sobre placas de componentes aprimoradas, consulte [Placas de componentes aprimoradas no Infrastructure Composer](#) e [Placas de componentes aprimoradas no Infrastructure Composer](#)

## Procedimento

O painel de propriedades do recurso simplifica a configuração e adiciona guias que simplificam a configuração do cartão. Para usar esse painel, execute as seguintes etapas:

1. Clique duas vezes em um cartão para abrir o painel Propriedades do recurso.
2. Clique em um cartão e selecione Detalhes para abrir o painel de propriedades do recurso.
3. Para Infrastructure Composer a partir do AWS Management Console, selecione Modelo para mostrar o código do seu aplicativo. Configure diretamente daqui.

A imagem a seguir mostra como isso pode ser feito:

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' palette with a search bar and a list of components: API Gateway, Cognito UserPool, Cognito UserPoolClient, DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, and Lambda Function. The central 'Canvas' area displays a YAML template for a resource named 'Table'. The template includes the following content:

```

1 Transform: AWS::Serverless-2016-10-31
2 Resources:
3   Itel:
4     Type: AWS::DynamoDB::Table
5     Properties:
6       AttributeDefinitions:
7         - AttributeName: id
8           AttributeType: S
9       BillingMode: PAY_PER_REQUEST
10      KeySchema:
11        - AttributeName: id
12          KeyType: HASH
13      StreamSpecification:
14        StreamViewType: NEW_AND_OLD_IMAGES

```

On the right, the 'Resource properties' panel for the 'Table' resource is visible. It includes a description 'A key-value database', a 'Logical ID' field with the value 'Table', a 'Partition key' field with the value 'id', a 'Partition key type' dropdown menu set to 'String', a 'Sort key' checkbox which is unchecked, and an 'Expiration key' field with a description: 'Attribute that specifies the expiration timestamp for an item'.

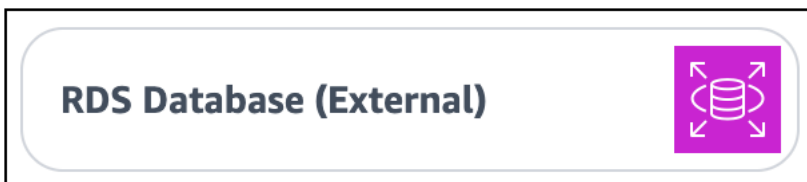
## Usando o Infrastructure Composer com o Amazon Relational Database Service (Amazon) RDS

AWS Infrastructure Composer apresenta uma integração com o Amazon Relational Database Service (RDSAmazon). Usando a placa de componente aprimorada do RDSbanco de dados (externo) no Infrastructure Composer, você pode conectar seu aplicativo à Amazon RDS DB clusters, instâncias e proxies definidos em outro modelo AWS CloudFormation ou AWS Serverless Application Model (AWS SAM).

O cartão de componente aprimorado do RDSbanco de dados (externo) representa RDS os recursos da Amazon que estão definidos em outro modelo. Isso inclui:

- Amazon RDS DB cluster ou instância definida em outro modelo
- Amazon RDS DB proxy

A placa de componente aprimorada RDSDatabase (External) está disponível na paleta Resources.



Para usar esse cartão, arraste-o para a tela do Infrastructure Composer, configure-o e conecte-o a outros recursos.

Você pode conectar seu aplicativo à Amazon externa RDS DB cluster ou instância por meio de uma função Lambda.

## Requisitos

Para usar esse recurso, você deve atender aos seguintes requisitos:

1. Sua Amazon externa RDS DB cluster, instância ou proxy devem ser usados AWS Secrets Manager para gerenciar a senha do usuário. Para saber mais, consulte [Gerenciamento de senhas com a Amazon RDS e AWS Secrets Manager](#) no Guia RDS do usuário da Amazon.
2. Seu aplicativo no Infrastructure Composer deve ser um novo projeto ou deve ter sido originalmente criado no Infrastructure Composer.

## Procedimento

Etapa 1: Configurar a placa de RDS banco de dados externa

Na paleta Resources, arraste um cartão de componente aprimorado do RDS banco de dados (externo) para a tela.

Selecione o cartão e escolha Detalhes ou clique duas vezes no cartão para abrir o painel de propriedades do recurso. O painel de propriedades do recurso do cartão aparecerá:



VPC
Details | Group | Delete

RDS Database (External)

**ExternalRDS**

## RDS Database (External)

RDS database cluster or instance defined outside of the template. This card will create 3 stack parameters by default. Specify values in this form or at deployment time. You can use “!ImportValue” or SSM with dynamic reference if value is stored elsewhere.

---

**Logical ID**

A unique name for your RDS database. This value will be used for environment variables and parameters in your template.

ExternalRDS

**Database Secret**

Secrets Manager secret to fetch database credentials. This field creates a stack parameter with name {Logical ID + SecretArn}.

**Database Hostname**

Hostname to connect to the RDS DB cluster or instance. For RDS Proxy, use the Proxy endpoint. This field creates a stack parameter with name {Logical ID + Hostname}.


**Database Port**

Port to connect to the RDS DB cluster or instance. This field creates a stack parameter with name {Logical ID + Port}.

Você pode configurar o seguinte aqui:

- ID lógica — Um nome exclusivo para sua Amazon externa RDS DB cluster, instância ou proxy. Essa ID não precisa corresponder ao valor lógico da sua Amazon externa RDS DB recurso.
- Segredo do banco de dados — Um identificador do AWS Secrets Manager segredo associado à sua Amazon RDS DB cluster, instância ou proxy. Esse campo aceita os seguintes valores:
  - Valor estático — Um identificador exclusivo do segredo do banco de dados, como o segredoARN. Veja um exemplo a seguir: `arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-name-1a2b3c`. Para obter mais informações, consulte [Conceitos do AWS Secrets Manager](#) no Manual do usuário do AWS Secrets Manager .

- Valor de saída — Quando um segredo do Secrets Manager é implantado AWS CloudFormation, um valor de saída é criado. Você pode especificar o valor de saída aqui usando a função [Fn::ImportValue](#) intrínseca. Por exemplo, `!ImportValue MySecret`.
- Valor do SSM Parameter Store — Você pode armazenar seu segredo no SSM Parameter Store e especificar seu valor usando uma referência dinâmica. Por exemplo, `{{resolve:ssm:MySecret}}`. Para obter mais informações, consulte [SSMos parâmetros](#) no Guia AWS CloudFormation do usuário.
- Nome do host do banco de dados — O nome do host que pode ser usado para se conectar à sua Amazon RDS DB cluster, instância ou proxy. Esse valor é especificado no modelo externo que define seu RDS recurso da Amazon. Os seguintes valores são aceitos:
  - Valor estático — Um identificador exclusivo do nome do host do banco de dados, como o endereço do endpoint. Veja um exemplo a seguir: `mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-2.rds.amazonaws.com`.
  - Valor de saída — O valor de saída de uma Amazon implantada RDS DB cluster, instância ou proxy. Você pode especificar o valor de saída usando a função [Fn::ImportValue](#) intrínseca. Por exemplo, `!ImportValue myStack-myDatabase-abcd1234`.
  - Valor do SSM Parameter Store — Você pode armazenar o nome do host do banco de dados no SSM Parameter Store e especificar seu valor usando uma referência dinâmica. Por exemplo, `{{resolve:ssm:MyDatabase}}`.
- Porta do banco de dados — O número da porta que pode ser usada para se conectar à sua Amazon RDS DB cluster, instância ou proxy. Esse valor é especificado no modelo externo que define seu RDS recurso da Amazon. Os seguintes valores são aceitos:
  - Valor estático — A porta do banco de dados. Por exemplo, `3306`.
  - Valor de saída — O valor de saída de uma Amazon implantada RDS DB cluster, instância ou proxy. Por exemplo, `!ImportValue myStack-MyRDSInstancePort`.
  - Valor do SSM Parameter Store — Você pode armazenar o nome do host do banco de dados no SSM Parameter Store e especificar seu valor usando uma referência dinâmica. Por exemplo, `{{resolve:ssm:MyRDSInstancePort}}`.

 Note

Somente o valor lógico da ID deve ser configurado aqui. Você pode configurar as outras propriedades no momento da implantação, se preferir.

## Etapa 2: Conectar uma placa de função Lambda

Na paleta Resources, arraste um cartão de componente aprimorado da Função Lambda para a tela.

Conecte a porta esquerda da placa Lambda Function à porta direita da placa RDSDatabase (externa).



O Infrastructure Composer provisionará seu modelo para facilitar essa conexão.

O que o Infrastructure Composer faz para criar sua conexão

Ao concluir o procedimento listado acima, o Infrastructure Composer executa ações específicas para conectar sua função Lambda ao seu banco de dados.

Ao especificar a Amazon externa RDS DB cluster, instância ou proxy

Quando você arrasta um cartão RDS de banco de dados (externo) para a tela, o Infrastructure Composer atualiza as Parameters seções Metadata e as seções do seu modelo conforme necessário. Veja um exemplo a seguir:

```
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
```

[Metadados](#) é uma seção de AWS CloudFormation modelo usada para armazenar detalhes sobre seu modelo. Os metadados específicos do Infrastructure Composer são armazenados sob a chave de

`AWS::Composer::ExternalResources` metadados. Aqui, o Infrastructure Composer armazena os valores que você especifica para sua Amazon RDS DB cluster, instância ou proxy.

A seção [Parâmetros](#) de um AWS CloudFormation modelo é usada para armazenar valores personalizados que podem ser inseridos em todo o modelo na implantação. Dependendo do tipo de valores que você fornece, o Infrastructure Composer pode armazenar valores aqui para sua Amazon RDS DB cluster, instância ou proxy e especifique-os em todo o seu modelo.

Os valores de string na `Parameters` seção `Metadata` e usam o valor de ID lógico que você especifica em seu cartão RDS de banco de dados (externo). Se você atualizar a ID lógica, os valores da string serão alterados.

Ao conectar a função Lambda ao seu banco de dados

Quando você conecta uma placa de função Lambda à placa `RDSDatabase` (externa), o Infrastructure Composer provisiona variáveis de ambiente e políticas AWS Identity and Access Management (IAM). Veja um exemplo a seguir:

```
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
```

Variáveis de [ambiente](#) são variáveis que podem ser usadas por sua função em tempo de execução. Para saber mais, consulte [Como usar variáveis de ambiente do Lambda](#) no Guia do AWS Lambda desenvolvedor.

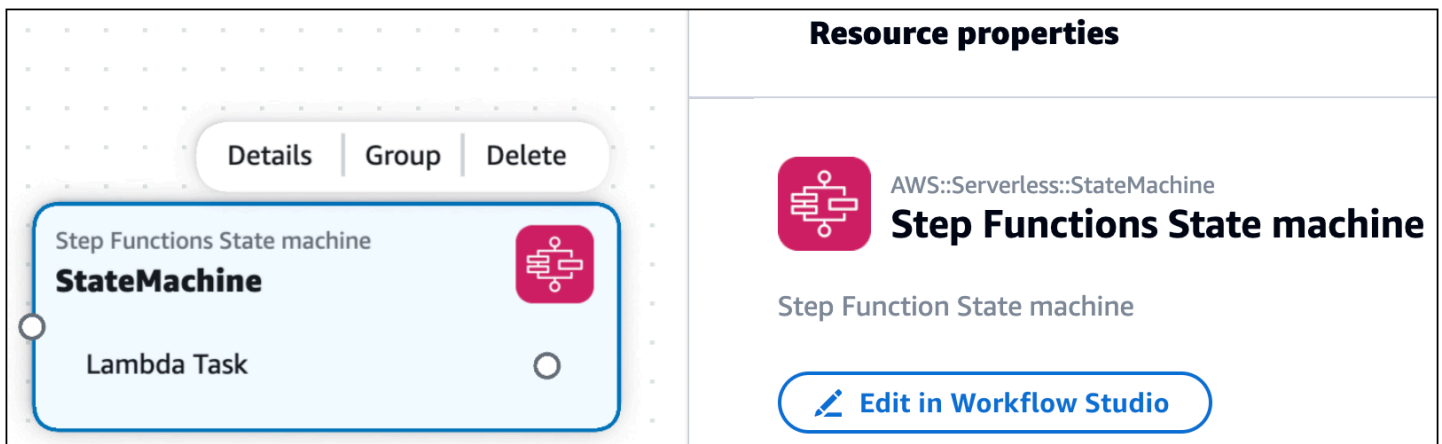
[As políticas](#) fornecem permissões para sua função. Aqui, o Infrastructure Composer cria uma política para permitir o acesso de leitura de sua função ao Secrets Manager para obter sua senha de acesso à Amazon RDS DB cluster, instância ou proxy.

## Usando AWS Infrastructure Composer com AWS Step Functions

AWS Infrastructure Composer apresenta uma integração com [AWS Step Functions Workflow Studio](#). Use o Infrastructure Composer para fazer o seguinte:

- Inicie Step Functions Workflow Studio diretamente no Infrastructure Composer.
- Crie e gerencie novos fluxos de trabalho ou importe fluxos de trabalho existentes para o Infrastructure Composer.
- Integre seus fluxos de trabalho com outros AWS recursos usando a tela do Infrastructure Composer.

A imagem a seguir é de uma placa de máquina Step Functions State



Com Step Functions Workflow Studio no Infrastructure Composer, você pode usar os benefícios de dois designers visuais poderosos em um único lugar. Conforme você projeta seu fluxo de trabalho e aplicativo, o Infrastructure Composer cria sua infraestrutura como código (IaC) para orientá-lo na implantação.

### Tópicos

- [Políticas do IAM](#)
- [Introdução ao Step Functions Workflow Studio no Infrastructure Composer](#)
- [Usando Step Functions Workflow Studio no Infrastructure Composer](#)
- [Saiba mais](#)

## Políticas do IAM

Quando você conecta tarefas do seu fluxo de trabalho aos recursos, o Infrastructure Composer cria automaticamente as políticas AWS Identity and Access Management (IAM) necessárias para autorizar a interação entre seus recursos. Veja um exemplo a seguir:

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

Se necessário, você pode adicionar mais IAM políticas ao seu modelo.

## Introdução ao Step Functions Workflow Studio no Infrastructure Composer

Para começar, você pode criar novos fluxos de trabalho ou importar fluxos de trabalho existentes.

Para criar um novo fluxo de trabalho

1. Na paleta Resources, arraste um cartão de componente aprimorado da máquina Step Functions State para a tela.




Quando você arrasta um cartão de máquina Step Functions State para a tela, o Infrastructure Composer cria o seguinte:

- Um [AWS::Serverless::StateMachine](#) recurso que define sua máquina de estados. Por padrão, o Infrastructure Composer cria um fluxo de trabalho padrão. Para criar um fluxo de trabalho expresso, altere o Type valor em seu modelo de STANDARD para EXPRESS.

- Um [AWS::Logs::LogGroup](#) recurso que define um grupo de CloudWatch registros da Amazon para sua máquina de estado.
2. Abra o painel de propriedades do recurso do cartão e selecione Editar no Workflow Studio para abrir Workflow Studio dentro do Infrastructure Composer.

Step Functions Workflow Studio abre no modo Design. Para saber mais, consulte [Modo de design](#) no Guia do AWS Step Functions desenvolvedor.

 Note

Você pode modificar o Infrastructure Composer para salvar sua definição de máquina de estado em um arquivo externo. Para saber mais, consulte [Trabalhando com arquivos externos](#).

3. Crie seu fluxo de trabalho e escolha Salvar. Para sair Workflow Studio, escolha Retornar ao Infrastructure Composer.

O Infrastructure Composer define seu fluxo de trabalho usando a Defintion propriedade do `AWS::Serverless::StateMachine` recurso.

4. Você pode modificar seu fluxo de trabalho fazendo qualquer um dos seguintes:
  - Abra o Workflow Studio novamente e modifique seu fluxo de trabalho.
  - Para o Infrastructure Composer a partir do console, você pode abrir a visualização Modelo do seu aplicativo e modificar seu modelo. Se estiver usando a sincronização local, você pode modificar seu fluxo de trabalho em seu localIDE. O Infrastructure Composer detectará suas alterações e atualizará seu fluxo de trabalho no Infrastructure Composer.
  - Para o Infrastructure Composer do Toolkit for VS Code, você pode modificar diretamente seu modelo. O Infrastructure Composer detectará suas alterações e atualizará seu fluxo de trabalho no Infrastructure Composer.

## Para importar fluxos de trabalho existentes

Você pode importar fluxos de trabalho de aplicativos definidos usando modelos AWS Serverless Application Model (AWS SAM). Use qualquer máquina de estado definida com o tipo de `AWS::Serverless::StateMachine` recurso e ela será visualizada como uma placa de componente aprimorada da máquina Step Functions State que você pode usar para iniciar Workflow Studio.

O `AWS::Serverless::StateMachine` recurso pode definir fluxos de trabalho usando uma das seguintes propriedades:

- [Definition](#)— O fluxo de trabalho é definido no AWS SAM modelo como um objeto.
- [DefinitionUri](#)— O fluxo de trabalho é definido em um arquivo externo usando o [Amazon States Language](#). O caminho local do arquivo é então especificado com essa propriedade.

## Propriedade de definição

### Infrastructure Composer a partir do console

Para fluxos de trabalho definidos usando a `Definition` propriedade, você pode importar um único modelo ou o projeto inteiro.

- Modelo — Para obter instruções sobre como importar um modelo, consulte [Importar um modelo de projeto existente no console do Infrastructure Composer](#). Para salvar as alterações feitas no Infrastructure Composer, você deve exportar seu modelo.
- Projeto — Ao importar um projeto, você deve ativar a sincronização local. As alterações que você faz são salvas automaticamente em sua máquina local. Para obter instruções sobre como importar um projeto, consulte [Importar uma pasta de projeto existente no console do Infrastructure Composer](#).

### Compositor de infraestrutura do Toolkit for VS Code

Para fluxos de trabalho definidos usando a `Definition` propriedade, você pode abrir o Infrastructure Composer a partir do seu modelo. Para obter instruções, consulte [Acesse o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code](#).

## DefinitionUri propriedade

### Infrastructure Composer a partir do console

Para fluxos de trabalho definidos usando a `DefinitionUri` propriedade, você deve importar o projeto e ativar a sincronização local. Para obter instruções sobre como importar um projeto, consulte [Importar uma pasta de projeto existente no console do Infrastructure Composer](#).

### Compositor de infraestrutura do Toolkit for VS Code

Para fluxos de trabalho definidos usando a `DefinitionUri` propriedade, você pode abrir o Infrastructure Composer a partir do seu modelo. Para obter instruções, consulte [Acesse o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code](#).



## Usando Step Functions Workflow Studio no Infrastructure Composer

### Crie fluxos de trabalho

O Infrastructure Composer usa substituições de definição para mapear tarefas de fluxo de trabalho para recursos em seu aplicativo. Para saber mais sobre substituições de definições, consulte [DefinitionSubstitutions](#) o Guia do AWS Serverless Application Model desenvolvedor.

Quando você cria tarefas no Workflow Studio, especifique uma substituição de definição para cada tarefa. Em seguida, você pode conectar tarefas a recursos na tela do Infrastructure Composer.

Para especificar uma substituição de definição em Workflow Studio

1. Abra a guia Configuração da tarefa e localize o campo APIParâmetros.

2. Se o campo APIParâmetros tiver uma opção suspensa, escolha Inserir uma AWS CloudFormation substituição. Em seguida, forneça um nome exclusivo.

Para tarefas que se conectam ao mesmo recurso, especifique a mesma substituição de definição para cada tarefa. Para usar uma substituição de definição existente, escolha Selecionar uma AWS CloudFormation substituição e selecione a substituição a ser usada.

- Se o campo `APIParâmetros` contiver um JSON objeto, modifique a entrada que especifica o nome do recurso para usar uma substituição de definição. No exemplo a seguir, mudamos `"MyDynamoDBTable"` para `"${RecordTransaction}"`.

The screenshot displays the AWS Step Functions console. On the left, a workflow diagram shows a sequence of tasks: 'Start' leads to 'Lambda: Invoke Check Stock Value', which then leads to a 'Choice state Choice'. The choice state has two paths: one for the condition `$.stock_price <= 50` leading to 'Lambda: Invoke Buy Stock', and a 'Default' path leading to 'Lambda: Invoke Sell Stock'. Both paths converge into a 'DynamoDB: PutItem Record Transaction' task, which finally leads to 'End'.

On the right, the configuration panel for the 'Record Transaction' state is shown. It includes the following details:

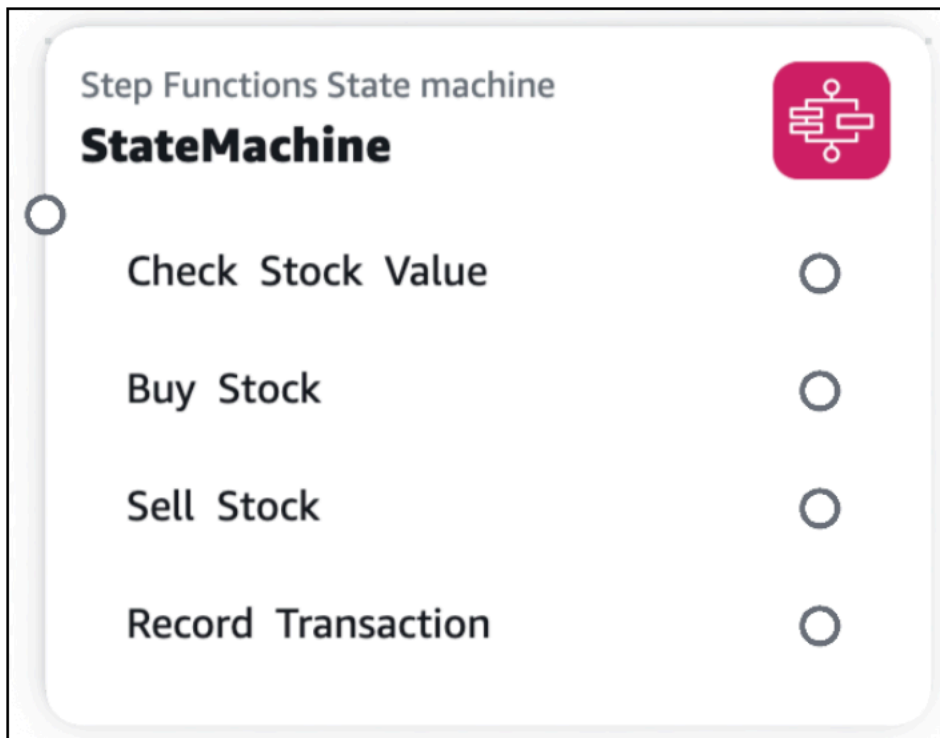
- State name:** Record Transaction
- API:** DynamoDB: PutItem
- Integration type:** Optimized
- API Parameters:** A JSON object defining the table name and item structure:
 

```
1 {
2   "TableName": "${RecordTransaction}",
3   "Item": {
4     "Column": {
5       "S": "MyEntry"
6     }
7   }
8 }
```

Below the JSON editor, a note states: "Must be valid JSON. To reference a node in this state's JSON input, the key must end with `$.` (for example `"key2.$": "$inputValue"`). [Info](#)

- Selecione Salvar e retornar ao Infrastructure Composer.

As tarefas do seu fluxo de trabalho serão visualizadas na placa da máquina Step Functions State.



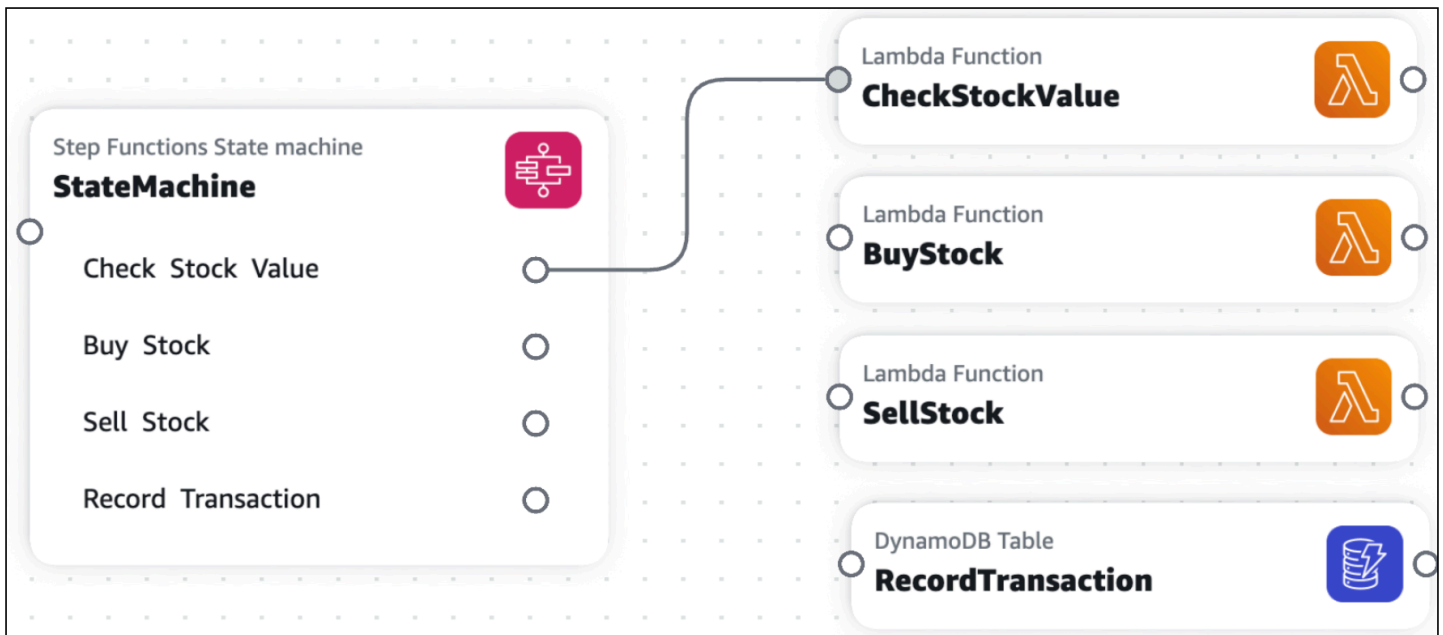
Conecte recursos às tarefas do fluxo de trabalho

Você pode criar conexões no Infrastructure Composer entre tarefas de fluxo de trabalho suportadas e placas compatíveis do Infrastructure Composer.

- Tarefas de fluxo de trabalho suportadas — As tarefas para Serviços da AWS isso são otimizadas para Step Functions. Para saber mais, consulte [Integrações otimizadas para Step Functions](#) no Guia do AWS Step Functions desenvolvedor.
- Placas Infrastructure Composer suportadas — Placas de componentes aprimoradas são suportadas. Para saber mais sobre cartões no Infrastructure Composer, consulte [Configurar e modificar cartões no Infrastructure Composer](#).

Ao criar uma conexão, a AWS service (Serviço da AWS) tarefa e o cartão devem corresponder. Por exemplo, você pode conectar uma tarefa de fluxo de trabalho que invoca uma função Lambda a um cartão de componente aprimorado da Função Lambda.

Para criar uma conexão, clique e arraste a porta de uma tarefa para a porta esquerda de uma placa de componente aprimorada.



O Infrastructure Composer atualizará automaticamente seu `DefinitionSubstitution` valor para definir sua conexão. Veja um exemplo a seguir:

```

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:
      ...

```

## Trabalhando com arquivos externos

Quando você cria um fluxo de trabalho a partir do cartão de máquina Step Functions State, o Infrastructure Composer salva sua definição de máquina de estado em seu modelo usando a `Definition` propriedade. Você pode configurar o Infrastructure Composer para salvar sua definição de máquina de estado em um arquivo externo.

### Note

Para usar esse recurso com o Infrastructure Composer do AWS Management Console, você deve ter a sincronização local ativada. Para obter mais informações, consulte [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#).

Para salvar sua definição de máquina de estado em um arquivo externo

1. Abra o painel de propriedades do recurso da sua placa de máquina Step Functions State.
2. Selecione a opção Usar arquivo externo para definição de máquina de estado.
3. Forneça um caminho e um nome relativos para o arquivo de definição da máquina de estado.
4. Escolha Salvar.

O Infrastructure Composer fará o seguinte:

1. Mova a definição da máquina de estado do `Definition` campo para o arquivo externo.
2. Salve sua definição de máquina de estado em um arquivo externo usando a Amazon States Language.
3. Modifique seu modelo para referenciar o arquivo externo usando o `DefinitionUri` campo.

Saiba mais

Para saber mais sobre Step Functions no Infrastructure Composer, veja o seguinte:

- [Usando Workflow Studio no Infrastructure Composer](#) no Guia do AWS Step Functions desenvolvedor.
- [DefinitionSubstitutions em AWS SAM modelos](#) no Guia do AWS Step Functions desenvolvedor.

## Cartões padrão no Infrastructure Composer

Todos os AWS CloudFormation recursos estão disponíveis para uso como cartões de recursos padrão do IaC na paleta Recursos. Depois de ser arrastado para a tela visual, um cartão de recursos padrão do IaC se torna um cartão de componente padrão. Isso significa simplesmente que a placa é um ou mais recursos padrão de IaC. Para obter mais exemplos e detalhes, consulte os tópicos desta seção.

Você pode modificar seu código de infraestrutura por meio da visualização Modelo e da janela Propriedades do recurso. Por exemplo, a seguir está um exemplo de modelo inicial de um recurso Alexa::ASK::Skill padrão de IaC:

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

Um modelo inicial de cartão de recursos IaC padrão consiste no seguinte:

- O tipo de AWS CloudFormation recurso.
- Propriedades necessárias ou comumente usadas.
- O tipo necessário do valor a ser fornecido para cada propriedade.

### Note

Você pode usar: Amazon Q para gerar sugestões de código de infraestrutura para cartões de recursos padrão. Para saber mais, consulte [Usando AWS Infrastructure Composer com Amazon Q Developer](#).

## Procedimento

Você pode modificar o código de infraestrutura de cada recurso em uma placa de componente padrão por meio do painel Propriedades do recurso.

Para modificar uma placa de componente padrão

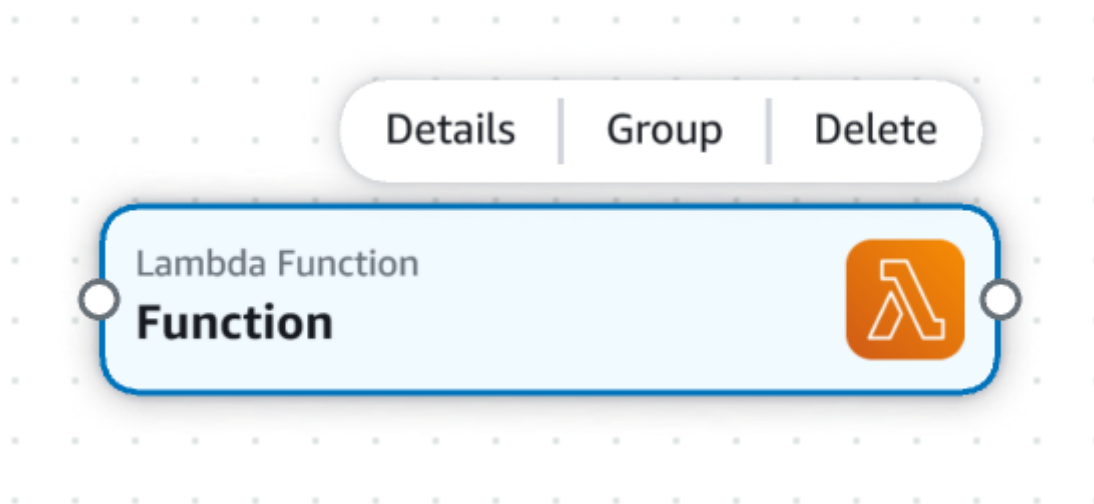
1. Abra o painel de propriedades do recurso da placa componente padrão do IaC.
2. No campo Edição, selecione o recurso padrão do IaC a ser editado na lista suspensa.
3. Modifique seu código de infraestrutura e salve.

## Excluir cartões no Infrastructure Composer

Esta seção fornece instruções para excluir cartões em AWS Infrastructure Composer.

### Placas de componentes aprimoradas

Para excluir um cartão de componente aprimorado, selecione um cartão que você tenha colocado na tela visual. No menu Ações do cartão, selecione Excluir.



### Placas de componentes padrão

Para excluir placas de componentes padrão, você deve remover manualmente o código de infraestrutura de cada AWS CloudFormation recurso do seu modelo. A seguir está uma maneira simples de fazer isso:

1. Anote a ID lógica do recurso a ser excluído.

2. Em seu modelo, localize o recurso por sua ID lógica na Outputs seção Resources ou.
3. Exclua o recurso do seu modelo. Isso inclui a ID lógica do recurso e seus valores aninhados, como Type e Properties
4. Verifique a visualização do Canvas para verificar se o recurso foi removido da sua tela.

## Veja as atualizações de código com o Change Inspector no Infrastructure Composer

Conforme você projeta no console do Infrastructure Composer, seu código de infraestrutura é criado automaticamente. Use o Change Inspector para ver as atualizações do código do seu modelo e saber o que o Infrastructure Composer está criando para você.

Este tópico aborda o uso do Infrastructure Composer da AWS Management Console ou da AWS Toolkit for Visual Studio Code extensão.

O Change Inspector é uma ferramenta visual dentro do Infrastructure Composer que mostra atualizações recentes de código.

- Conforme você projeta seu aplicativo, as mensagens são exibidas na parte inferior da tela visual. Essas mensagens fornecem comentários sobre as ações que você está realizando.
- Quando suportado, você pode expandir uma mensagem para ver o Inspector de Alterações.
- O Change Inspector exibe as alterações de código da sua interação mais recente.

O exemplo a seguir demonstra como o inspetor de mudanças funciona:



The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' list with various AWS services. The main canvas displays a 'Lambda Function' resource connected to an 'S3 Bucket' resource. Below the canvas, the 'Change Inspector' panel is open, showing the code for the connection. The code is as follows:

```

15 + Environment:
16 +   Variables:
17 +     BUCKET_BUCKET_NAME: !Ref Bucket
18 +     BUCKET_BUCKET_ARN: !GetAtt Bucket.Arn
19 +   Policies:
20 +     - Statement:
21 +       - Effect: Allow
22 +         Action:
23 +           - s3:GetObject
24 +           - s3:GetObjectAcl
25 +           - s3:GetObjectLegalHold
26 +           - s3:GetObjectRetention
27 +

```

On the right side, the 'Resource properties' panel for the 'S3 Bucket' is visible, showing options like 'Stores files', 'Logical ID', 'Activate static website hosting', and 'Block Public Access'.

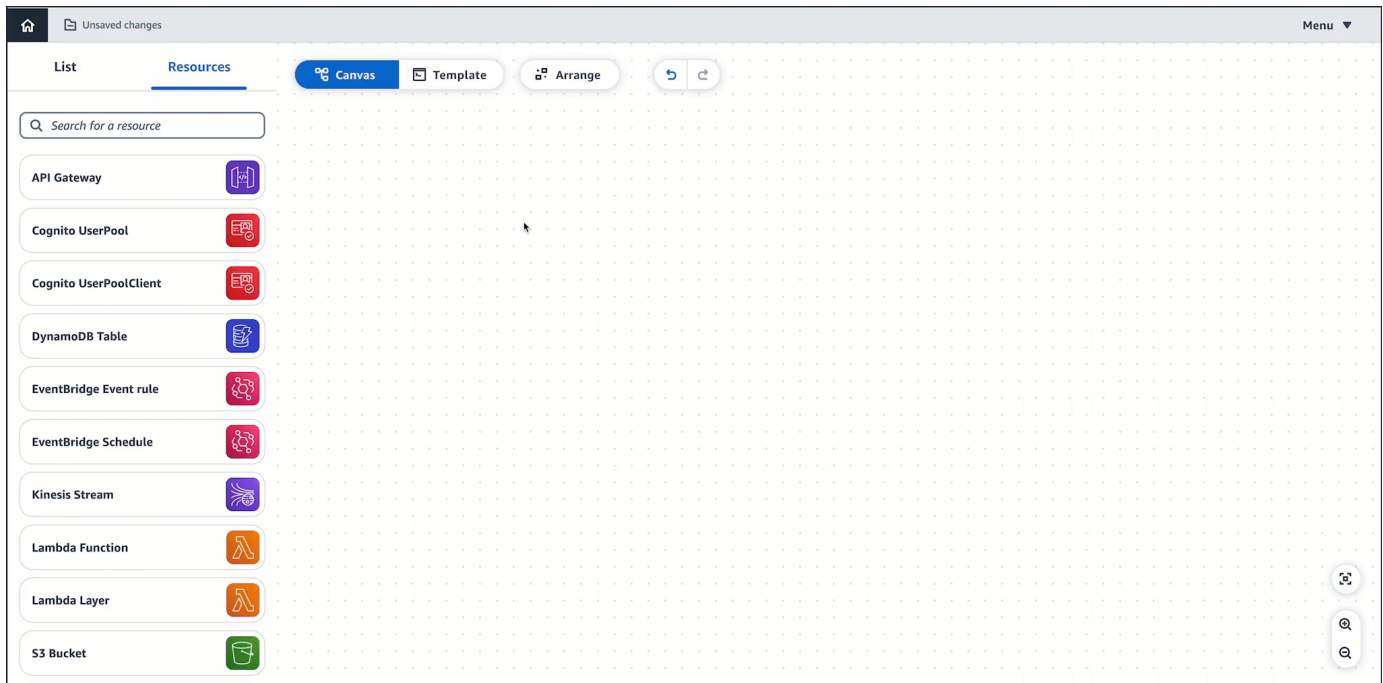
## Benefícios do Change Inspector

O Change Inspector é uma ótima maneira de visualizar o código de modelo que o Infrastructure Composer cria para você. Também é uma ótima maneira de aprender a escrever código de infraestrutura. Ao projetar aplicativos no Infrastructure Composer, veja as atualizações de código no Change Inspector para saber mais sobre o código necessário para provisionar seu design.

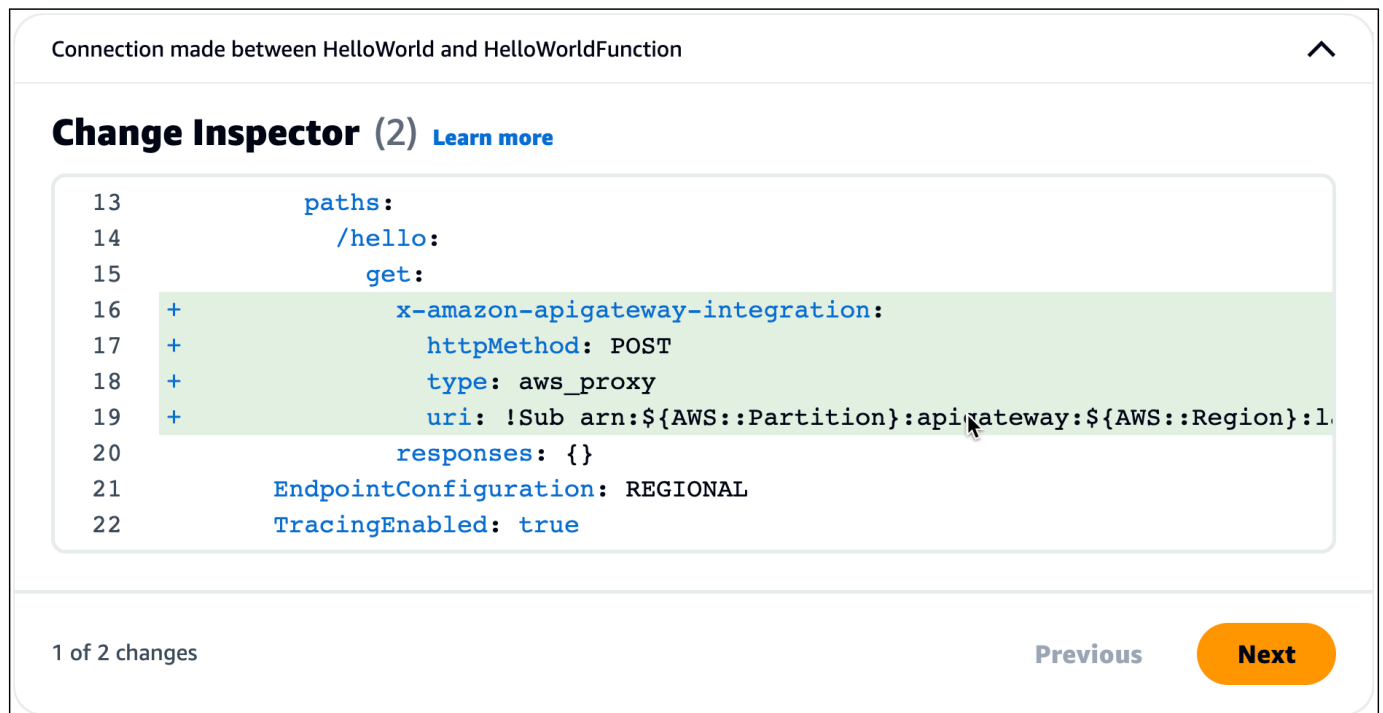
## Procedimento

Para usar o Change Inspector

1. Expanda uma mensagem para abrir o Change Inspector.



2. Veja o código que foi composto automaticamente para você.



- O código destacado em verde indica o código recém-adicionado.
- O código destacado em vermelho indica um código recém-removido.
- Os números das linhas indicam a localização em seu modelo.

- Quando várias secções do seu modelo são atualizadas, o Inspector de Alterações as organiza. Selecione os botões Anterior e Avançar para ver todas as alterações.



### Note

Para o Infrastructure Composer a partir do console, você pode visualizar as alterações de código no contexto de todo o seu modelo, usando a Visualização de modelo. Você também pode sincronizar o Infrastructure Composer com um local IDE e visualizar todo o seu modelo em sua máquina local. Para saber mais, consulte [Conecte o console do Infrastructure Composer ao seu console local IDE](#).

## Saiba mais

Para obter mais informações sobre o código que o Infrastructure Composer cria, consulte o seguinte:

- [Conexões de cartão no Infrastructure Composer](#).

# Faça referência a arquivos externos no Infrastructure Composer

Você pode usar arquivos externos com seus modelos AWS Serverless Application Model (AWS SAM) para reutilizar códigos repetidos e organizar seus projetos. Por exemplo, você pode ter vários REST API recursos do Amazon API Gateway descritos por um OpenAPI especificação. Em vez de replicar o OpenAPI código de especificação em seu modelo, você pode criar um arquivo externo e referenciá-lo para cada um dos seus recursos.

AWS Infrastructure Composer suporta os seguintes casos de uso de arquivos externos:

- API Gateway REST API recursos definidos por recursos externos OpenAPI arquivos de especificação.
- AWS Step Functions recursos de máquina de estado definidos por arquivos externos de definição de máquina de estado.

Para saber mais sobre como configurar arquivos externos para recursos compatíveis, consulte o seguinte:

- [DefinitionBody](#) para `AWS::Serverless::Api`.
- [DefinitionUri](#) para `AWS::Serverless::StateMachine`.

## Note

Para referenciar arquivos externos com o Infrastructure Composer a partir do console do Infrastructure Composer, você deve usar o Infrastructure Composer no modo de sincronização local. Para obter mais informações, consulte [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#).

## Tópicos

- [Melhores práticas para arquivos de referência externos do Infrastructure Composer](#)
- [Crie uma referência de arquivo externo no Infrastructure Composer](#)
- [Carregar um projeto com uma referência de arquivo externo no Infrastructure Composer](#)
- [Crie um aplicativo que faça referência a um arquivo externo no Infrastructure Composer](#)
- [Referenciar um OpenAPI arquivo externo de especificação com Infrastructure Composer](#)

# Melhores práticas para arquivos de referência externos do Infrastructure Composer

## Use o Infrastructure Composer com um local IDE

Ao usar o Infrastructure Composer com um local IDE no modo de sincronização local, você pode usar seu local IDE para visualizar e modificar arquivos externos. O conteúdo dos arquivos externos compatíveis que são referenciados em seu modelo será atualizado automaticamente na tela do Infrastructure Composer. Para saber mais, consulte [Conecte o console do Infrastructure Composer ao seu console local IDE](#).

## Mantenha os arquivos externos no diretório principal do seu projeto

Você pode criar subdiretórios no diretório principal do seu projeto para organizar seus arquivos externos. O Infrastructure Composer não pode acessar arquivos externos armazenados em um diretório fora do diretório principal do seu projeto.

## Implante seu aplicativo usando o AWS SAM CLI

Ao implantar seu aplicativo no Nuvem AWS, os arquivos externos locais precisam primeiro ser carregados em um local acessível, como o Amazon Simple Storage Service (Amazon S3). Você pode usar o AWS SAM CLI para facilitar automaticamente esse processo. Para saber mais, consulte [Carregar arquivos locais na implantação](#) no Guia do AWS Serverless Application Model desenvolvedor.

## Crie uma referência de arquivo externo no Infrastructure Composer

Você pode criar uma referência de arquivo externo no painel de propriedades do recurso dos recursos suportados.

Para criar uma referência de arquivo externo

1. Em uma placa de componente aprimorada do APIGateway ou do Step Functions, selecione Detalhes para abrir o painel de propriedades do recurso.
2. Localize e selecione a opção Usar arquivo externo.
3. Especifique o caminho relativo para o arquivo externo. Esse é o caminho do seu `template.yaml` arquivo para o arquivo externo.

Por exemplo, para referenciar o arquivo `api-spec.yaml` externo a partir da estrutura do projeto a seguir, especifique `./api-spec.yaml` como seu caminho relativo.

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

#### Note

Se o arquivo externo e seu caminho especificado não existirem, o Infrastructure Composer o criará.

4. Salve suas alterações.

## Carregar um projeto com uma referência de arquivo externo no Infrastructure Composer

Siga as etapas listadas nesta página para carregar um projeto do Infrastructure Composer com uma referência de arquivo externo.

Do console do Infrastructure Composer

1. Siga as etapas listadas em [Importar um modelo de projeto existente no console do Infrastructure Composer](#).
2. Confirme se o Infrastructure Composer solicita que você se conecte à pasta raiz do seu projeto

Se o seu navegador suportar o acesso ao sistema de arquivosAPI, o Infrastructure Composer solicitará que você se conecte à pasta raiz do seu projeto. O Infrastructure Composer abrirá seu projeto no modo de sincronização local para suportar seu arquivo externo. Se o arquivo externo referenciado não for suportado, você receberá uma mensagem de erro. Para obter mais informações sobre mensagens de erro, consulte [Solução de problemas](#).

## Do Toolkit for VS Code

1. Siga as etapas listadas em [Acesse o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code](#).
2. Abra o modelo que você deseja visualizar no Infrastructure Composer.

Quando você acessa o Infrastructure Composer a partir de um modelo, o Infrastructure Composer detectará automaticamente seu arquivo externo. Se o arquivo externo referenciado não for suportado, você receberá uma mensagem de erro. Para obter mais informações sobre mensagens de erro, consulte [Solução de problemas](#).

## Crie um aplicativo que faça referência a um arquivo externo no Infrastructure Composer

Este exemplo usa o AWS SAM CLI para criar um aplicativo que faça referência a um arquivo externo para sua definição de máquina de estado. Em seguida, você carrega seu projeto no Infrastructure Composer com seu arquivo externo devidamente referenciado.

### Exemplo

1. Primeiro, use o AWS SAM CLI `sam init` comando para inicializar um novo aplicativo chamado `demo`. Durante o fluxo interativo, selecione o modelo de início rápido do fluxo de trabalho em várias etapas.

```
$ 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
```

```
...
```

Esse aplicativo faz referência a um arquivo externo para a definição da máquina de estado.

```
...
```



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

O arquivo externo está localizado no statemachine subdiretório do nosso aplicativo.

```
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. Em seguida, carregue seu aplicativo no Infrastructure Composer a partir do console. Na página inicial do Infrastructure Composer, selecione Carregar um CloudFormation modelo.
3. Selecione a pasta demo do nosso projeto e permita que o prompt visualize os arquivos. Selecione nosso `template.yaml` arquivo e selecione Criar. Quando solicitado, selecione Salvar alterações.

## Open project folder ✕

**Project location**  
Select the folder that contains your existing project.

📁 Select folder

✔️ demo

**Template file**  
We will use the project location to automatically detect a template file. If you have multiple files in the folder, select from the dropdown. A copy of your template file will be stored in a folder named `.aws-composer` at the root of your project location.

template.yaml ▾

Cancel
Create

O Infrastructure Composer detecta automaticamente o arquivo de definição da máquina de estado externa e o carrega. Selecione nosso `StockTradingStateMachine` recurso e escolha Detalhes para mostrar o painel de propriedades do recurso. Aqui, você pode ver que o Infrastructure Composer se conectou automaticamente ao nosso arquivo externo de definição de máquina de estado.

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' list with various AWS services. The main area is a canvas showing a state machine diagram with four tasks: 'Check Stock Value', 'Sell Stock', 'Buy Stock', and 'Record Transaction'. The 'StockTradingStateMachine' resource is selected, and its details panel is open on the right. The details panel shows the state machine definition and a checkbox for 'Use external file for state machine definition' which is checked.

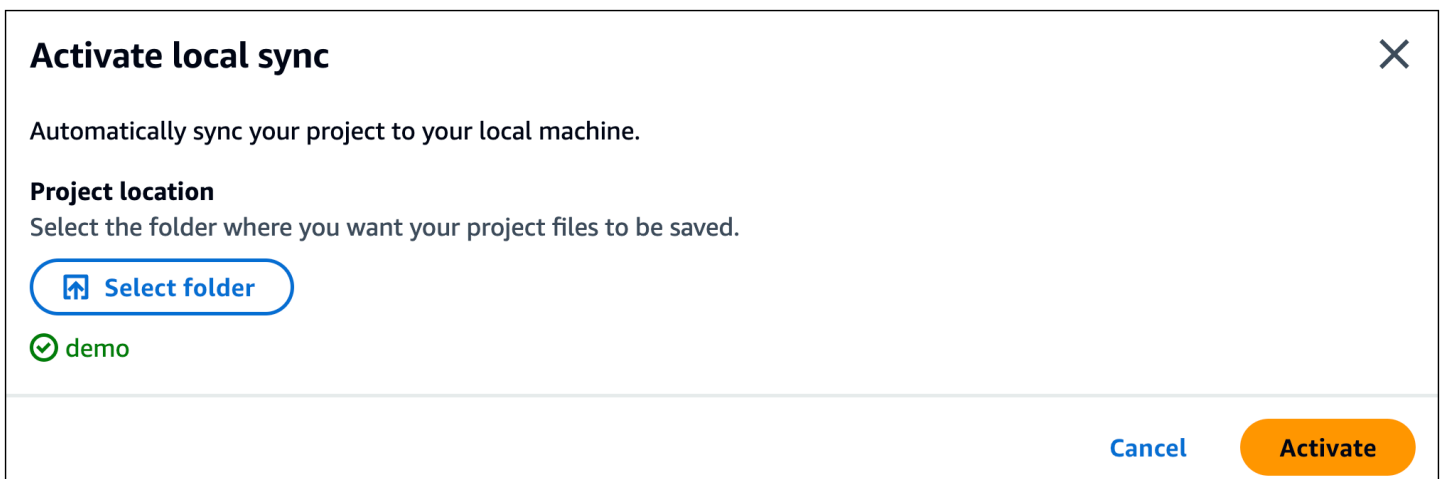
Quaisquer alterações feitas no arquivo de definição da máquina de estado serão refletidas automaticamente no Infrastructure Composer.

## Referenciar um OpenAPI arquivo externo de especificação com Infrastructure Composer

Este exemplo usa o Infrastructure Composer do console para referenciar um externo OpenAPI arquivo de especificação que define um API Gateway REST API.

Primeiro, crie um novo projeto na página inicial do Infrastructure Composer.

Em seguida, ative a sincronização local selecionando Ativar sincronização local no Menu. Crie uma nova pasta chamada demo, permita que o prompt visualize os arquivos e selecione Ativar. Quando solicitado, selecione Salvar alterações.



Em seguida, arraste um cartão Amazon API Gateway para a tela. Selecione Detalhes para abrir o painel de propriedades do recurso.

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' panel with a search bar and a list of AWS services including API Gateway, Cognito UserPool, Cognito UserPoolClient, DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, Lambda Function, Lambda Layer, and S3 Bucket. The main canvas displays a resource named 'Api' of type 'API Gateway' with a 'GET /' endpoint. Above the resource are 'Details', 'Group', and 'Delete' buttons. On the right, the 'Resource properties' panel is open, showing the 'API Gateway' resource type and its description: 'Creates an HTTP endpoint to route API requests and responses'. The 'Logical ID' is set to 'Api'. Under 'Authorizers', it states 'No authorizers associated with the resource.' and provides an 'Add authorizer' button. The 'Default authorizer' is set to 'None'. Under 'Routes', the 'Method' is set to 'GET' and the 'Path' is empty.

No painel Propriedades do recurso, configure o seguinte e salve.

- Selecione a opção Usar arquivo externo para definição de API.
- Entrada `./api-spec.yaml` como caminho relativo para o arquivo externo

## Use external file for api definition



## Relative path to external file

`./api-spec.yaml`

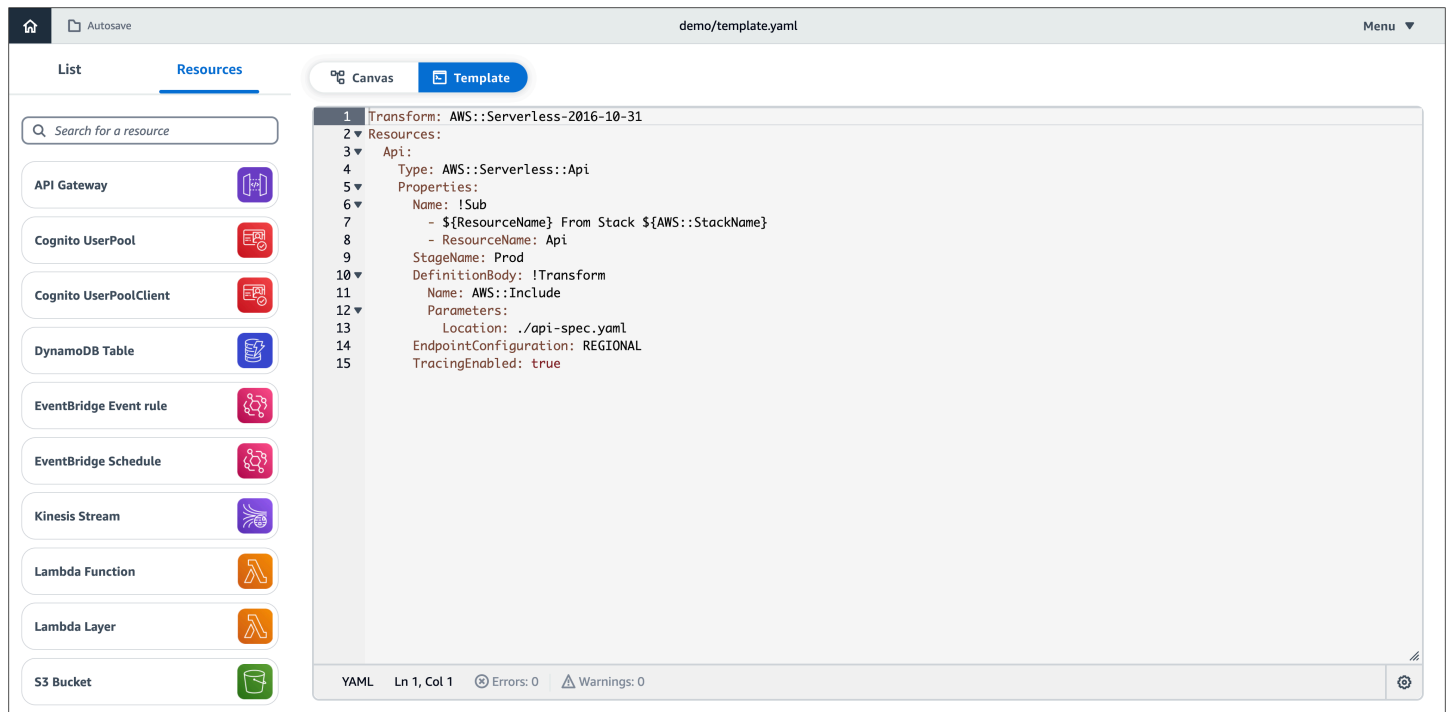
Isso cria o seguinte diretório em nossa máquina local:

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

Agora, você pode configurar o arquivo externo em nossa máquina local. Usando nosso IDE, abra o `api-spec.yaml` localizado na pasta do seu projeto. Substitua seu conteúdo pelo seguinte:

```
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
```

Na visualização Modelo do Infrastructure Composer, você pode ver que o Infrastructure Composer atualizou automaticamente seu modelo para referenciar o arquivo externo.

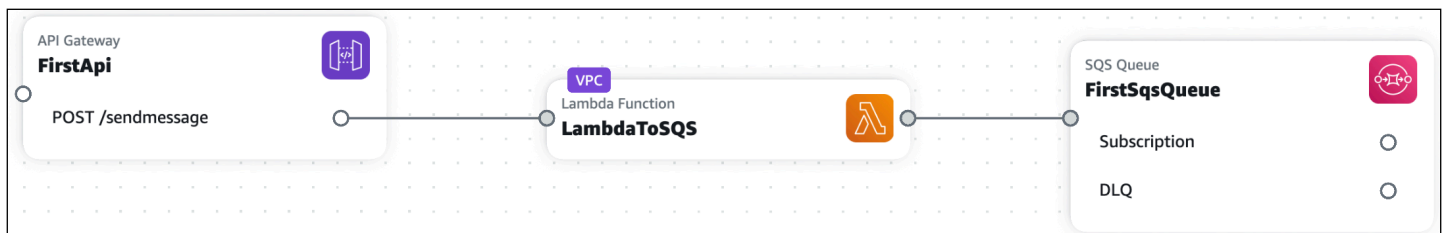


## Integre o Infrastructure Composer com a Amazon Virtual Private Cloud (AmazonVPC)

AWS Infrastructure Composer apresenta uma integração com o serviço Amazon Virtual Private Cloud (AmazonVPC). Usando o Infrastructure Composer, você pode fazer o seguinte:

- Identifique os recursos em sua tela que estão em uma VPC VPCtag visual.
- Configure AWS Lambda funções a VPCs partir de um modelo externo.

A imagem a seguir mostra um exemplo de um aplicativo com uma função Lambda configurada com um VPC



Para saber mais sobre a AmazonVPC, consulte [O que é a AmazonVPC?](#) no Guia do VPC usuário da Amazon.

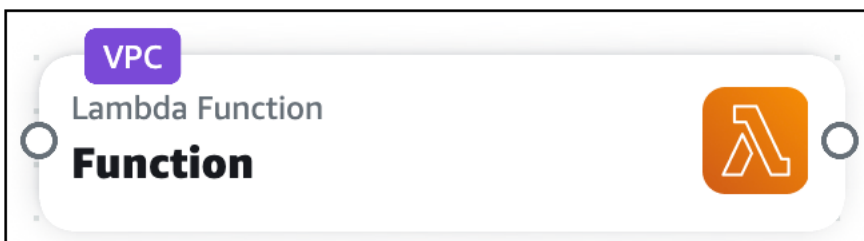
## Tópicos

- [Identifique os recursos do Infrastructure Composer e as informações relacionadas em um VPC](#)
- [Configure funções Lambda com funções externas VPCs no Infrastructure Composer](#)
- [Parâmetros em modelos importados para um externo VPC com o Infrastructure Composer](#)
- [Adicionando novos parâmetros aos modelos importados com o Infrastructure Composer](#)
- [Configure uma função Lambda e uma VPC definida em outro modelo com o Infrastructure Composer](#)

## Identifique os recursos do Infrastructure Composer e as informações relacionadas em um VPC

Para integrar o Infrastructure Composer com a AmazonVPC, você deve primeiro identificar os recursos em um VPC e as informações necessárias para concluir uma integração. Isso também inclui informações de configuração relacionadas a grupos de segurança, identificadores de sub-rede, tipos de parâmetros, tipos e SSM tipos de valores estáticos.

O Infrastructure Composer visualiza recursos em um VPC usando uma VPCtag. Essa etiqueta é aplicada aos cartões na tela. Veja a seguir um exemplo de uma função Lambda com uma VPC tag:



VPCas etiquetas são aplicadas aos cartões na tela quando você faz o seguinte:

- Configure uma função Lambda com um Infrastructure VPC Composer.
- Importe um modelo que contenha recursos configurados com umVPC.

## Identificadores de grupo de segurança e sub-rede

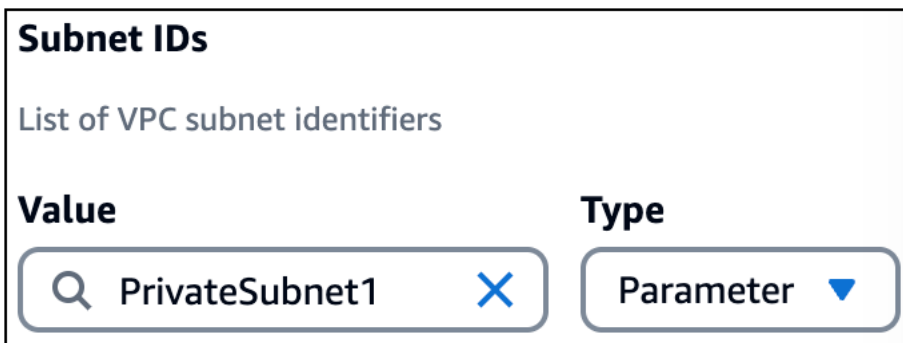
Uma função Lambda pode ser configurada com vários grupos de segurança e sub-redes. Para configurar um grupo de segurança ou sub-rede para uma função Lambda, forneça um valor e um tipo.

- Valor — Um identificador para o grupo de segurança ou sub-rede. Os valores aceitos variam de acordo com o tipo.
- Tipo — Os seguintes tipos de valores são permitidos:
  - Nome do parâmetro
  - AWS Systems Manager (SSM) Armazenamento de parâmetros
  - Valor estático

## Tipo de parâmetro

A `Parameters` seção de um AWS CloudFormation modelo pode ser usada para armazenar informações de recursos em vários modelos. Para obter mais informações sobre parâmetros, consulte [Parâmetros](#) no Guia AWS CloudFormation do usuário.

Para o tipo de parâmetro, você pode fornecer um nome de parâmetro. No exemplo a seguir, fornecemos um valor de nome de `PrivateSubnet1` parâmetro:



**Subnet IDs**

List of VPC subnet identifiers

**Value** **Type**

PrivateSubnet1 × Parameter ▼

Quando você fornece um nome de parâmetro, o Infrastructure Composer o define na `Parameters` seção do seu modelo. Em seguida, o Infrastructure Composer faz referência ao parâmetro em seu recurso de função Lambda. Veja um exemplo a seguir:

```
...
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
```

## Tipo de SSM

O SSM Parameter Store fornece um armazenamento seguro e hierárquico para gerenciamento de dados de configuração e gerenciamento de segredos. Para obter mais informações, consulte o [Armazenamento de parâmetros do AWS Systems Manager](#), no Guia do usuário do AWS Systems Manager .

Para o SSMtipo, você pode fornecer os seguintes valores:

- Referência dinâmica a um valor do SSM Parameter Store.
- ID lógica de um `AWS::SSM::Parameter` recurso definido em seu modelo.

### Referência dinâmica

Você pode referenciar um valor do SSM Parameter Store usando uma referência dinâmica no seguinte formato: `{{resolve:ssm:reference-key}}`. Para obter mais informações, consulte [SSMos parâmetros](#) no Guia AWS CloudFormation do usuário.

O Infrastructure Composer cria o código de infraestrutura para configurar sua função Lambda com o valor do Parameter StoreSSM. Veja um exemplo a seguir:

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

### ID lógico

Você pode referenciar um `AWS::SSM::Parameter` recurso no mesmo modelo por ID lógica.

Veja a seguir um exemplo de um `AWS::SSM::Parameter` recurso chamado `PrivateSubnet1Parameter` que armazena o ID da sub-rede para `PrivateSubnet1`:

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

Veja a seguir um exemplo desse valor de recurso fornecido pelo ID lógico para a função Lambda:

### Subnet IDs

List of VPC subnet identifiers

Value	Type
<input type="text" value="PrivateSubnet1Parameter"/>	<input type="text" value="SSM"/>

O Infrastructure Composer cria o código de infraestrutura para configurar sua função Lambda com SSM o parâmetro:

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

## Tipo de valor estático

Quando um grupo de segurança ou sub-rede é implantado AWS CloudFormation, um valor de ID é criado. Você pode fornecer esse ID como um valor estático.

Para o tipo de valor estático, os valores a seguir são válidos:

- Para grupos de segurança, forneça `GroupId` o. Para obter mais informações, consulte [Valores de retorno](#) no Guia AWS CloudFormation do usuário. Veja um exemplo a seguir: `sg-0b61d5c742dc2c773`.
- Para sub-redes, forneça o. `SubnetId` Para obter mais informações, consulte [Valores de retorno](#) no Guia AWS CloudFormation do usuário. Veja um exemplo a seguir: `subnet-01234567890abcdef`.

O Infrastructure Composer cria o código de infraestrutura para configurar sua função Lambda com o valor estático. Veja um exemplo a seguir:

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

## Usando vários tipos

Para grupos de segurança e sub-redes, você pode usar vários tipos juntos. Veja a seguir um exemplo que configura três grupos de segurança para uma função Lambda fornecendo valores de tipos diferentes:

### Security group IDs

List of VPC security group identifiers

Value	Type
<input type="text" value="MySecurityGroup"/>	Parameter
	<button>Remove</button>
<input type="text" value="sg-0b61d5c742dc2c773"/>	Static value
	<button>Remove</button>
<input type="text" value="{{resolve::ssm::demo/sg-0b61d5c742dc23}}"/>	SSM
	<button>Remove</button>

Add new item

O Infrastructure Composer faz referência a todos os três valores da `SecurityGroupIds` propriedade:

```
...
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
```

## Configure funções Lambda com funções externas VPCs no Infrastructure Composer

Para começar a configurar uma função Lambda com VPC uma definida em outro modelo, use a placa de componente aprimorada da Função Lambda. Esse cartão representa uma função Lambda usando o tipo de `AWS::Serverless::Function` recurso AWS Serverless Application Model (AWS SAM).

Para configurar uma função Lambda com a VPC de um modelo externo

1. No painel de propriedades do recurso da Função Lambda, expanda a seção suspensa de VPCconfigurações (avançadas).
2. Selecione Atribuir a externo VPC.
3. Forneça valores para os grupos de segurança e sub-redes a serem configurados para a função Lambda. Para mais detalhes, consulte [Identificadores de grupo de segurança e sub-rede](#).
4. Salve suas alterações.

## Parâmetros em modelos importados para um externo VPC com o Infrastructure Composer

Quando você importa um modelo existente com parâmetros definidos para os grupos de segurança e sub-redes de um externoVPC, o Infrastructure Composer fornece uma lista suspensa para selecionar seus parâmetros.

Veja a seguir um exemplo da Parameters seção de um modelo importado:

```
    ...
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
...

```

Ao configurar uma função externa VPC para uma nova função Lambda na tela, esses parâmetros estarão disponíveis em uma lista suspensa. Veja um exemplo a seguir:

### Subnet IDs

List of VPC subnet identifiers

Value	Type
<input type="text" value="🔍  "/>	<input type="text" value="Parameter"/> ▼
VPCSubnets	
VPCSubnet	

## Limitações ao importar tipos de parâmetros da lista

Normalmente, você pode especificar vários identificadores de grupo de segurança e sub-rede para cada função Lambda. Se o modelo existente contiver tipos de parâmetros de lista, como `List<AWS::EC2::SecurityGroup::Id>` ou `List<AWS::EC2::Subnet::Id>`, você só poderá especificar um identificador.

Para obter mais informações sobre o tipo de listas de parâmetros, consulte [Tipos AWS de parâmetros específicos suportados](#) no Guia do AWS CloudFormation usuário.

Veja a seguir um exemplo de um modelo definido `VPCSecurityGroups` como um tipo de parâmetro de lista:

```

...
Parameters:
  VPCSecurityGroups:
    Description: Security group IDs generated by Infrastructure Composer

```

```
Type: List<AWS::EC2::SecurityGroup::Id>
```

```
...
```

No Infrastructure Composer, se você selecionar o `VPCSecurityGroups` valor como um identificador de grupo de segurança para uma função Lambda, você verá a seguinte mensagem:

### Security group IDs

List of VPC security group identifiers

Value	Type
<input style="width: 90%; border: none;" type="text" value="VPCSecurityGroups"/> <span style="float: right; color: blue; font-size: 1.2em;">✕</span>	<span style="font-weight: bold;">Parameter</span> <span style="float: right; color: blue; font-size: 1.2em;">▼</span>
<span style="border: 1px solid #ccc; border-radius: 15px; padding: 5px 20px; display: inline-block;">Add new item</span>	
<p style="color: #666; font-size: 0.9em;">Only one List&lt;AWS::EC2::SecurityGroup::Id&gt; parameter type can be provided.</p>	

Essa limitação ocorre porque as `SubnetIds` propriedades `SecurityGroupIds` e de um `AWS::Lambda::Function VpcConfig` objeto aceitam somente uma lista de valores de string. Como um único tipo de parâmetro de lista contém uma lista de cadeias de caracteres, ele pode ser o único objeto fornecido quando especificado.

Para tipos de parâmetros de lista, veja a seguir um exemplo de como eles são definidos no modelo quando configurados com uma função Lambda:

```
...
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
```

## Adicionando novos parâmetros aos modelos importados com o Infrastructure Composer

Ao importar um modelo existente com parâmetros definidos, você também pode criar novos parâmetros. Em vez de selecionar um parâmetro existente na lista suspensa, forneça um novo tipo e valor. Veja a seguir um exemplo que cria um novo parâmetro chamado `MySecurityGroup`:

### Security group IDs

List of VPC security group identifiers

Value	Type
<input style="width: 90%; border: none;" type="text" value="MySecurityGroup"/> <span style="float: right; border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">×</span>	Parameter <span style="float: right;">▼</span>
Use: "MySecurityGroup"	
VPCSecurityGroups	

Para todos os novos valores que você fornece no painel Propriedades do recurso para a função Lambda, o Infrastructure Composer os define em uma lista abaixo das `SubnetIds` propriedades `SecurityGroupIds` ou de uma função Lambda. Veja um exemplo a seguir:

```
...
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SecurityGroupIds:
          - sg-94b3a1f6
        SubnetIds:
          - !Ref SubnetParameter
          - !Ref VPCSubnet
```



Se você quiser referenciar a ID lógica de um tipo de parâmetro de lista a partir de um modelo externo, recomendamos que você use a visualização Modelo e modifique diretamente seu modelo. A ID lógica de um tipo de parâmetro de lista deve sempre ser fornecida como um valor único e como o único valor.

```

...
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

```

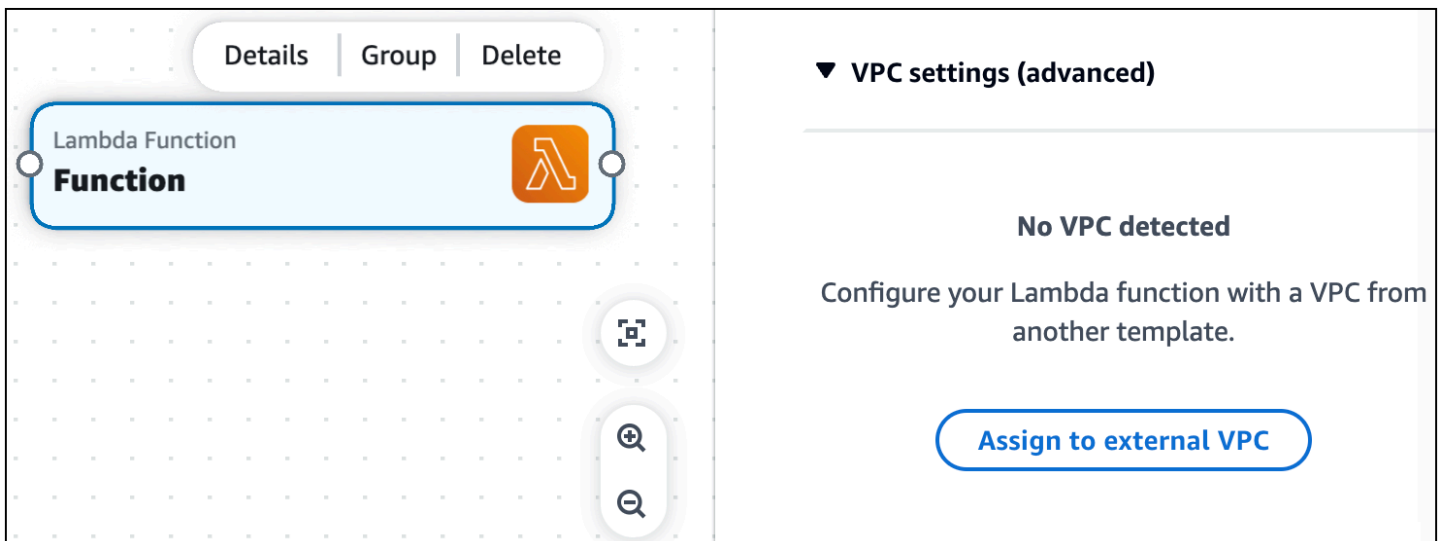
## Configure uma função Lambda e uma VPC definida em outro modelo com o Infrastructure Composer

Neste exemplo, configuramos uma função Lambda no Infrastructure Composer com uma VPC definida em outro modelo.

Começamos arrastando um cartão de componente aprimorado da Função Lambda para a tela.

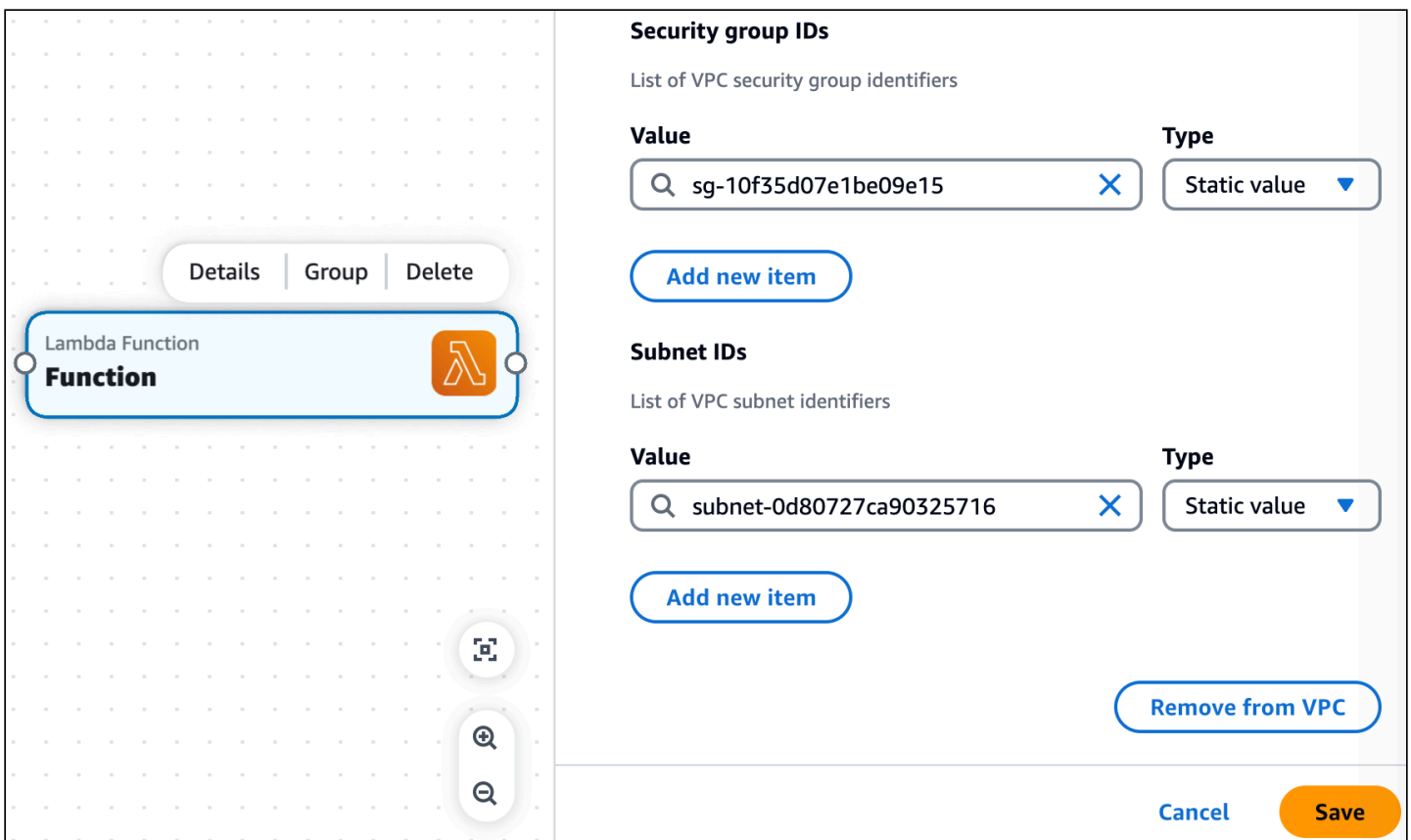


Em seguida, abrimos o painel de propriedades do recurso do cartão e expandimos a VPCseção suspensão de configurações (avançadas).

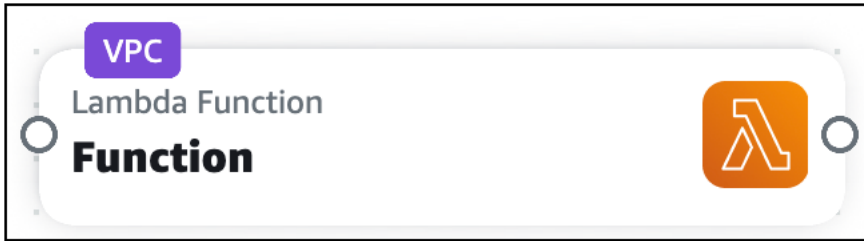


Em seguida, selecionamos Atribuir ao externo VPC para começar a configurar a VPC partir de um modelo externo.

Neste exemplo, referenciamos uma ID de grupo de segurança e uma ID de sub-rede. Esses valores são criados quando o modelo que define o VPC é implantado. Escolhemos o tipo de valor estático e inserimos o valor do nosso IDs. Selecionamos Salvar quando terminar.



Agora que nossa função Lambda está configurada com nossa VPC, a VPC tag é exibida em nosso cartão.



O Infrastructure Composer criou o código de infraestrutura para configurar nossa função Lambda com o grupo de segurança e a sub-rede do externo. VPC

```
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}
```

# Implemente seu aplicativo sem servidor Infrastructure Composer na nuvem AWS

Use AWS Infrastructure Composer para projetar aplicativos sem servidor prontos para implantação. Para implantar, use qualquer serviço AWS CloudFormation compatível. Recomendamos usar o [AWS Serverless Application Model \(AWS SAM\)](#).

AWS SAM é uma estrutura de código aberto que fornece ferramentas de desenvolvedor para criar e executar aplicativos sem servidor no. AWS Com AWS SAM a sintaxe abreviada, os desenvolvedores declaram AWS CloudFormation recursos e recursos especializados sem servidor que são transformados em infraestrutura durante a implantação.

## AWS SAM Conceitos importantes

Antes de usar AWS SAM, é importante que você se familiarize com alguns de seus conceitos fundamentais.

- [Como AWS SAM funciona](#): este tópico, que está no Guia do AWS Serverless Application Model desenvolvedor, fornece informações importantes sobre os componentes principais que você usa para criar seu aplicativo sem servidor: O AWS SAM CLI, o AWS SAM projeto e o AWS SAM modelo.
- [Como usar AWS Serverless Application Model \(AWS SAM\)](#): Este tópico, que está no Guia do AWS Serverless Application Model desenvolvedor, fornece uma visão geral de alto nível das etapas que você precisa concluir AWS SAM para implantar seu aplicativo na AWS nuvem.

Ao projetar seu aplicativo no Infrastructure Composer, você pode usar o `aws sam sync` comando para ter o AWS SAM CLI detecte automaticamente as alterações locais e implante essas alterações no AWS CloudFormation. Para saber mais, consulte [Usando o sam sync](#) no Guia do AWS Serverless Application Model desenvolvedor.

## Próximas etapas

Consulte [Configurado para implantação com o AWS SAM CLI e Infrastructure Composer](#) para se preparar para implantar seu aplicativo.

# Configurado para implantação com o AWS SAM CLI e Infrastructure Composer

Para implantar seu aplicativo com AWS SAM, primeiro você precisa instalar e acessar o AWS CLI e o AWS SAM CLI. Os tópicos desta seção fornecem detalhes sobre como fazer isso.

## Instale o AWS CLI

Recomendamos instalar e configurar o AWS CLI antes de instalar o AWS SAM CLI. Para obter instruções, consulte [Instalar ou atualizar para a versão mais recente do AWS CLI](#) no Guia AWS Command Line Interface do Usuário.

### Note

Depois de instalar o AWS CLI, você deve configurar as AWS credenciais. Para saber mais, consulte [Configuração rápida](#) no Guia do AWS Command Line Interface usuário.

## Instale o AWS SAM CLI

Para instalar o AWS SAM CLI, consulte [Instalando o AWS SAM CLI](#) no Guia do desenvolvedor do AWS Serverless Application Model .

## Acesse o AWS SAM CLI

Se você usa o Infrastructure Composer a partir do AWS Management Console, você tem as seguintes opções para usar o AWS SAM CLI.

### Ativar o modo de sincronização local

Com o modo de sincronização local, a pasta do projeto, incluindo o AWS SAM modelo, é salva automaticamente na sua máquina local. O Infrastructure Composer estrutura o diretório do seu projeto de uma forma que AWS SAM reconhece. Você pode executar o AWS SAM CLI do diretório raiz do seu projeto.

Para obter mais informações sobre o modo de sincronização local, consulte [Sincronize e salve localmente seu projeto no console do Infrastructure Composer](#).

## Exporte seu modelo

Você pode exportar seu modelo para sua máquina local. Em seguida, execute o AWS SAM CLI da pasta principal que contém o modelo. Você também pode usar a `--template-file` opção com qualquer AWS SAM CLI comande e forneça o caminho para seu modelo.

### Use o Infrastructure Composer do AWS Toolkit for Visual Studio Code

Você pode usar o Infrastructure Composer do Toolkit for VS Code para levar o Infrastructure Composer à sua máquina local. Em seguida, use o Infrastructure Composer e o AWS SAM CLI do VS Code.

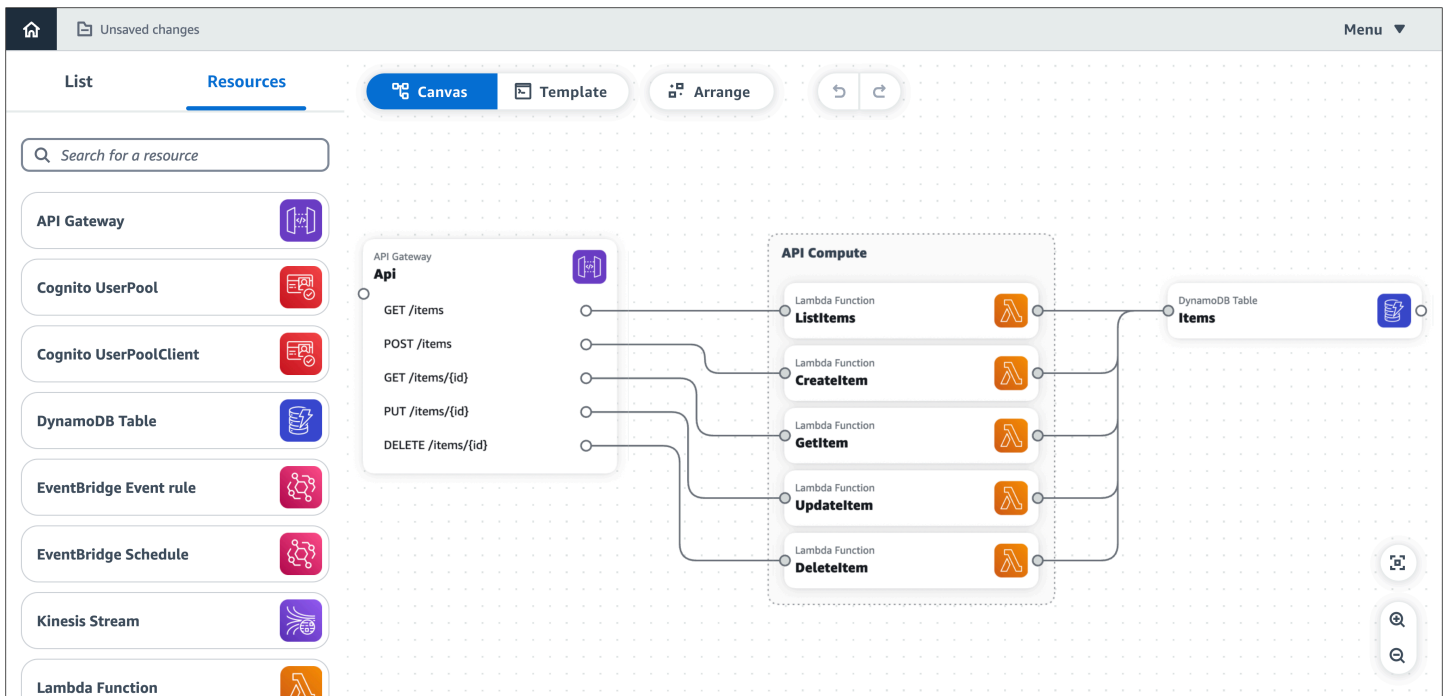
## Próximas etapas

Para implantar seu aplicativo, consulte [Use o Infrastructure Composer com AWS SAM para criar e implantar](#).

## Use o Infrastructure Composer com AWS SAM para criar e implantar

Agora que você concluiu [Configurado para implantação com o AWS SAM CLI e Infrastructure Composer](#), você pode implantar seu aplicativo com AWS SAM o Infrastructure Composer. Esta seção fornece um exemplo detalhando como você pode fazer isso. Você também pode consultar [Implantar seu aplicativo e recursos AWS SAM](#) no Guia do AWS Serverless Application Model desenvolvedor para obter instruções sobre como implantar seu aplicativo com AWS SAM.

Este exemplo mostra como criar e implantar o aplicativo de demonstração do Infrastructure Composer. O aplicativo de demonstração tem os seguintes recursos:



### Note

- Para saber mais sobre o aplicativo de demonstração, consulte [Carregue e modifique o projeto de demonstração do Infrastructure Composer](#).
- Neste exemplo, usamos o Infrastructure Composer com a sincronização local ativada.

1. Use o `sam build` comando para criar o aplicativo.

```
$ 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
```

O AWS SAM CLI cria o `./aws-sam` diretório na pasta do projeto. Esse diretório contém artefatos de construção para as funções Lambda do aplicativo. Aqui está uma saída do diretório do projeto:

```
.
### 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. Agora, o aplicativo está pronto para ser implantado. Nós usaremos `sam deploy --guided`. Isso prepara seu aplicativo para implantação por meio de uma série de prompts.

```
$ 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]:

```

O AWS SAM CLI exibe um resumo do que será implantado:

```

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-samcliarn-s3-demo-
bucket-1b3x26zbcdkqr
Capabilities         : ["CAPABILITY_IAM"]
Parameter overrides : {}
Signing Profiles    : {}

```

O AWS SAM CLI implanta o aplicativo, primeiro criando um conjunto de AWS CloudFormation alterações:

```

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 ResourceType	LogicalResourceId Replacement
+ Add AWS::ApiGateway::Deployment	ApiDeploymentccc153d135b N/A
+ Add AWS::ApiGateway::Stage	ApiProdStage N/A
+ Add AWS::ApiGateway::RestApi	Api N/A
+ Add AWS::Lambda::Permission	CreateItemApiPOSTitemsPermissionP N/A
+ Add AWS::IAM::Role	rod CreateItemRole N/A
+ Add AWS::Lambda::Function	CreateItem N/A
+ Add AWS::Lambda::Permission	DeleteItemApiDELETEitemsidPermiss N/A
+ Add AWS::IAM::Role	ionProd DeleteItemRole N/A
+ Add AWS::Lambda::Function	DeleteItem N/A
+ Add AWS::Lambda::Permission	GetItemApiGETitemsidPermissionPro N/A
+ Add AWS::IAM::Role	d GetItemRole N/A
+ Add AWS::Lambda::Function	GetItem N/A
+ Add AWS::DynamoDB::Table	Items N/A
+ Add AWS::Lambda::Permission	ListItemsApiGETitemsPermissionPro N/A
+ Add AWS::IAM::Role	d ListItemsRole N/A
+ Add AWS::Lambda::Function	ListItems N/A
+ Add AWS::Lambda::Permission	UpdateItemApiPUTitemsidPermission N/A
+ Add AWS::IAM::Role	Prod UpdateItemRole 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
```

Então, o AWS SAM CLI implanta o aplicativo:

```
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	
ApiDeploymentccc153d135b	-	
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	
ApiDeploymentccc153d135b	Resource creation Initiated	
CREATE_COMPLETE	AWS::ApiGateway::Deployment	
ApiDeploymentccc153d135b	-	
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	-	

```

CREATE_COMPLETE          AWS::Lambda::Permission          rod
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      -
-----

```

Finalmente, uma mensagem é exibida informando que a implantação foi bem-sucedida:

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

## Use o Infrastructure Composer com AWS SAM para excluir uma pilha

Este exemplo mostra como excluir uma AWS CloudFormation pilha usando o `aws-sam delete` comando.

Digite o comando `aws-sam delete` no campo AWS SAM CLI e confirme se você deseja excluir a pilha e o modelo:

```

$ aws-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 Infrastructure Composer solução de problemas

Os tópicos desta seção fornecem orientação sobre a solução de problemas de mensagens de erro durante o uso AWS Infrastructure Composer.

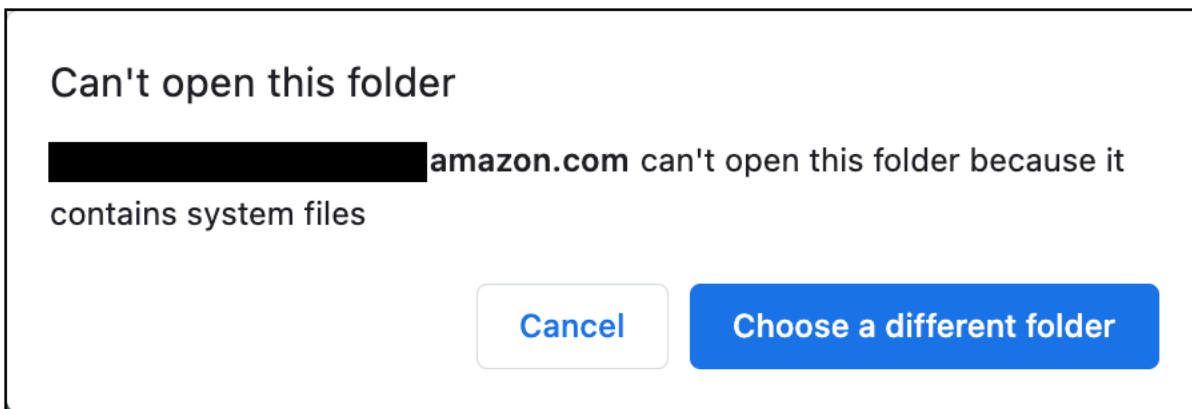
Tópicos

- [Mensagens de erro](#)

## Mensagens de erro

### “Não consigo abrir esta pasta”

Exemplo de erro:



Possível causa: o Infrastructure Composer não consegue acessar um diretório confidencial usando o modo de sincronização local.

Para saber mais sobre esse erro, consulte [O Data Infrastructure Composer obtém acesso a](#).

Tente se conectar a um diretório local diferente ou usar o Infrastructure Composer com a sincronização local desativada.

### “Modelo incompatível”

Exemplo de erro: ao carregar um novo projeto no Infrastructure Composer, você vê o seguinte:



Possível causa: Seu projeto contém um arquivo referenciado externamente que não é compatível com o Infrastructure Composer.

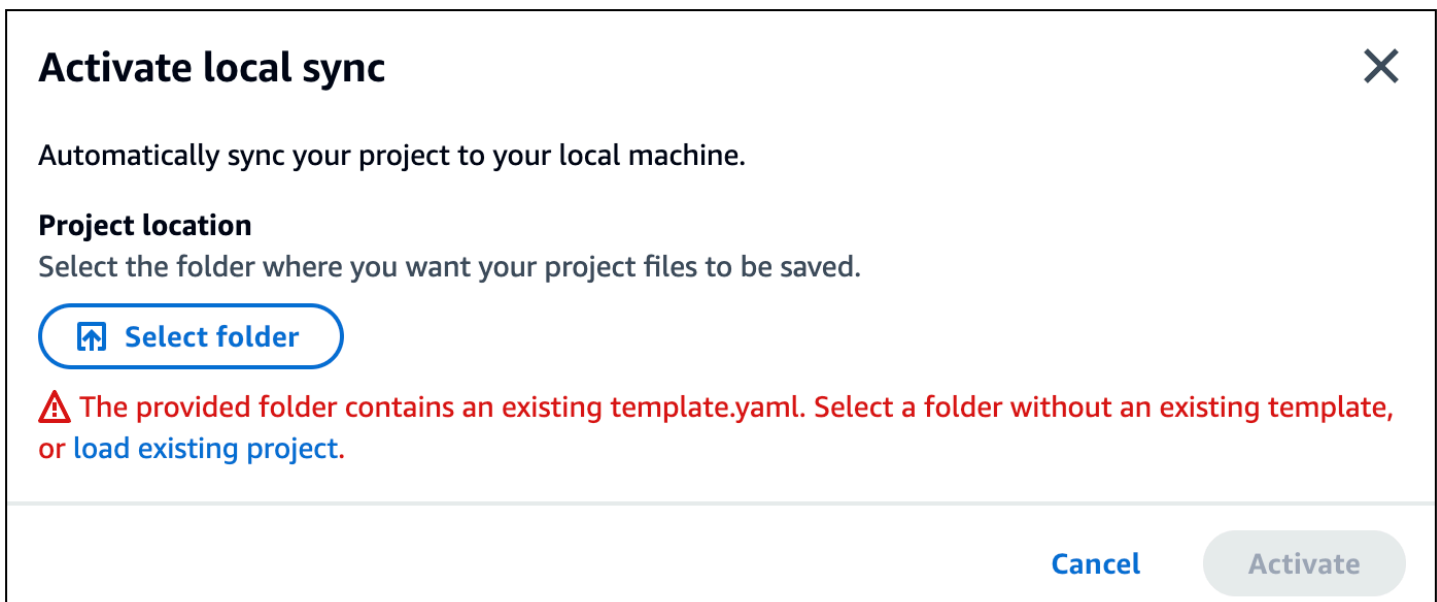
Para saber mais sobre os arquivos externos compatíveis no Infrastructure Composer, consulte [Arquivos externos de referência](#).

Possível causa: Seu projeto está vinculado a um arquivo externo em um diretório local diferente.

Mova seu arquivo referenciado externamente para um subdiretório do diretório que você selecionou para usar com o modo de sincronização local do Infrastructure Composer.

## “A pasta fornecida contém um template.yaml existente”

Ao tentar ativar a sincronização local, você vê o seguinte erro:



Possível causa: sua pasta selecionada já contém um arquivo template.yaml.

Selecione outro diretório que não contenha um modelo de aplicativo ou crie um novo diretório.

## “Seu navegador não tem permissões para salvar seu projeto nessa pasta...”

Possível causa: o Infrastructure Composer não consegue acessar um diretório confidencial usando o modo de sincronização local.

Para saber mais sobre esse erro, consulte [O Data Infrastructure Composer obtém acesso a](#).

Tente se conectar a um diretório local diferente ou usar o Infrastructure Composer com a sincronização local desativada.

# Segurança em AWS Infrastructure Composer

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade aplicáveis AWS Infrastructure Composer, consulte [AWS Serviços no escopo do programa de conformidade AWS](#) .
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Infrastructure Composer. Os tópicos a seguir mostram como configurar o Infrastructure Composer para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do Infrastructure Composer.

## Tópicos

- [Proteção de dados em AWS Infrastructure Composer](#)
- [AWS Identity and Access Management para AWS Infrastructure Composer](#)
- [Validação de conformidade para AWS Infrastructure Composer](#)
- [Resiliência em AWS Infrastructure Composer](#)

## Proteção de dados em AWS Infrastructure Composer

O modelo de [responsabilidade AWS compartilhada O modelo](#) se aplica à proteção de dados no AWS Infrastructure Composer. Conforme descrito neste modelo, AWS é responsável por proteger

a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte [Privacidade de dados FAQ](#). Para obter informações sobre proteção de dados na Europa, consulte o [Modelo de Responsabilidade AWS Compartilhada e GDPR](#) a postagem no blog AWS de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use a autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a descobrir e proteger dados confidenciais armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Infrastructure Composer ou outro Serviços da AWS usando o console, API, AWS CLI, ou AWS SDKs. Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

**Note**

Todos os dados que você insere no Infrastructure Composer são usados com o único propósito de fornecer funcionalidade no Infrastructure Composer e gerar arquivos e diretórios de projeto que são salvos localmente em sua máquina. O Infrastructure Composer não salva, armazena ou transmite nenhum desses dados.

## Criptografia de dados

O Infrastructure Composer não criptografa o conteúdo do cliente, pois os dados não são salvos, armazenados ou transmitidos.

### Criptografia em repouso

O Infrastructure Composer não criptografa o conteúdo do cliente, pois os dados não são salvos, armazenados ou transmitidos.

### Criptografia em trânsito

O Infrastructure Composer não criptografa o conteúdo do cliente, pois os dados não são salvos, armazenados ou transmitidos.

### Gerenciamento de chaves

O Infrastructure Composer não oferece suporte ao gerenciamento de chaves, pois o conteúdo do cliente não é salvo, armazenado ou transmitido.

### Privacidade do tráfego entre redes

O Infrastructure Composer não gera tráfego com clientes e aplicativos locais.

## AWS Identity and Access Management para AWS Infrastructure Composer

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões)

para usar os recursos do Infrastructure Composer. IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

## Tópicos

- [Público](#)
- [Autenticação com identidades](#)
- [Como gerenciar acesso usando políticas](#)
- [Como AWS Infrastructure Composer funciona com IAM](#)

## Público

O Infrastructure Composer requer, no mínimo, acesso somente de leitura ao AWS Management Console. Qualquer usuário com essa autorização pode usar todos os recursos do Infrastructure Composer. O acesso granular a recursos específicos do Infrastructure Composer não é suportado.

## Autenticação com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades usando funções do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de acesso AWS. Para obter mais informações sobre como fazer login em AWS, consulte [Como fazer login Conta da AWS](#) no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte [AWS Signature versão 4 para API solicitações](#) no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia do AWS IAM Identity Center usuário e [Autenticação AWS multifator IAM no](#) Guia do IAMusuário.

## Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa de tarefas que requerem login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do usuário do IAM.

## Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

## Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas

e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Casos de uso para IAM usuários](#) no Guia IAM do usuário.

## Funções do IAM

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ela é semelhante a um usuário do IAM, mas não está associada a uma pessoa específica. Para assumir temporariamente uma IAM função no AWS Management Console, você pode [alternar de um usuário para uma IAM função \(console\)](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Métodos para assumir uma função](#) no Guia IAM do usuário.

As funções do IAM com credenciais temporária são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para obter informações sobre funções para federação, consulte [Criar uma função para um provedor de identidade terceirizado \(federação\)](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Guia do Usuário do AWS IAM Identity Center .
- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.



- **Acesso entre contas:** é possível usar uma função do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte [Acesso a recursos entre contas IAM no Guia](#) do IAM usuário.
- **Acesso entre serviços** — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
- **Sessões de acesso direto (FAS)** — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que, em seguida, inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- **Função de serviço** — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar uma função para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.
- **Função vinculada ao serviço** — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.
- **Aplicativos em execução na Amazon EC2** — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. É preferível fazer isso do que armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas que estejam em execução na instância

do EC2 obtenham credenciais temporárias. Para obter mais informações, consulte [Usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

## Como gerenciar acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder aos usuários permissões para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode, então, adicionar as políticas do IAM às funções, e os usuários podem assumir as funções.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

### Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais atributos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia](#) do IAM usuário.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos

e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha no Guia do IAMusuário](#).

## Políticas baseadas em recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. As políticas de confiança de função do IAM e as políticas de bucket do Amazon S3 são exemplos de políticas baseadas em recursos. Em serviços que seja compatível com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

## Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAMusuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou

o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para entidades do IAM](#) no Guia do usuário do IAM.

- Políticas de controle de serviço (SCPs) — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- Políticas de controle de recursos (RCPs) — RCPs são JSON políticas que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as IAM políticas anexadas a cada recurso que você possui. Isso RCP limita as permissões de recursos nas contas dos membros e pode afetar as permissões efetivas das identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da AWS desse suporte RCPs, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para ter mais informações, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

## Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.

## Como AWS Infrastructure Composer funciona com IAM

AWS Infrastructure Composer requer, no mínimo, acesso somente de leitura ao AWS Management Console. Qualquer usuário com essa autorização pode usar todos os recursos do Infrastructure Composer. O acesso granular a recursos específicos do Infrastructure Composer não é suportado.

Ao implantar o modelo e os arquivos do seu projeto AWS CloudFormation, você precisará que as permissões necessárias estejam em vigor. Para saber mais, consulte [Controlar o acesso com AWS Identity and Access Management](#) no AWS CloudFormation Guia do usuário.

A tabela a seguir mostra com quais IAM recursos podem ser usados AWS Infrastructure Composer.

Recurso do IAM	Suporte ao Infrastructure Composer
<a href="#">Políticas baseadas em identidade</a>	Não
<a href="#">Políticas baseadas em recurso</a>	Não
<a href="#">Ações de políticas</a>	Não
<a href="#">Recursos de políticas</a>	Não
<a href="#">Chaves de condição de políticas</a>	Não
<a href="#">ACLs</a>	Não
<a href="#">ABAC(tags nas políticas)</a>	Não
<a href="#">Credenciais temporárias</a>	Sim
<a href="#">Permissões de entidade principal</a>	Não
<a href="#">Perfis de serviço</a>	Não
<a href="#">Funções vinculadas ao serviço</a>	Não

Para obter uma visão de alto nível de como o Infrastructure Composer e outros AWS serviços funcionam com a maioria dos IAM recursos, consulte [AWS os serviços que funcionam com IAM](#) no Guia do IAM usuário.

## Políticas baseadas em identidade para o Infrastructure Composer

Suporta políticas baseadas em identidade: Não

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais atributos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia](#) do IAM usuário.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a [referência IAM JSON de elementos de política](#) no Guia IAM do usuário.

## Políticas baseadas em recursos no Infrastructure Composer

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. As políticas de confiança de função do IAM e as políticas de bucket do Amazon S3 são exemplos de políticas baseadas em recursos. Em serviços que seja compatível com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como o principal em uma política baseada em recurso. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será

necessária. Para obter mais informações, [consulte Acesso a recursos entre contas IAM no Guia do IAM usuário](#).

## Ações políticas para o Infrastructure Composer

Apoia ações políticas: Não

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da AWS API operação associada. Há algumas exceções, como ações somente com permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações do Infrastructure Composer, consulte [Ações definidas pelo AWS Infrastructure Composer na Referência](#) de Autorização de Serviço.

## Recursos de política para o Infrastructure Composer

Oferece compatibilidade com recursos de políticas: não

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Resource` JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu [Amazon Resource Name \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (\*) para indicar que a instrução se aplica a todos os recursos.

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



Para ver uma lista dos tipos de recursos do Infrastructure Composer e seus ARNs, consulte [Recursos definidos pelo AWS Infrastructure Composer na Referência de Autorização de Serviço](#). Para saber com quais ações você pode especificar cada recurso, consulte [Ações definidas pelo AWS Infrastructure Composer. ARN](#)

## Chaves de condição de política para o Infrastructure Composer

Suporte a chaves de condição de políticas específicas do serviço: não

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco de `Condition`) permite que você especifique condições nas quais uma declaração está em vigor. O elemento `Condition` é opcional. É possível criar expressões de condição que usam [operadores de condição](#), como igual a ou menor que, para corresponder a condição na política aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte [Elementos de política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia IAM do usuário.

Para ver uma lista das chaves de condição do Infrastructure Composer, consulte [Chaves de condição do AWS Infrastructure Composer](#) na Referência de Autorização de Serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas pelo AWS Infrastructure Composer](#).

## ACLs no Infrastructure Composer

Suporte a ACLs: Não



As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

## ABAC com o Infrastructure Composer

Suportes ABAC (tags nas políticas): Não

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com todas as três chaves de condição somente de alguns tipos de recurso, o valor será Parcial.

Para obter mais informações sobre ABAC, consulte [Definir permissões com ABAC autorização](#) no Guia IAM do usuário. Para ver um tutorial com etapas de configuração ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\) no Guia](#) do IAM usuário.

## Usando credenciais temporárias com o Infrastructure Composer

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS nesse trabalho IAM](#) no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no

console como usuário e alterna perfis. Para obter mais informações sobre a troca de funções, consulte [Alternar de um usuário para uma IAM função \(console\)](#) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias em IAM](#).

Você pode usar credenciais temporárias para acessar o Infrastructure Composer por meio do AWS Management Console. Por exemplo, consulte [Habilitar o acesso personalizado do agente de identidade ao AWS console](#) no Guia IAM do usuário.

## Permissões principais entre serviços para o Infrastructure Composer

Suporta sessões de acesso direto (FAS): Não

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que, em seguida, inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).

## Funções de serviço do Infrastructure Composer

Compatível com perfis de serviço: não

Um perfil de serviço é um [perfil do IAM](#) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar uma função para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.

### Warning

Alterar as permissões de uma função de serviço pode interromper a funcionalidade do Infrastructure Composer. Edite as funções de serviço somente quando o Infrastructure Composer fornecer orientação para fazer isso.

## Funções vinculadas a serviços para o Infrastructure Composer

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

Para obter detalhes sobre a criação ou o gerenciamento de funções vinculadas a serviço, consulte [Serviços da AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

## Validação de conformidade para AWS Infrastructure Composer

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Governança e conformidade de segurança](#): esses guias de implementação de solução abordam considerações sobre a arquitetura e fornecem etapas para implantar recursos de segurança e conformidade.
- [Referência de serviços qualificados para HIPAA](#): lista os serviços qualificados para HIPAA. Nem todos Serviços da AWS são elegíveis para a HIPAA.
- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.

- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

## Resiliência em AWS Infrastructure Composer

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Todos os dados que você insere no Infrastructure Composer são usados com o único propósito de fornecer funcionalidade no Infrastructure Composer e gerar arquivos e diretórios de projeto que são salvos localmente em sua máquina. O Infrastructure Composer não salva nem armazena nenhum desses dados.

# Histórico de documentos para Infrastructure Composer

A tabela a seguir descreve as versões importantes da documentação do Infrastructure Composer. Para receber notificações sobre atualizações desta documentação, você pode assinar um RSS feed.

- Última atualização da documentação: 30 de novembro de 2023

Alteração	Descrição	Data
<a href="#">Reestruturado e atualizado o conteúdo em todo o guia do desenvolvedor</a>	Reorganizado e reestruturado o guia para melhorar a capacidade de descoberta e a usabilidade. Títulos atualizados e aprimorados. Fornecidos detalhes adicionais na apresentação de tópicos e conceitos.	1.º de agosto de 2024
<a href="#">Foi adicionada documentação para usar o Infrastructure Composer no modo CloudFormation console e reestruturou o Guia do desenvolvedor do Infrastructure Composer.</a>	AWS Infrastructure Composer agora pode ser usado no modo AWS CloudFormation console. Para saber mais, consulte <a href="#">Usando o Infrastructure Composer no modo CloudFormation console</a> . Além disso, grande parte do conteúdo do guia do usuário foi reorganizada para criar uma experiência simplificada.	28 de março de 2024
<a href="#">Documentação adicionada para a integração do Infrastructure Composer com CodeWhisperer</a>	AWS Infrastructure Composer do Toolkit for VS Code fornece uma integração com a Amazon CodeWhisperer. Para saber mais, consulte <a href="#">Usando AWS Infrastructure Composer</a>	30 de novembro de 2023

<a href="#">Documentação adicionada para implantar seu aplicativo com o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code</a>	<a href="#">com a Amazon CodeWhisperer</a> . Use o botão de sincronização da tela do Infrastructure Composer para implantar seu aplicativo no Nuvem AWS. Para saber mais, consulte <a href="#">Implantar seu aplicativo com o sam sync</a> .	30 de novembro de 2023
<a href="#">Documentação adicionada para o Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code</a>	Agora você pode usar o Infrastructure Composer do VS Code com o AWS Toolkit for Visual Studio Code Para saber mais, consulte <a href="#">Usando AWS Infrastructure Composer a partir do AWS Toolkit for Visual Studio Code</a> .	30 de novembro de 2023
<a href="#">Foi adicionada a integração com o Step Functions Workflow Studio</a>	Inicie o Step Functions Workflow Studio a partir da tela do Infrastructure Composer. Para saber mais, consulte <a href="#">Usando AWS Infrastructure Composer com AWS Step Functions</a> .	27 de novembro de 2023
<a href="#">Foi adicionada a integração do console Lambda e do Infrastructure Composer</a>	Inicie a tela do Infrastructure Composer a partir do console Lambda. Para saber mais, consulte <a href="#">Usando AWS Infrastructure Composer com o AWS Lambda console</a> .	14 de novembro de 2023

[A Amazon foi adicionada VPC como um serviço em destaque com o Infrastructure Composer](#)

O Infrastructure Composer apresenta uma VPC tag para visualizar recursos configurados com um VPC. Você também pode configurar as funções do Lambda com as VPCs definidas em um modelo externo. Para saber mais, consulte [Usando o Infrastructure Composer com a Amazon VPC](#).

17 de outubro de 2023

[A Amazon foi adicionada RDS como um serviço em destaque com o Infrastructure Composer](#)

Conecte seu aplicativo Infrastructure Composer a um cluster ou instância de banco de RDS dados Amazon definido em um modelo externo. Para saber mais, consulte [Usando o Infrastructure Composer com a Amazon RDS](#).

17 de outubro de 2023

[Foi adicionado suporte ao Infrastructure Composer para projetar com todos os recursos AWS CloudFormation](#)

Selecione qualquer AWS CloudFormation recurso da paleta Resources para criar seus aplicativos. Para saber mais, consulte [Trabalhar com qualquer AWS CloudFormation recurso](#).

26 de setembro de 2023

[Documentação adicionada para cartões no Infrastructure Composer](#)

O Infrastructure Composer suporta vários tipos de cartões que você pode usar para projetar e criar seu aplicativo. Para saber mais, consulte [Criação com cartões no Infrastructure Composer](#).

20 de setembro de 2023



<a href="#">Documentação adicionada para o recurso de desfazer e refazer</a>	Use os botões desfazer e refazer na tela do Infrastructure Composer. Para saber mais, consulte <a href="#">Desfazer e refazer</a> .	1º de agosto de 2023
<a href="#">Documentação adicionada para o modo de sincronização local</a>	Use o modo de sincronização local para sincronizar e salvar automaticamente seu projeto em sua máquina local. Para saber mais, consulte <a href="#">Modo de sincronização local</a> .	1º de agosto de 2023
<a href="#">Documentação adicionada para o recurso de exportação de tela</a>	Use o recurso de exportação de tela para exportar a tela do seu aplicativo como uma imagem para sua máquina local. Para saber mais, consulte <a href="#">Exportar tela</a> .	1º de agosto de 2023
<a href="#">Suporte do Infrastructure Composer para referências de arquivos externos</a>	Faça referência a arquivos externos para recursos compatíveis no Infrastructure Composer. Para saber mais, consulte Como <a href="#">trabalhar com modelos que fazem referência a arquivos externos</a> .	17 de maio de 2023
<a href="#">Nova documentação sobre a conexão de recursos</a>	Conecte recursos para definir relacionamentos orientados por eventos entre os recursos em seu aplicativo. Para saber mais, consulte <a href="#">Conectando recursos usando a tela visual do Infrastructure Composer</a> .	7 de março de 2023

<a href="#">Novo recurso do Change Inspector</a>	Use o Change Inspector para ver as atualizações do código do seu modelo e saber o que o Infrastructure Composer está criando para você. Para saber mais, consulte <a href="#">Ver atualizações de código com o Inspector de Alterações</a> .	7 de março de 2023
<a href="#">O Infrastructure Composer agora está em disponibilidade geral</a>	AWS Infrastructure Composer agora está disponível ao público em geral. Para saber mais, consulte <a href="#">AWS Infrastructure Composer agora disponível ao público em geral - Crie visualmente aplicativos sem servidor com rapidez</a> .	7 de março de 2023
<a href="#">Expandiu os benefícios de usar o modo conectado</a>	Use o Infrastructure Composer no modo conectado com seu local IDE para acelerar o desenvolvimento. Para saber mais, consulte <a href="#">Usando o Infrastructure Composer com seu local IDE</a> .	7 de março de 2023
<a href="#">Tópico atualizado sobre o uso de outros AWS serviços para implantar seu aplicativo</a>	Use o Infrastructure Composer para projetar aplicativos sem servidor prontos para implantação. Use AWS SAM para implantar seu aplicativo sem servidor. Para saber mais, consulte <a href="#">Usando o Infrastructure Composer com AWS CloudFormation e. AWS SAM</a>	3 de março de 2023

[Seção de conceitos sem servidor adicionada](#)

Aprenda sobre os conceitos básicos sem servidor antes de usar o Infrastructure Composer. Para saber mais, consulte [Conceitos sem servidor](#).

2 de março de 2023

[Versão pública](#)

Lançamento público inicial do Infrastructure Composer.

1º de dezembro de 2022

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.