



Guía para desarrolladores

# AWS Infrastructure Composer



# AWS Infrastructure Composer: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

---

# Table of Contents

¿Qué es Infrastructure Composer? .....	1
Componga su arquitectura .....	2
Defina sus plantillas .....	4
Intégrelo con sus flujos de trabajo .....	5
Formas de acceder a Infrastructure Composer .....	6
Más información .....	8
Pasos a seguir a continuación .....	8
Conceptos sobre sistemas sin servidor .....	8
Conceptos sobre sistemas sin servidor .....	9
Tarjetas .....	10
Tarjetas de componentes mejoradas .....	11
Ejemplo .....	12
Tarjetas de componentes estándar .....	13
Conexiones de tarjetas .....	17
Conexiones entre tarjetas .....	17
Conexiones entre tarjetas de componentes mejorados .....	18
Conexiones hacia y desde tarjetas de recursos laC estándar .....	20
Introducción .....	21
Haga un recorrido por la consola .....	21
Pasos a seguir a continuación .....	22
Cargue y modifique .....	22
Paso 1: Abre la demostración .....	23
Paso 2: Explore el lienzo visual .....	23
Paso 3: Amplíe su arquitectura .....	27
Paso 4: Guarda tu solicitud .....	28
Pasos a seguir a continuación .....	29
Compilación .....	29
Propiedades de recursos .....	30
Paso 1: Crea tu proyecto .....	30
Añadir tarjetas .....	33
Paso 3: Configure su REST API .....	34
Paso 4: Configure sus funciones .....	35
Paso 5: Conecta tus tarjetas .....	36
Paso 6: Organice el lienzo .....	37

Agregar una tabla de DynamoDB .....	38
Paso 8: Revisa tu plantilla .....	39
Paso 9: Intégrelo en sus flujos de trabajo .....	40
Pasos a seguir a continuación .....	40
Dónde usar Infrastructure Composer .....	41
Consola Infrastructure Composer .....	41
Descripción visual .....	42
Administre su proyecto .....	45
Conéctese a su local IDE .....	48
Permitir el acceso a la página web .....	51
Sincronice y guarde de forma local .....	52
Importación desde la consola Lambda .....	56
Exporte el lienzo .....	56
CloudFormation modo consola .....	58
¿Por qué usar este modo? .....	58
Acceda a este modo .....	59
Visualice una implementación .....	59
Cree una plantilla nueva .....	60
Actualice una pila existente .....	61
AWS Toolkit for Visual Studio Code .....	63
Descripción visual .....	64
Acceda desde VS Code .....	65
Sincronizar con Nube de AWS .....	66
Infrastructure Composer con Amazon Q .....	68
¿Cómo componer .....	71
Coloque las tarjetas en el lienzo .....	71
Agrupe las tarjetas .....	72
Agrupar tarjetas de componentes mejorados .....	72
Agrupar una tarjeta de componentes estándar en otra .....	73
Connect tarjetas .....	75
Conexión de tarjetas de componentes mejorados .....	75
Conexión de tarjetas estándar .....	76
Ejemplos .....	78
Desconecte las tarjetas .....	80
Tarjetas con componentes mejorados .....	80
Tarjetas de componentes estándar .....	80

Organice las tarjetas .....	82
Configure y modifique las tarjetas .....	83
Tarjetas mejoradas .....	84
Tarjetas estándar .....	100
Eliminar tarjetas .....	101
Tarjetas de componentes mejoradas .....	101
Tarjetas de componentes estándar .....	102
Vea las actualizaciones de código .....	102
Ventajas del Inspector de Cambios .....	103
Procedimiento .....	103
Más información .....	105
Haga referencia a archivos externos .....	106
Prácticas recomendadas .....	107
Cree una referencia de archivo externo .....	107
Cargue un proyecto .....	108
Cree una aplicación mediante AWS SAM CLI .....	109
Haga referencia a un OpenAPI Especificación .....	113
Integre con Amazon VPC .....	116
Identifique los recursos y la información .....	117
Configure las funciones .....	123
Parámetros de las plantillas importadas .....	123
Añadir nuevos parámetros a las plantillas importadas .....	126
Configurar una función Lambda con a VPC en otra plantilla .....	127
Implemente en la nube AWS .....	130
Conceptos importantes AWS SAM .....	130
Pasos a seguir a continuación .....	130
Configure el AWS SAM CLI .....	131
Instale el AWS CLI .....	131
Instale el AWS SAM CLI .....	131
Acceda al AWS SAM CLI .....	131
Pasos a seguir a continuación .....	132
Cree e implemente .....	132
Eliminar una pila .....	140
Solución de problemas .....	142
Mensajes de error .....	142
«No se puede abrir esta carpeta» .....	142

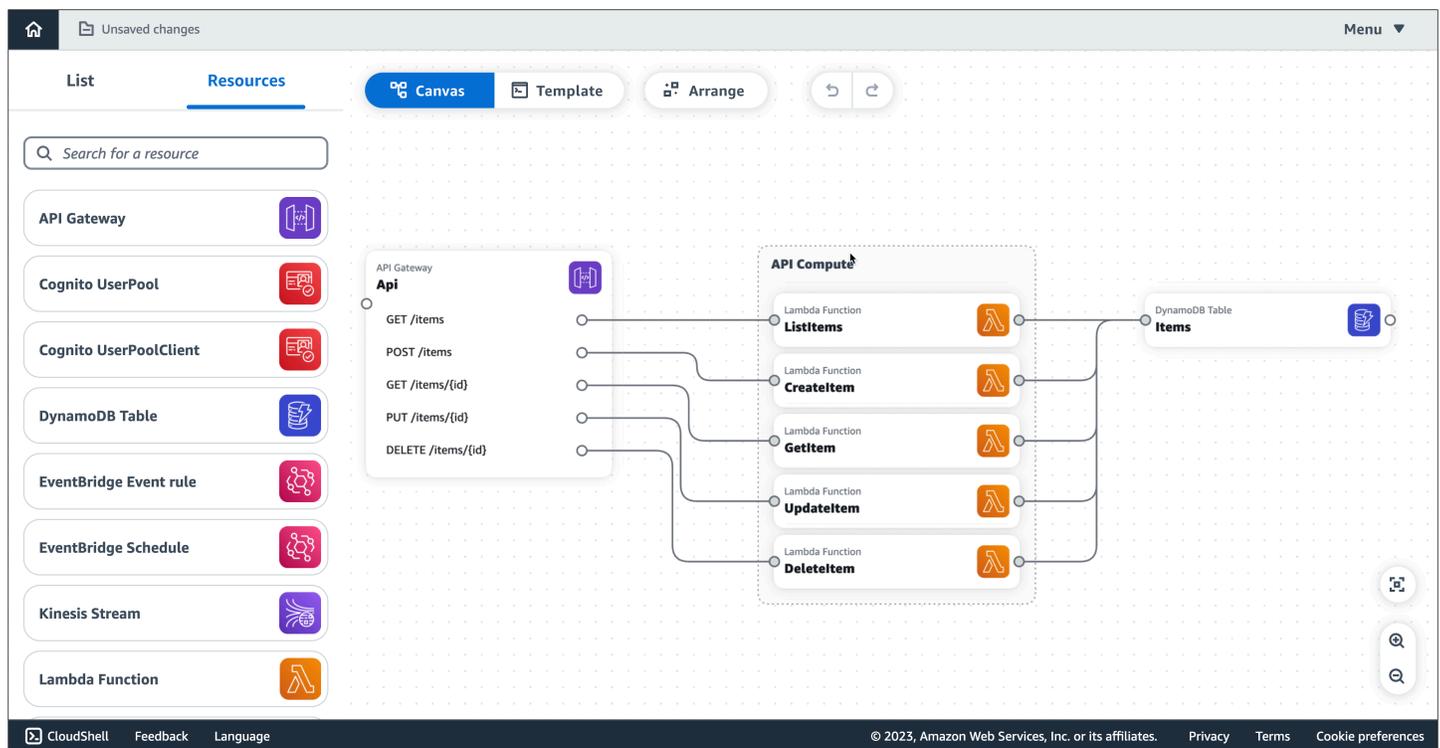
---

«Plantilla incompatible» .....	142
«La carpeta proporcionada contiene un template.yaml existente» .....	143
«Su navegador no tiene permisos para guardar el proyecto en esa carpeta...» .....	144
Seguridad .....	145
Protección de los datos .....	145
Cifrado de datos .....	147
Cifrado en tránsito .....	147
Administración de claves .....	147
Privacidad del tráfico entre redes .....	147
AWS Identity and Access Management .....	147
Público .....	148
Autenticación con identidades .....	148
Administración de acceso mediante políticas .....	152
¿Cómo AWS Infrastructure Composer funciona con IAM .....	155
Validación de conformidad .....	161
Resiliencia .....	163
Historial de documentos .....	164
.....	clxx

# ¿Qué es AWS Infrastructure Composer?

AWS Infrastructure Composer le permite componer visualmente aplicaciones modernas en AWS. Más específicamente, puede usar Infrastructure Composer para visualizar, crear e implementar aplicaciones modernas desde todos los AWS servicios compatibles AWS CloudFormation sin necesidad de ser un experto en ellos AWS CloudFormation.

A medida que configura su AWS CloudFormation infraestructura, a través de una drag-and-drop interfaz atractiva, Infrastructure Composer crea su infraestructura como plantillas de código (IaC) y, al mismo tiempo, sigue las prácticas AWS recomendadas. La siguiente imagen muestra lo fácil que es arrastrar, soltar, configurar y conectar recursos en el lienzo visual de Infrastructure Composer.



Infrastructure Composer se puede utilizar desde la consola de Infrastructure Composer, desde y en CloudFormation modo consola. AWS Toolkit for Visual Studio Code

## Temas

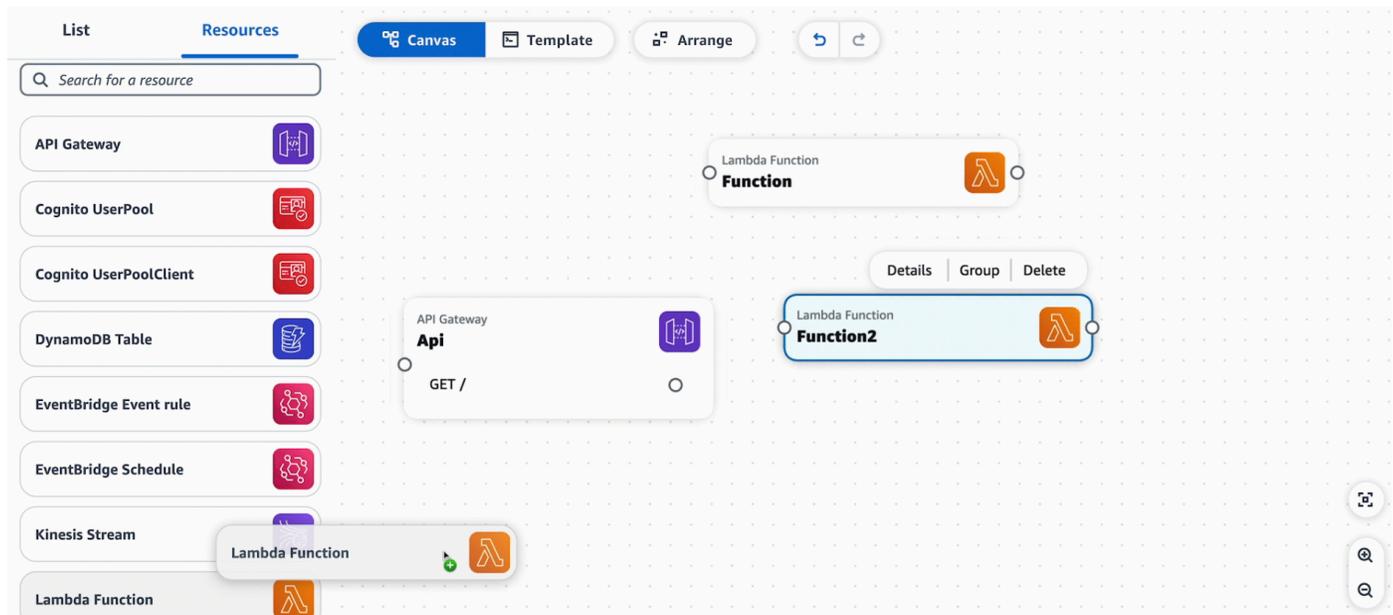
- [Cree la arquitectura de su aplicación](#)
- [Defina su infraestructura como plantillas de código \(IaC\)](#)
- [Intégrelo con sus flujos de trabajo existentes](#)
- [Formas de acceder a Infrastructure Composer](#)

- [Más información](#)
- [Pasos a seguir a continuación](#)
- [Conceptos sin servidor para AWS Infrastructure Composer](#)

## Cree la arquitectura de su aplicación

### Construye con cartas

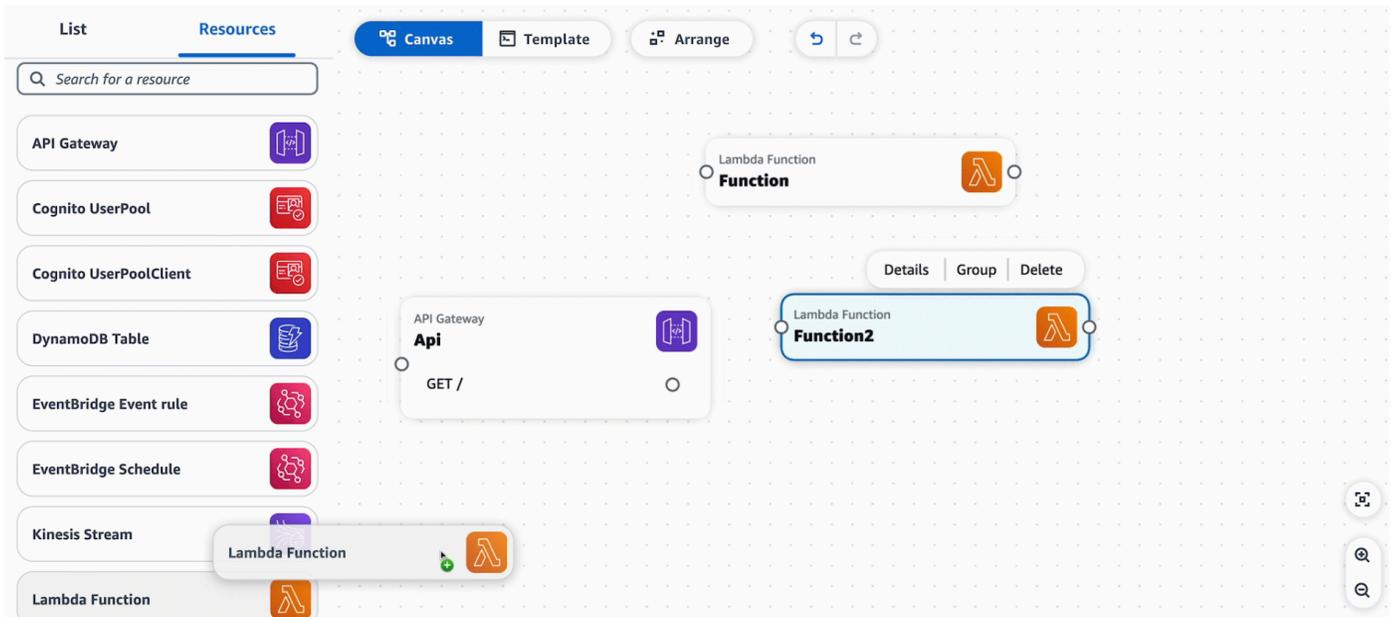
Coloque las tarjetas en el lienzo de Infrastructure Composer para visualizar y crear la arquitectura de su aplicación.



### Connect tarjetas

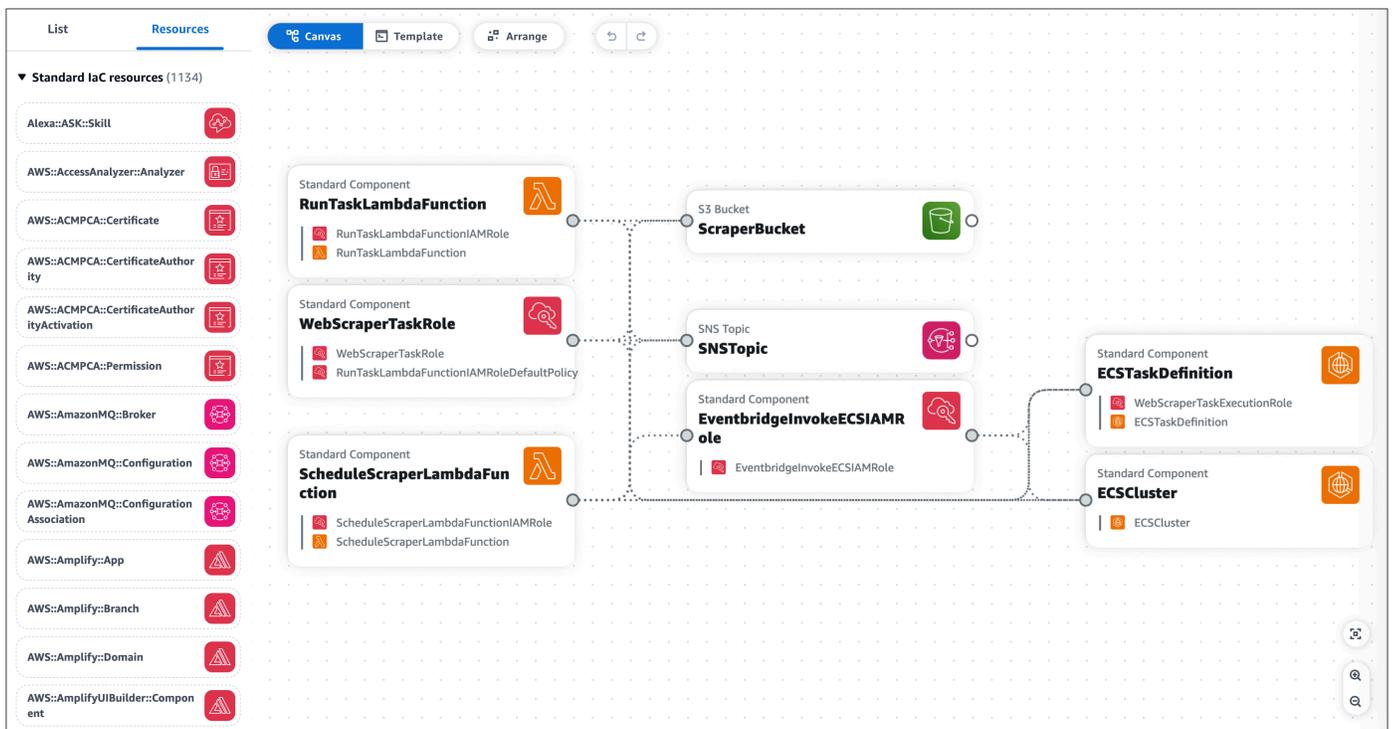
Configure la forma en que sus recursos interactúan entre sí conectándolos visualmente.

Especifique sus propiedades con más detalle a través de un panel de propiedades seleccionado.



### Trabaja con cualquier AWS CloudFormation recurso

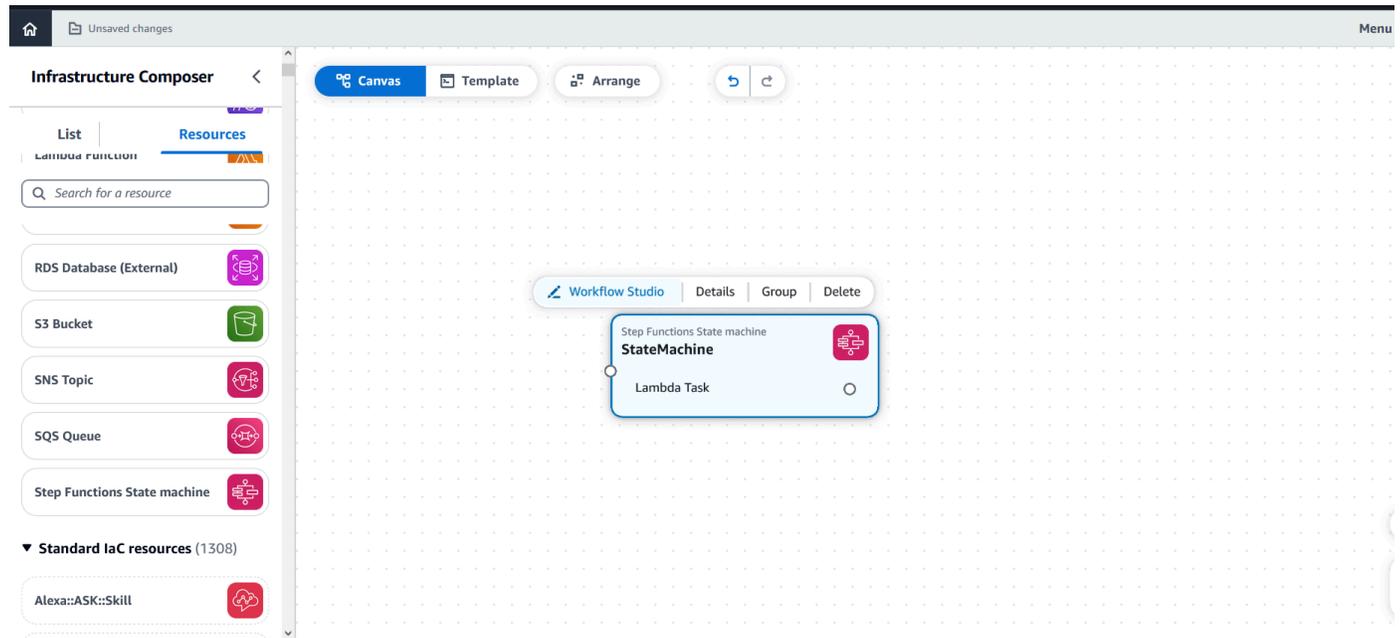
Arrastra cualquier AWS CloudFormation recurso al lienzo para crear la arquitectura de tu aplicación. Infrastructure Composer proporciona una plantilla de IaC inicial que puede usar para especificar las propiedades del recurso. Para obtener más información, consulte [Configurar y modificar tarjetas en Infrastructure Composer](#).



## Acceda a funciones adicionales con las funciones destacadas Servicios de AWS

Funciones de Infrastructure Composer Servicios de AWS que se suelen utilizar o configurar juntas al crear aplicaciones. Para obtener más información, consulte [Integre con Amazon VPC](#).

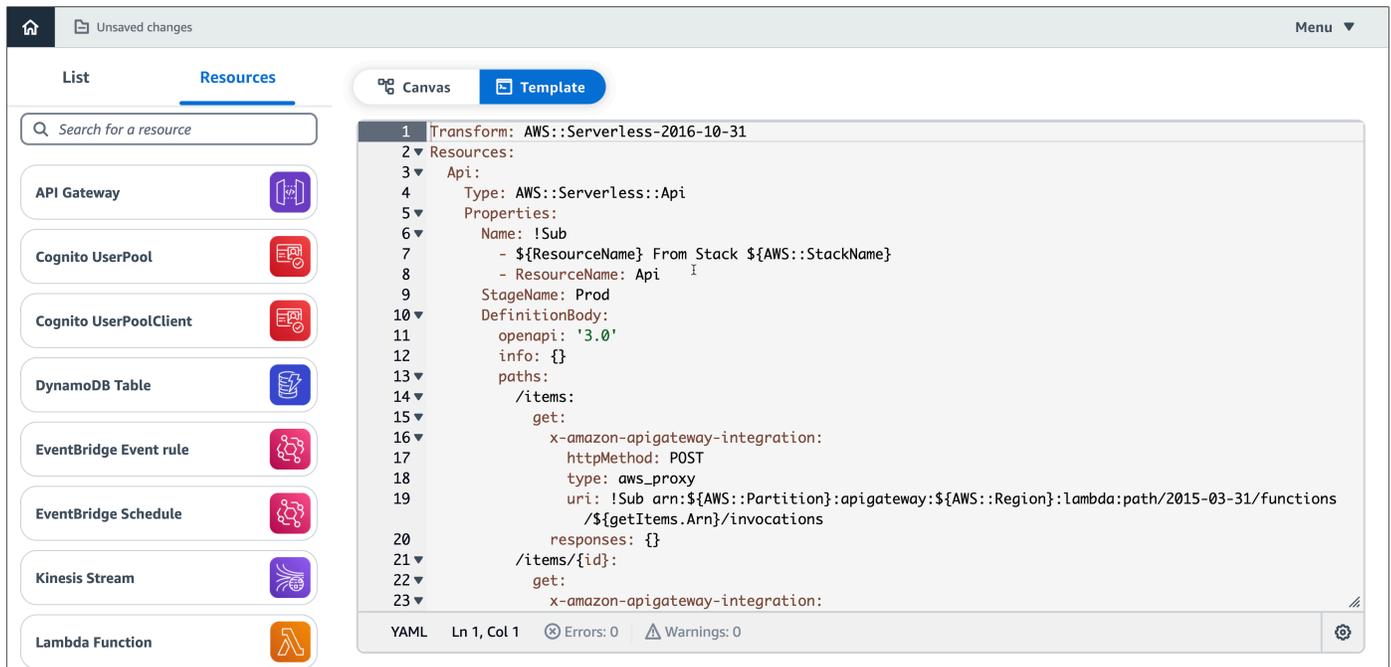
El siguiente es un ejemplo de la AWS Step Functions función, que proporciona una integración para lanzar Step Functions Workflow Studio directamente en el lienzo de Infrastructure Composer.



## Defina su infraestructura como plantillas de código (IaC)

Infrastructure Composer crea su código de infraestructura

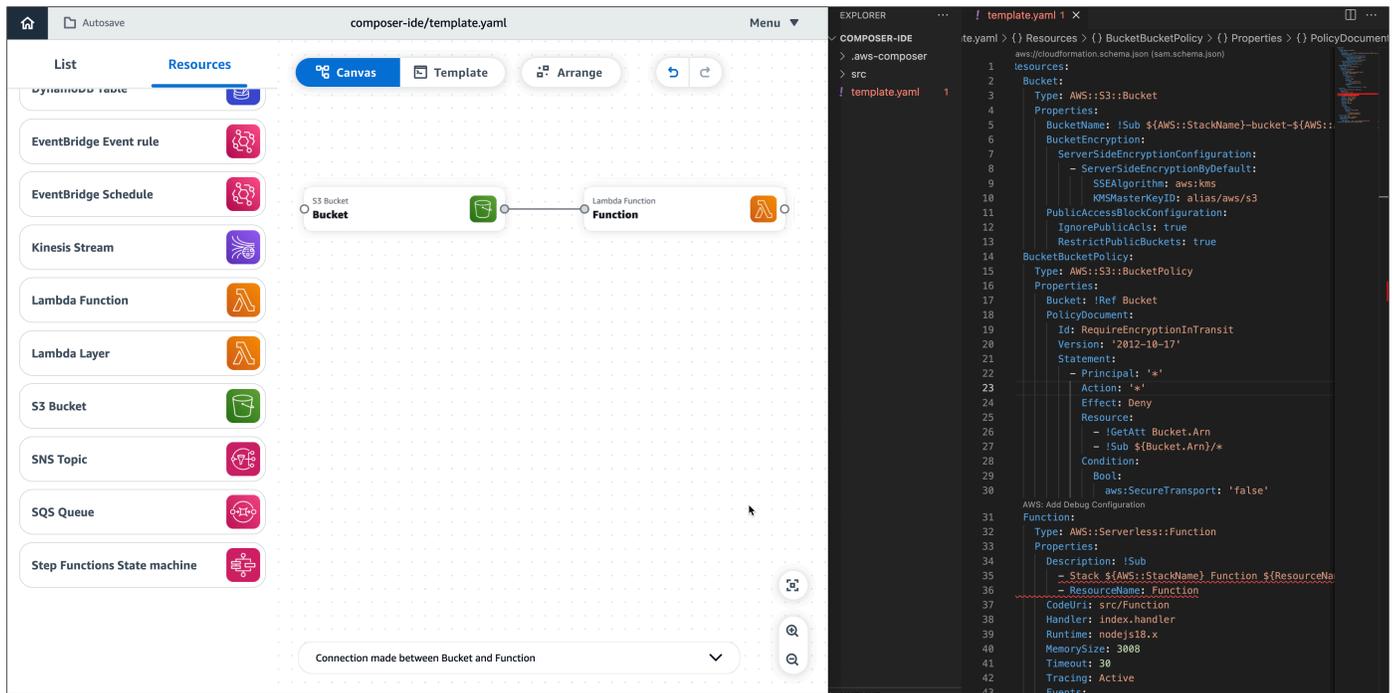
A medida que redacta, Infrastructure Composer crea automáticamente sus plantillas AWS CloudFormation and AWS Serverless Application Model (AWS SAM) siguiendo las prácticas AWS recomendadas. Puede ver y modificar las plantillas directamente desde Infrastructure Composer. Infrastructure Composer sincroniza automáticamente los cambios entre el lienzo visual y el código de la plantilla.



## Intégrelo con sus flujos de trabajo existentes

### Importe plantillas y proyectos existentes

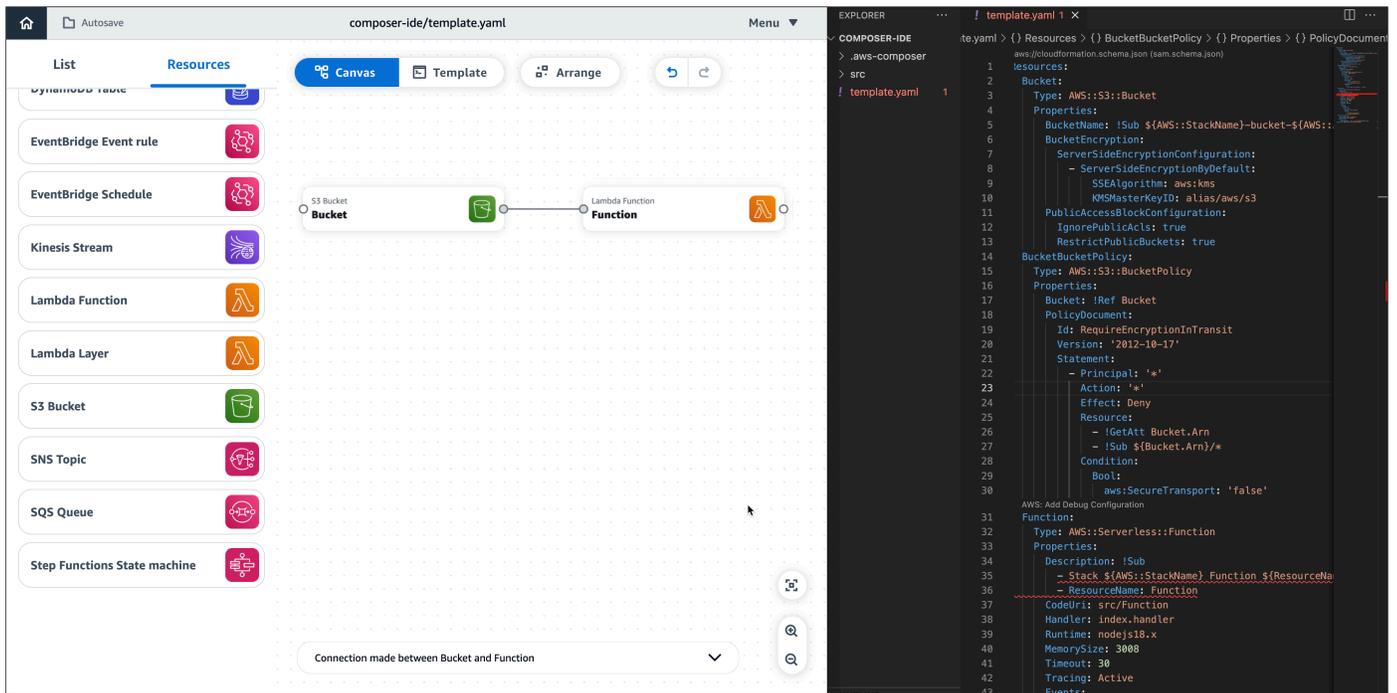
Importe AWS SAM plantillas AWS CloudFormation y plantillas existentes para visualizarlas y comprender mejor su diseño y modificarlas. Exporte las plantillas que cree en Infrastructure Composer e intégreelas en sus flujos de trabajo existentes para su implementación.



## Formas de acceder a Infrastructure Composer

### Desde la consola de Infrastructure Composer

Acceda a Infrastructure Composer a través de la consola de Infrastructure Composer para empezar rápidamente. Además, puede usar el modo de sincronización local para sincronizar y guardar Infrastructure Composer automáticamente con su máquina local.



## Desde la consola AWS CloudFormation

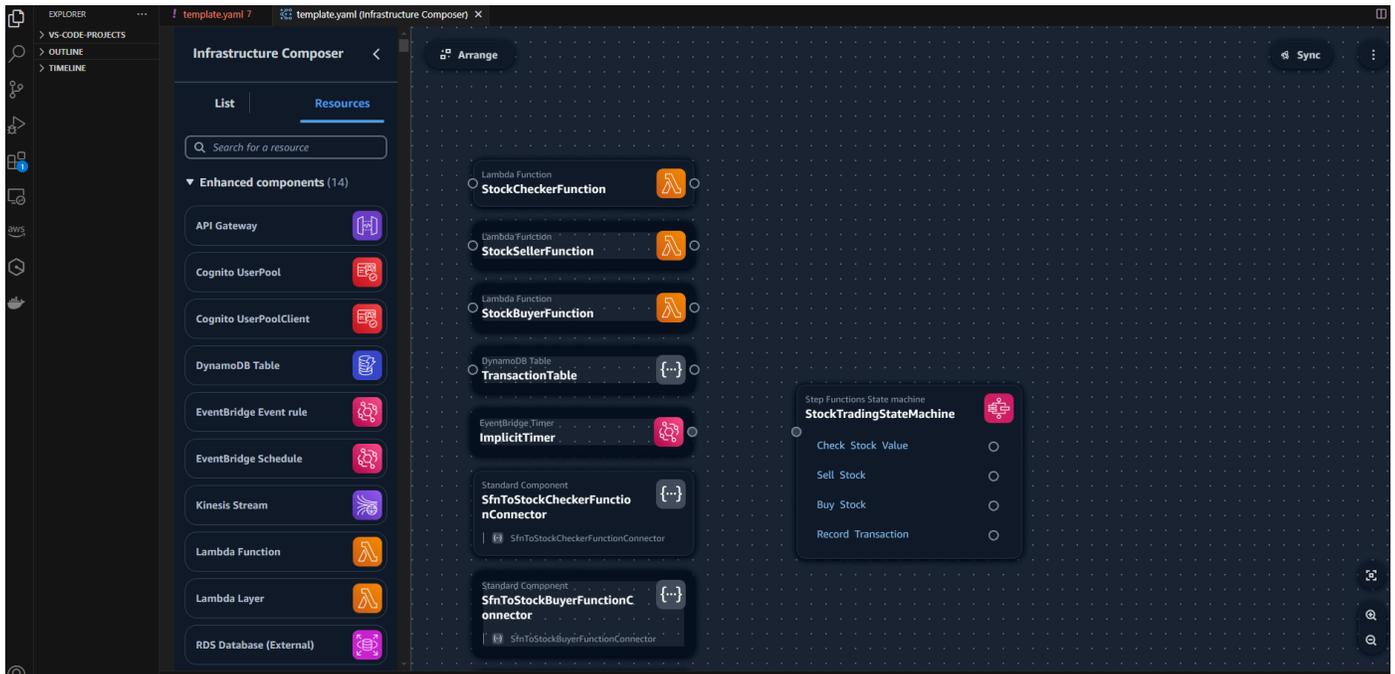
La consola Infrastructure Composer también es compatible con [el modo CloudFormation consola](#), una mejora con respecto a CloudFormation Designer, que se integra con el flujo de trabajo de la AWS CloudFormation pila. Esta nueva herramienta es ahora la herramienta recomendada para visualizar CloudFormation las plantillas.

## Desde la consola Lambda

Con Infrastructure Composer, también puede importar funciones de Lambda desde la consola de Lambda. Para obtener más información, consulte [Importación de funciones a Infrastructure Composer desde la consola Lambda](#).

## Desde AWS Toolkit for Visual Studio Code

Acceda a Infrastructure Composer a través de la extensión Toolkit for VS Code para llevar Infrastructure Composer a su entorno de desarrollo local.



## Más información

Para seguir aprendiendo sobre Infrastructure Composer, consulte los siguientes recursos:

- [Tarjetas Infrastructure Composer](#)
- [Redacte y cree aplicaciones sin servidor de forma visual | Serverless Office Hours](#): descripción general y demostración de Infrastructure Composer.

## Pasos a seguir a continuación

Para configurar Infrastructure Composer, consulte. [Introducción a la consola de Infrastructure Composer](#)

## Conceptos sin servidor para AWS Infrastructure Composer

Obtenga información sobre los conceptos básicos de la tecnología sin servidor antes de utilizarlos.  
AWS Infrastructure Composer

# Conceptos sobre sistemas sin servidor

## Arquitectura basada en eventos

Una aplicación sin servidor consta de AWS servicios individuales, como los de computación y Amazon DynamoDB AWS Lambda para la administración de bases de datos, cada uno de los cuales desempeña una función especializada. A continuación, estos servicios se integran de forma flexible entre sí mediante una arquitectura basada en eventos. Para obtener más información sobre la arquitectura basada en eventos, consulta [¿Qué es una arquitectura basada en eventos?](#)

## Infraestructura como código (IaC)

La infraestructura como código (IaC) es una forma de tratar la infraestructura de la misma manera que los desarrolladores tratan el código, aplicando el mismo rigor que el desarrollo del código de aplicaciones al aprovisionamiento de la infraestructura. Usted define su infraestructura en un archivo de plantilla, la implementa y AWS crea los recursos por usted. AWS ConIAC, usted define en el código lo que AWS quiere aprovisionar. Para obtener más información, consulte [La infraestructura como código](#) en la introducción a un DevOps AWS AWS documento técnico.

## Tecnologías sin servidor

Con las tecnologías AWS sin servidor, puede crear y ejecutar aplicaciones sin tener que administrar sus propios servidores. Toda la administración de los servidores se lleva a cabo de forma automática AWS, lo que proporciona numerosas ventajas, como el escalado automático y la alta disponibilidad integrada, lo que le permite llevar su idea a la producción rápidamente. Al utilizar tecnologías sin servidor, puede centrarse en el núcleo de su producto sin tener que preocuparse por la administración y el funcionamiento de los servidores. Para obtener más información sobre la tecnología sin servidor, consulte [Serverless on](#). AWS

Para obtener una introducción básica a los principales servicios AWS sin servidor, consulte [Serverless 101: Comprender los servicios sin servidor en Serverless Land](#).

# Tarjetas Infrastructure Composer

Infrastructure Composer simplifica el proceso de escribir la infraestructura como código (IaC) para los AWS CloudFormation recursos. [Para utilizar Infrastructure Composer de forma eficaz, hay dos conceptos básicos que debe entender primero: las tarjetas y las conexiones de tarjetas de Infrastructure Composer.](#)

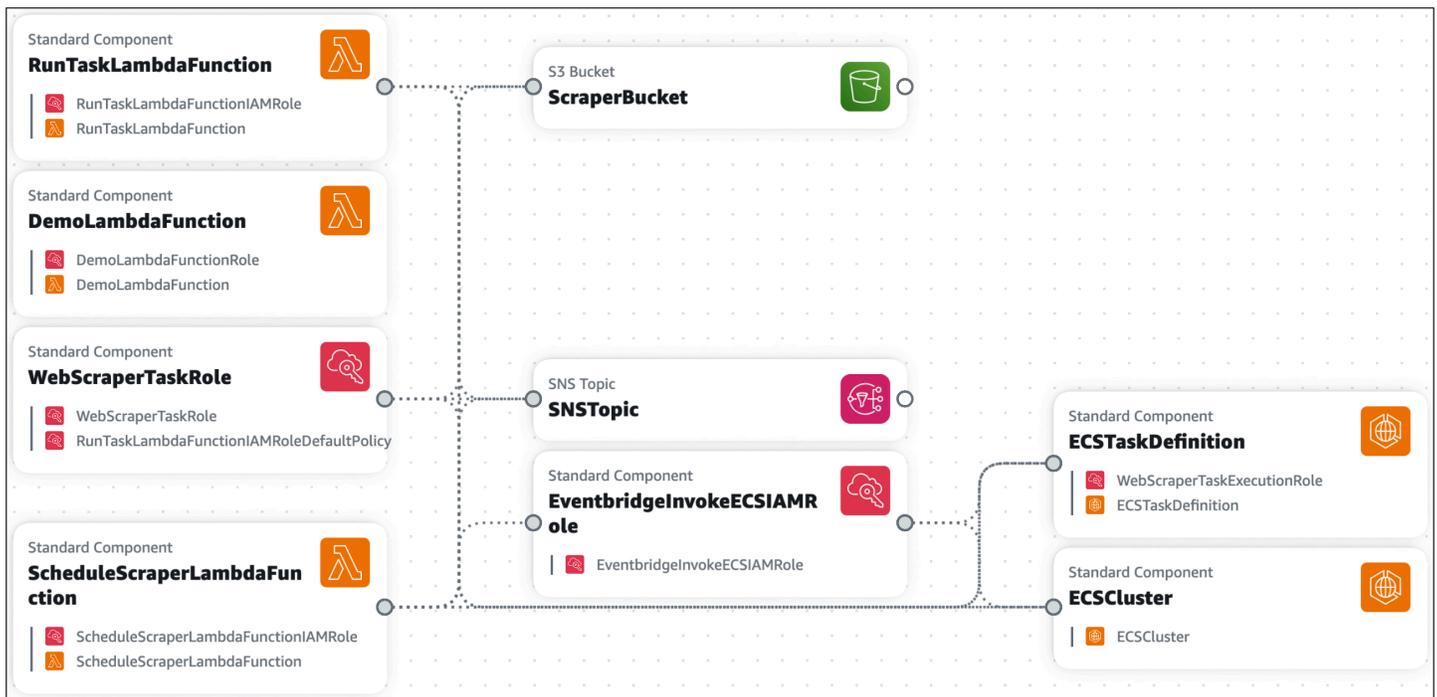
En Infrastructure Composer, las tarjetas representan AWS CloudFormation recursos. Existen dos categorías generales de tarjetas:

- [Tarjeta con componentes mejorada](#): colección de AWS CloudFormation recursos que se han combinado en una sola tarjeta seleccionada que mejora la facilidad de uso y la funcionalidad, y está diseñada para una amplia variedad de casos de uso. Las tarjetas con componentes mejorados son las primeras cartas que aparecen en la paleta de recursos de Infrastructure Composer.
- [Tarjeta de recursos iAC estándar](#): un único AWS CloudFormation recurso. Cada tarjeta de recursos IaC estándar, una vez arrastrada al lienzo, lleva la etiqueta Componente estándar y se puede combinar en varios recursos.

## Note

Dependiendo de la carta, una carta de recursos IaC estándar puede etiquetarse como carta de componente estándar después de arrastrarla al lienzo visual. Esto simplemente significa que la carta es una colección de una o más cartas de recursos IaC estándar.

Si bien algunos tipos de tarjetas están disponibles en la paleta Recursos, las tarjetas también pueden aparecer en el lienzo al importar una plantilla AWS CloudFormation o AWS Serverless Application Model (AWS SAM) existente a Infrastructure Composer. La siguiente imagen es un ejemplo de una aplicación importada que contiene varios tipos de tarjetas:



## Temas

- [Tarjetas de componentes mejoradas en Infrastructure Composer](#)
- [Tarjetas de componentes estándar en Infrastructure Composer](#)
- [Conexiones de tarjetas en Infrastructure Composer](#)

## Tarjetas de componentes mejoradas en Infrastructure Composer

Infrastructure Composer crea y administra las tarjetas de componentes mejoradas. Cada tarjeta contiene AWS CloudFormation recursos que se suelen utilizar juntos al crear aplicaciones AWS. Infrastructure Composer crea su código de infraestructura siguiendo las prácticas AWS recomendadas. Las tarjetas de componentes mejoradas son una excelente forma de empezar a diseñar su aplicación.

Las tarjetas de componentes mejorados están disponibles en la paleta de recursos, en la sección de componentes mejorados.

Las tarjetas de componentes mejoradas se pueden configurar y utilizar por completo en Infrastructure Composer para diseñar y crear aplicaciones sin servidor. Recomendamos utilizar tarjetas de componentes mejoradas al diseñar las aplicaciones sin código existente.

En esta tabla se muestran nuestros componentes mejorados con enlaces a la especificación de la plantilla AWS CloudFormation or AWS Serverless Application Model (AWS SAM) del recurso destacado de la tarjeta:

Tarjeta	Referencia
Amazon API Gateway	<a href="#">AWS:: Sin servidor:: API</a>
Amazon Cognito UserPool	<a href="#">AWS: :Cognito:: UserPool</a>
Amazon Cognito UserPoolClient	<a href="#">AWS: :Cognito:: UserPoolClient</a>
Tabla Amazon DynamoDB	<a href="#">AWS:: DynamoDB:: Tabla</a>
Regla de Amazon EventBridge Event	<a href="#">AWS: :Events: :Regla</a>
EventBridge Calendario	<a href="#">AWS: :Scheduler: :Horario</a>
Amazon Kinesis Stream	<a href="#">AWS: :Kinesis: :Stream</a>
AWS Lambda Función	<a href="#">AWS: :Serverless: :Función</a>
Capa Lambda	<a href="#">AWS:: Sin servidor:: LayerVersion</a>
Bucket de Amazon Simple Storage Service (Amazon S3)	<a href="#">AWS: :S3: :Bucket</a>
Tema de Amazon Simple Notification Service (AmazonSNS)	<a href="#">AWS::SNS: Tema</a>
Cola de Amazon Simple Queue Service (AmazonSQS)	<a href="#">AWS:: :Queue SQS</a>
AWS Step Functions Máquina de estados	<a href="#">AWS:: Sin servidor:: StateMachine</a>

## Ejemplo

El siguiente es un ejemplo de un componente mejorado de S3 Bucket:



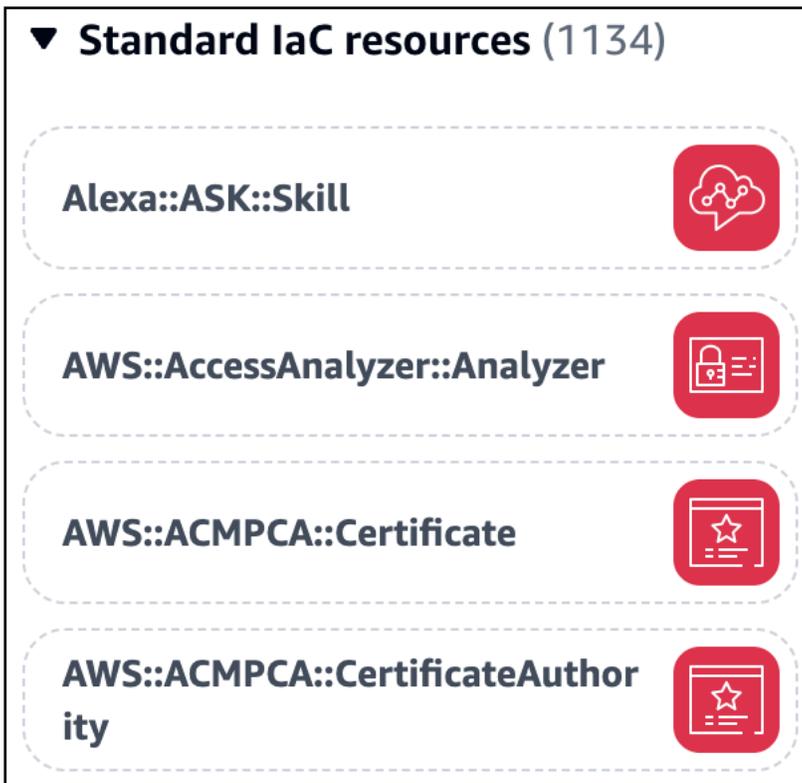
Al arrastrar una tarjeta de componentes de S3 Bucket al lienzo y ver la plantilla, verá que se han agregado AWS CloudFormation los dos recursos siguientes a la plantilla:

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

La tarjeta de componentes mejorada de S3 Bucket representa dos AWS CloudFormation recursos que son necesarios para que un bucket de Amazon Simple Storage Service (Amazon S3) interactúe con otros servicios de la aplicación.

## Tarjetas de componentes estándar en Infrastructure Composer

Antes de colocar una tarjeta de componente estándar en el lienzo visual de Infrastructure Composer, aparece como tarjeta de recursos estándar (IaC) en la paleta de recursos de Infrastructure Composer. Una tarjeta de recursos estándar (iAC) representa un único AWS CloudFormation recurso. Cada tarjeta de recursos IaC estándar, una vez colocada en el lienzo visual, se convierte en una tarjeta denominada Componente estándar y se puede combinar para representar varios AWS CloudFormation recursos.



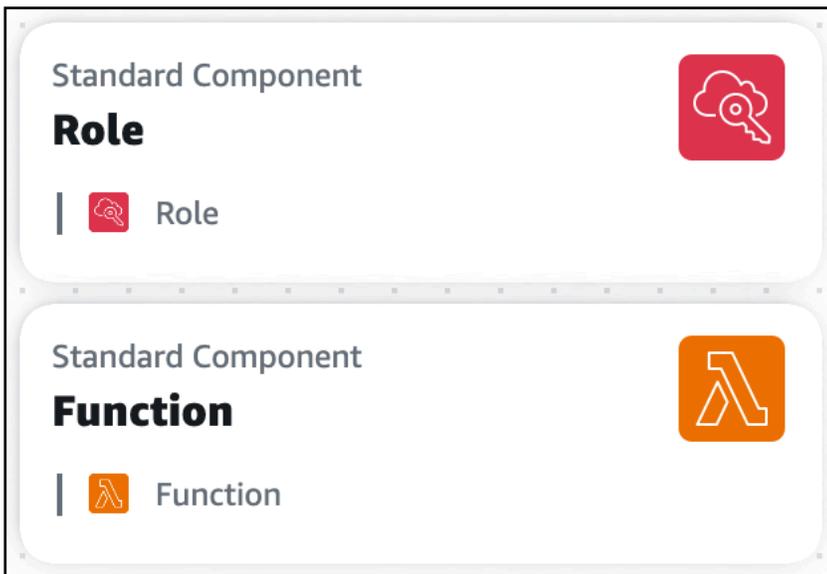
Cada tarjeta de recursos IaC estándar se puede identificar por su tipo de AWS CloudFormation recurso. El siguiente es un ejemplo de una tarjeta de recursos IaC estándar que representa un tipo de `AWS::ECS::Cluster` AWS CloudFormation recurso:



Cada tarjeta de componentes estándar visualiza los AWS CloudFormation recursos que contiene. El siguiente es un ejemplo de una tarjeta de componentes estándar que incluye dos recursos IaC estándar:



Al configurar las propiedades de las tarjetas de componentes estándar, Infrastructure Composer puede combinar tarjetas relacionadas. Por ejemplo, aquí hay dos tarjetas de componentes estándar:



En el panel de propiedades del recurso de la tarjeta de componente estándar que representa un `AWS::Lambda::Function` recurso, hacemos referencia a la función AWS Identity and Access Management (IAM) por su identificador lógico:

**Resource properties** ✕

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

**Editing** Function ▾

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

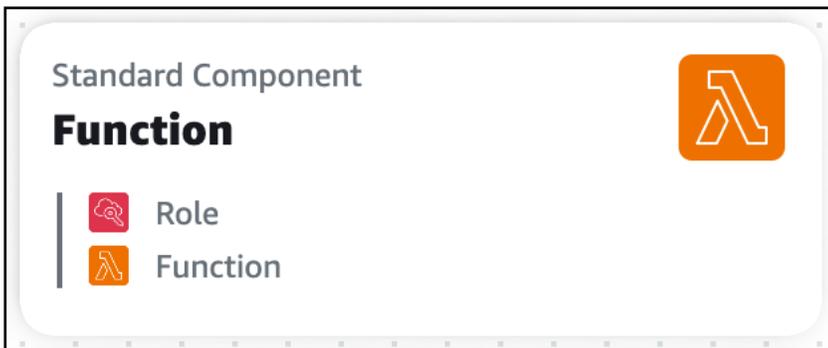
Function

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

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

[Resource reference](#) 

Tras guardar nuestra plantilla, las dos tarjetas de componentes estándar se combinan en una única tarjeta de componentes estándar.



## Conexiones de tarjetas en Infrastructure Composer

En AWS Infrastructure Composer, una conexión entre dos cartas se muestra visualmente mediante una línea. Estas líneas representan las relaciones impulsadas por eventos dentro de la aplicación.

Temas

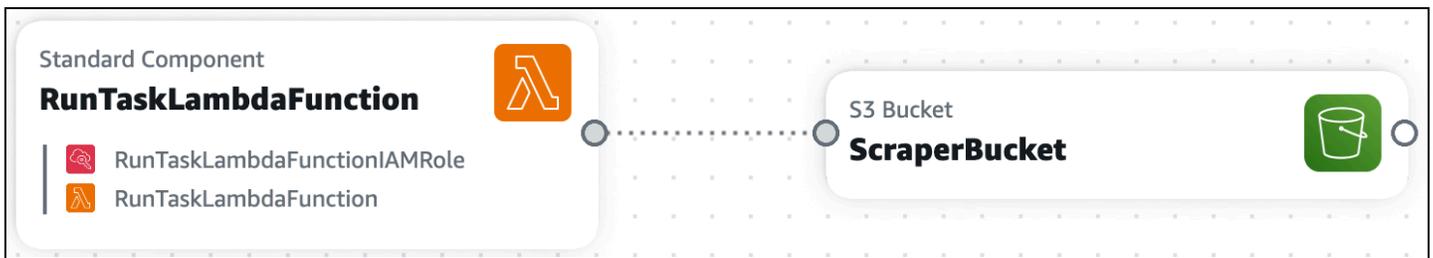
- [Conexiones entre tarjetas](#)
- [Conexiones entre tarjetas de componentes mejorados](#)
- [Conexiones hacia y desde tarjetas de recursos laC estándar](#)

### Conexiones entre tarjetas

La forma de conectar las tarjetas varía según el tipo de tarjeta. Cada tarjeta mejorada tiene al menos un puerto de conexión. Para conectarlas, basta con seleccionar un puerto conector y arrastrarlo hasta el puerto de otra tarjeta. Infrastructure Composer conectará los dos recursos o mostrará un mensaje que indica que esta configuración no es compatible.



Como se ha visto anteriormente, las líneas entre las tarjetas de componentes mejorados son sólidas. Por el contrario, las tarjetas de recursos iAC estándar (también denominadas tarjetas de componentes estándar) no tienen puertos de conexión. Para estas tarjetas, debe especificar estas relaciones basadas en eventos en la plantilla de la aplicación e Infrastructure Composer detectará automáticamente sus conexiones y las visualizará con una línea de puntos entre las tarjetas.

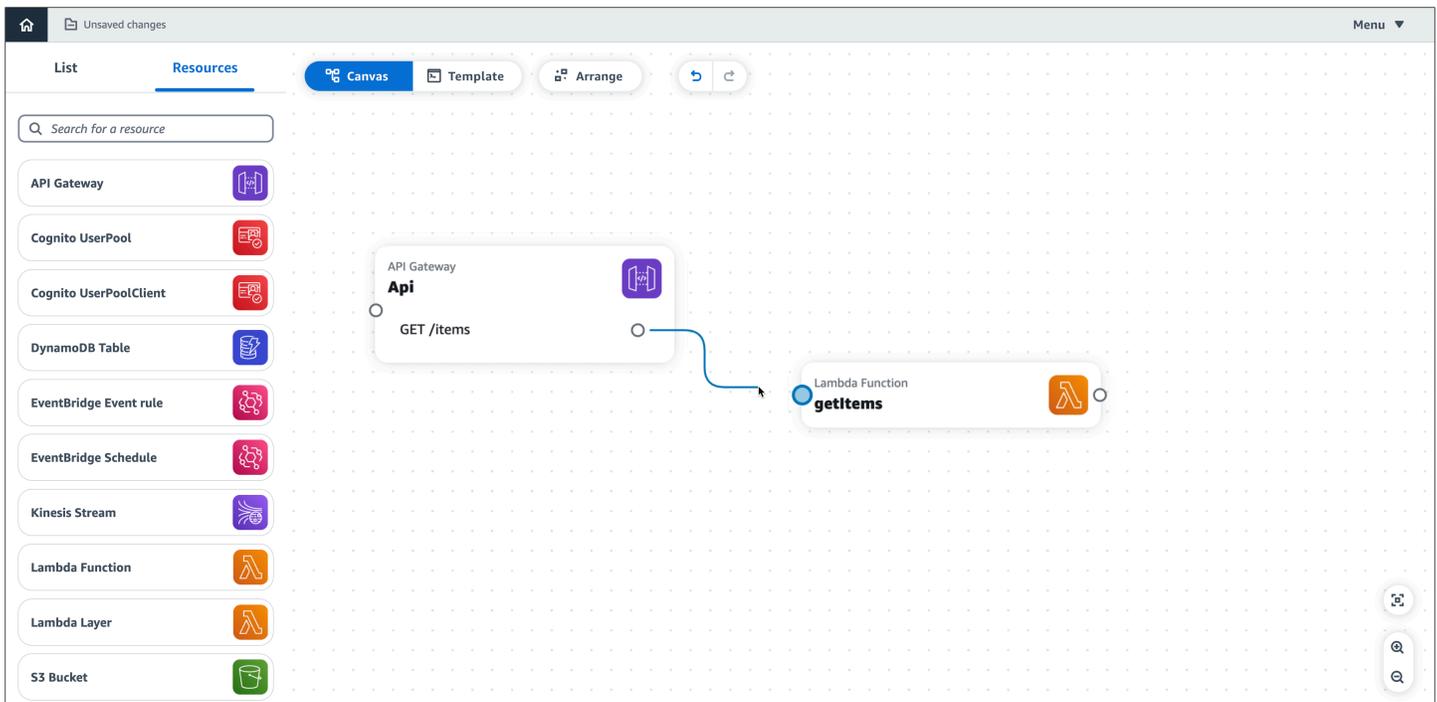


Para obtener más información, consulte las secciones siguientes.

## Conexiones entre tarjetas de componentes mejorados

En Infrastructure Composer, una conexión entre dos tarjetas de componentes mejorados se muestra visualmente mediante una línea continua. Estas líneas representan las relaciones impulsadas por eventos dentro de la aplicación.

Para conectar dos tarjetas, haga clic en un puerto de una tarjeta y arrástrelo hasta un puerto de otra tarjeta.



### Note

Las tarjetas de recursos iAC estándar no tienen puertos de conexión. En el caso de estas tarjetas, debe especificar sus relaciones basadas en eventos en la plantilla de la aplicación, e

Infrastructure Composer detectará automáticamente sus conexiones y las visualizará con una línea de puntos entre las tarjetas.

Para obtener más información, consulte [Connect tarjetas en el lienzo visual de Infrastructure Composer](#).

## ¿Qué componentes mejorados proporcionan las tarjetas

Las conexiones entre dos tarjetas, indicadas visualmente mediante una línea, proporcionan lo siguiente cuando es necesario:

- AWS Identity and Access Management (IAM) políticas
- Variables de entorno
- Eventos

### Políticas de IAM

Cuando un recurso necesita permiso para invocar otro recurso, Infrastructure Composer aprovisiona políticas basadas en recursos mediante plantillas de políticas AWS Serverless Application Model (AWS SAM).

- Para obtener más información sobre IAM los permisos y las políticas, consulte [Descripción general de la administración de acceso: permisos y políticas](#) en la IAM Guía del usuario.
- Para obtener más información sobre las plantillas AWS SAM de políticas, consulte [las plantillas de AWS SAM políticas](#) en la Guía para AWS Serverless Application Model desarrolladores.

### Variables de entorno

Las variables de entorno son valores temporales que se pueden cambiar para afectar al comportamiento de los recursos. Cuando es necesario, Infrastructure Composer define el código de infraestructura para utilizar las variables de entorno entre los recursos.

### Eventos

Los recursos pueden invocar otro recurso a través de diferentes tipos de eventos. Cuando es necesario, Infrastructure Composer define el código de infraestructura necesario para que los recursos interactúen a través de los tipos de eventos.

## Conexiones hacia y desde tarjetas de recursos IaC estándar

Todos los AWS CloudFormation recursos están disponibles para su uso como tarjetas de recursos estándar de IaC en la paleta de recursos. Al arrastrar una tarjeta de recursos de iAC estándar al lienzo, una tarjeta de recursos de iAC estándar se convierte en una tarjeta de componentes estándar, lo que hace que Infrastructure Composer cree una plantilla inicial para el recurso en la aplicación.

Para obtener más información, consulte [Tarjetas estándar en Infrastructure Composer](#).

# Introducción a la consola de Infrastructure Composer

Utilice los temas de esta sección para configurar AWS Infrastructure Composer y aprender a diseñar una aplicación utilizando su lienzo visual. El recorrido y los tutoriales de esta sección se muestran en la consola de Infrastructure Composer, que es la experiencia de usuario predeterminada. En los temas de esta sección se muestra cómo completar los requisitos previos para usar Infrastructure Composer, usar la consola de Infrastructure Composer, cargar y modificar un proyecto y crear su primera aplicación.

Infrastructure Composer también está disponible desde AWS Toolkit for Visual Studio Code y en modo CloudFormation consola. Las experiencias entre las herramientas son generalmente las mismas, pero hay algunas diferencias entre cada una. Para obtener información detallada sobre el uso de Infrastructure Composer en cada una de estas herramientas, consulte [Dónde puede usar Infrastructure Composer](#).

## Temas

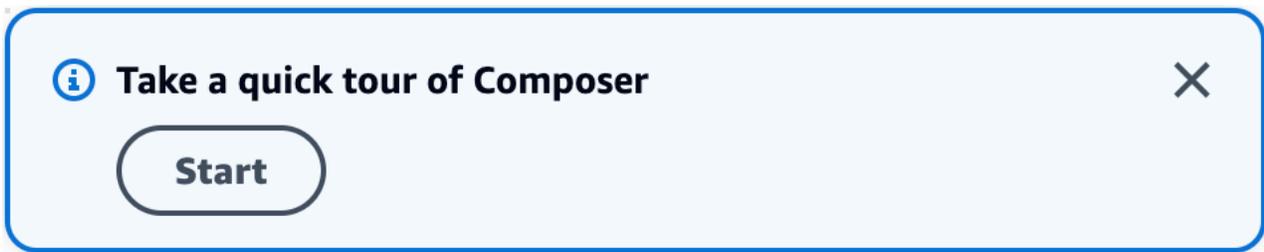
- [Haga un recorrido por la consola de Infrastructure Composer](#)
- [Cargue y modifique el proyecto de demostración de Infrastructure Composer](#)
- [Cree su primera aplicación con Infrastructure Composer](#)

## Haga un recorrido por la consola de Infrastructure Composer

Para hacerse una idea general de cómo AWS Infrastructure Composer funciona, realice el recorrido que viene integrado en la consola de Infrastructure Composer. Para obtener una descripción general de la consola de Infrastructure Composer, consulte [Haga un recorrido por la consola de Infrastructure Composer](#). Para obtener instrucciones detalladas sobre el uso de Infrastructure Composer, consulte [Cómo componer en AWS Infrastructure Composer](#).

Para hacer un recorrido por Infrastructure Composer

1. Inicie sesión en la [consola de Infrastructure Composer](#).
2. En la página de inicio, elija Abrir demostración.
3. En la esquina superior derecha, en la ventana Haga un recorrido rápido por Composer, seleccione Iniciar.



4. En la ventana de recorrido de Composer, haga lo siguiente:

- Para pasar al siguiente paso, selecciona Siguiente.
- Para volver al paso anterior, selecciona Anterior.
- En el último paso, para finalizar el recorrido, selecciona Finalizar.

El recorrido ofrece una breve descripción general de las funciones básicas de Infrastructure Composer, como el uso, la configuración y la conexión de tarjetas. Para obtener más información, consulta [Cómo componer en AWS Infrastructure Composer](#).

## Pasos a seguir a continuación

Para cargar y modificar un proyecto en Infrastructure Composer, consulte. [Cargue y modifique el proyecto de demostración de Infrastructure Composer](#)

## Cargue y modifique el proyecto de demostración de Infrastructure Composer

Utilice este tutorial para familiarizarse con la interfaz de usuario de Infrastructure Composer y aprender a cargar, modificar y guardar el proyecto de demostración de Infrastructure Composer.

Este tutorial se realiza en la consola de Infrastructure Composer. Una vez completado, estará listo para empezar [Cree su primera aplicación con Infrastructure Composer](#).

### Temas

- [Paso 1: Abre la demostración](#)
- [Paso 2: Explore el lienzo visual de Infrastructure Composer](#)
- [Paso 3: Amplíe la arquitectura de su aplicación](#)
- [Paso 4: Guarde la aplicación](#)
- [Pasos a seguir a continuación](#)

## Paso 1: Abre la demostración

Comience a usar Infrastructure Composer creando un proyecto de demostración.

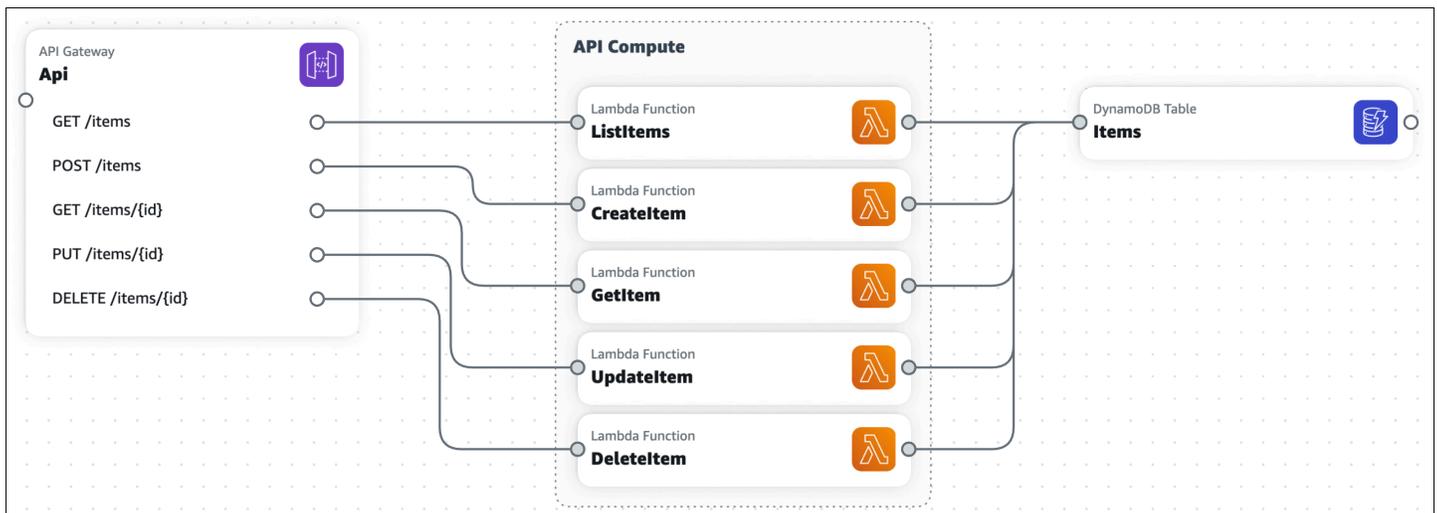
Para crear un proyecto de demostración

1. Inicie sesión en la [consola de Infrastructure Composer](#).
2. En la página de inicio, elija Abrir demostración.

La aplicación de demostración es una aplicación básica sin servidor para crear, leer, eliminar y actualizar (CRUD) que incluye:

- Un recurso de Amazon API Gateway con cinco rutas.
- Cinco AWS Lambda funciones.
- Una tabla de Amazon DynamoDB.

La siguiente imagen es de la demostración:

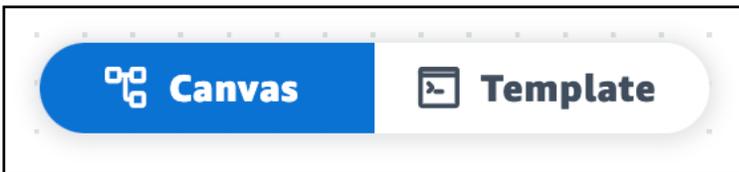


## Paso 2: Explore el lienzo visual de Infrastructure Composer

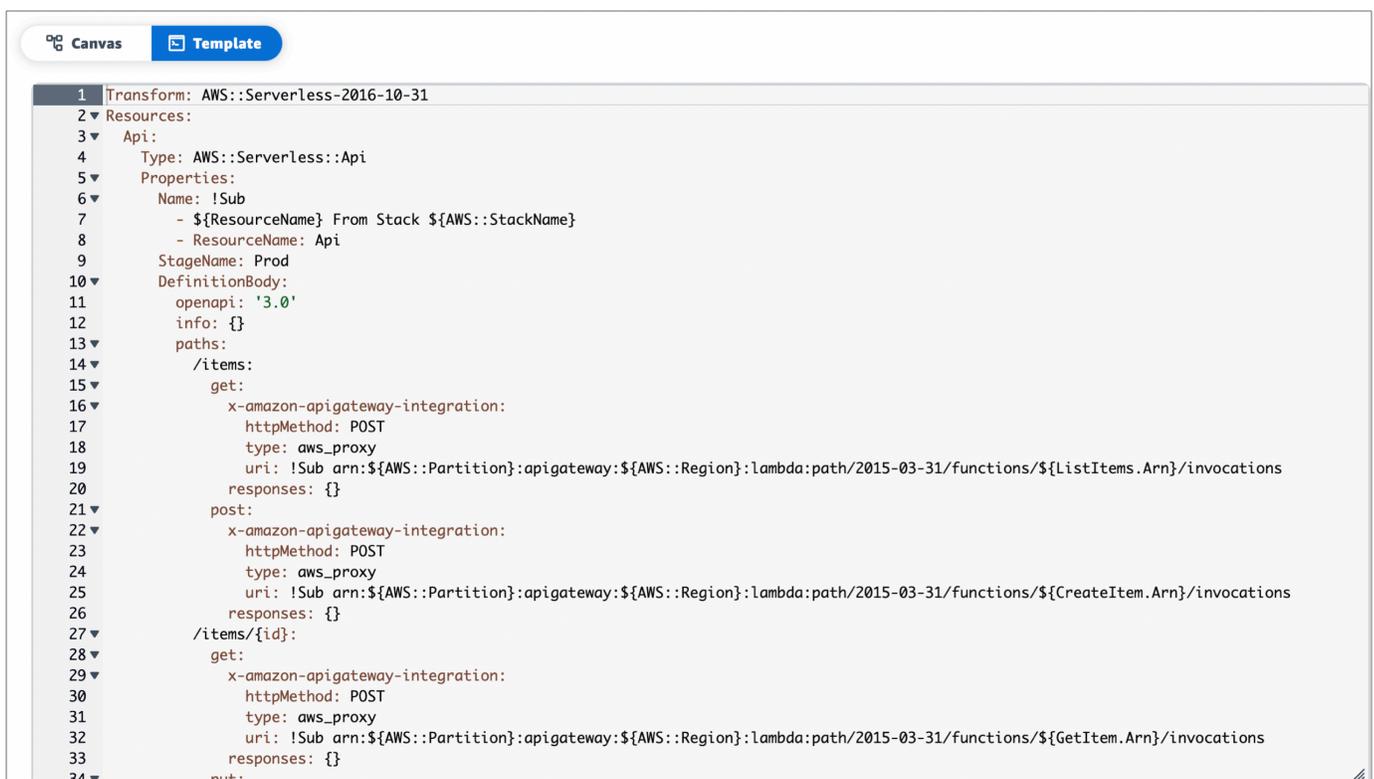
Conozca las características del lienzo visual para desarrollar su proyecto de demostración de Infrastructure Composer. Para obtener una descripción general del diseño del lienzo visual, consulte [Descripción visual](#).

## Para explorar las características del lienzo visual

1. Al abrir un proyecto de aplicación nuevo o existente, Infrastructure Composer carga la vista del lienzo, tal y como se indica sobre el área de la vista principal.

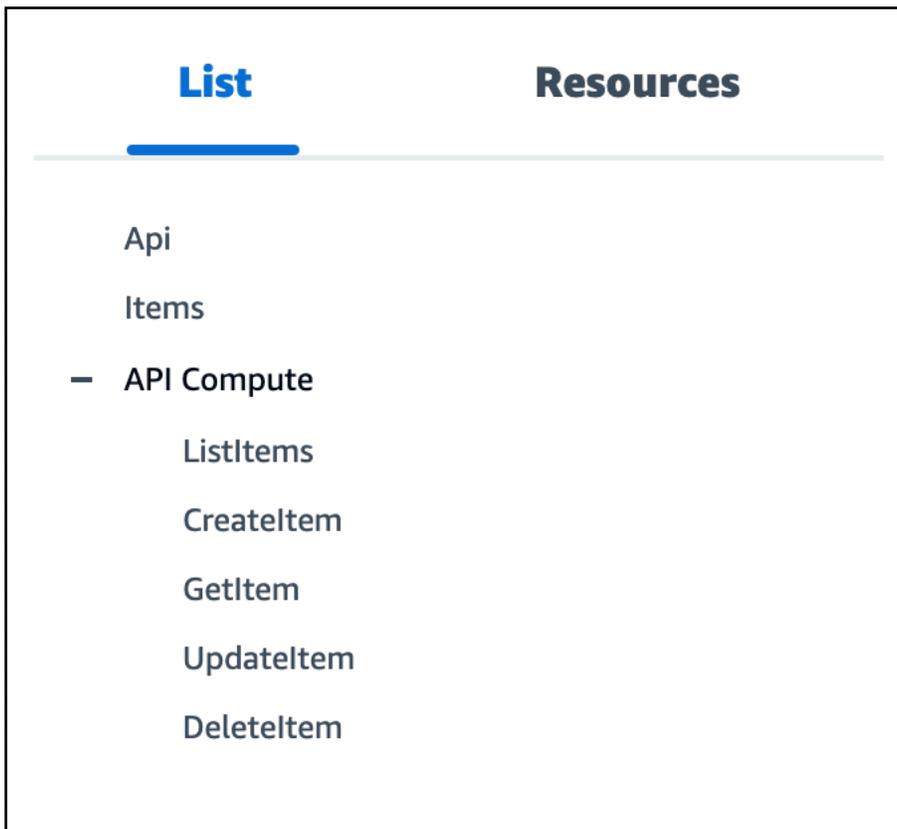


Para mostrar el código de infraestructura de la aplicación en el área de visualización principal, elija Plantilla. Por ejemplo, esta es la vista de plantilla AWS Serverless Application Model (AWS SAM) del proyecto de demostración de Infrastructure Composer.

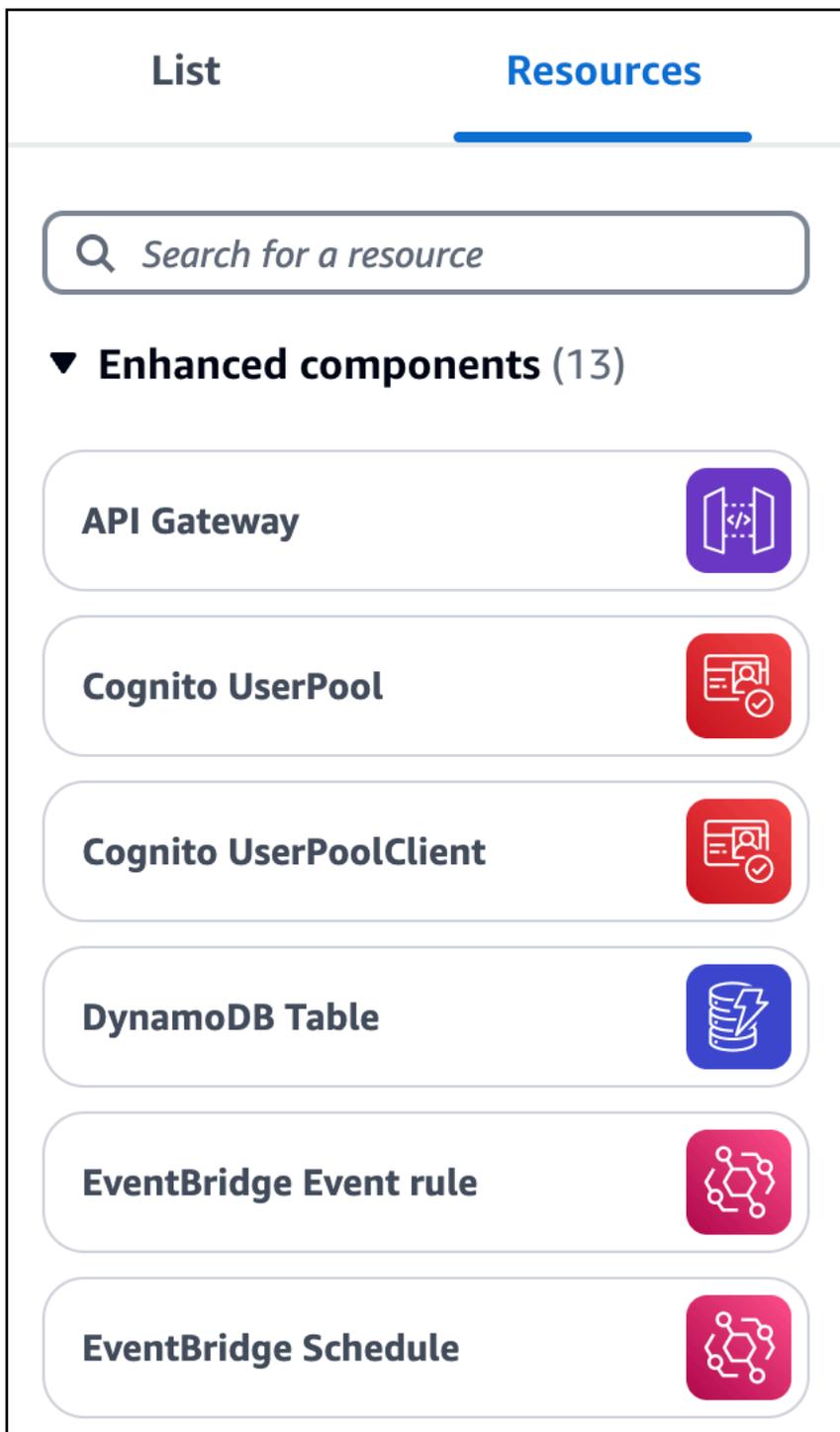
A screenshot of the Infrastructure Composer interface showing the 'Template' view. The interface has two tabs at the top: 'Canvas' and 'Template', with 'Template' selected. Below the tabs is a code editor displaying the AWS SAM JSON code for a serverless application. The code is as follows:

```
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 volver a mostrar la vista de lienzo de su aplicación, elija Canvas.
3. Para mostrar los recursos de la aplicación organizados en una vista de árbol, elija Lista.



4. Para mostrar la paleta de recursos, elija Recursos. Esta paleta incluye tarjetas que puede utilizar para ampliar la arquitectura de la aplicación. Puede buscar tarjetas o desplazarse por la lista.



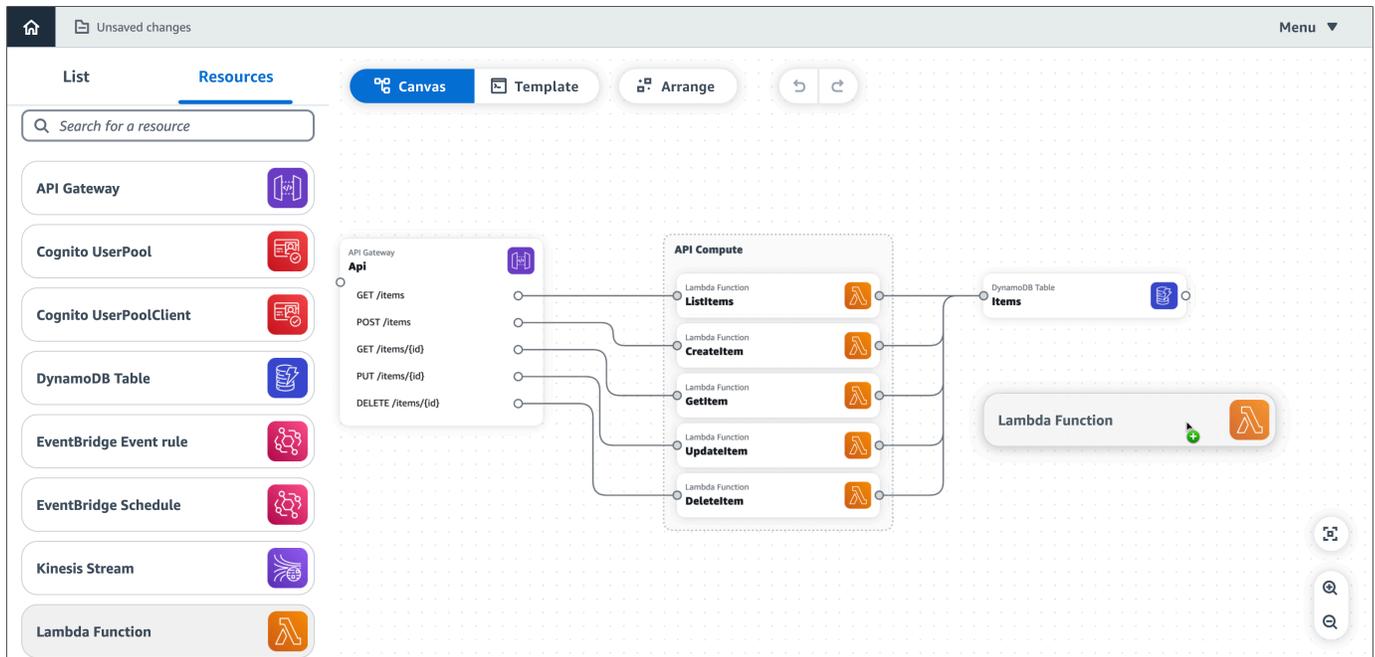
5. Para moverse por el lienzo visual, utilice gestos básicos. Para obtener más información, consulte [Coloque las tarjetas en el lienzo](#).

## Paso 3: Amplíe la arquitectura de su aplicación

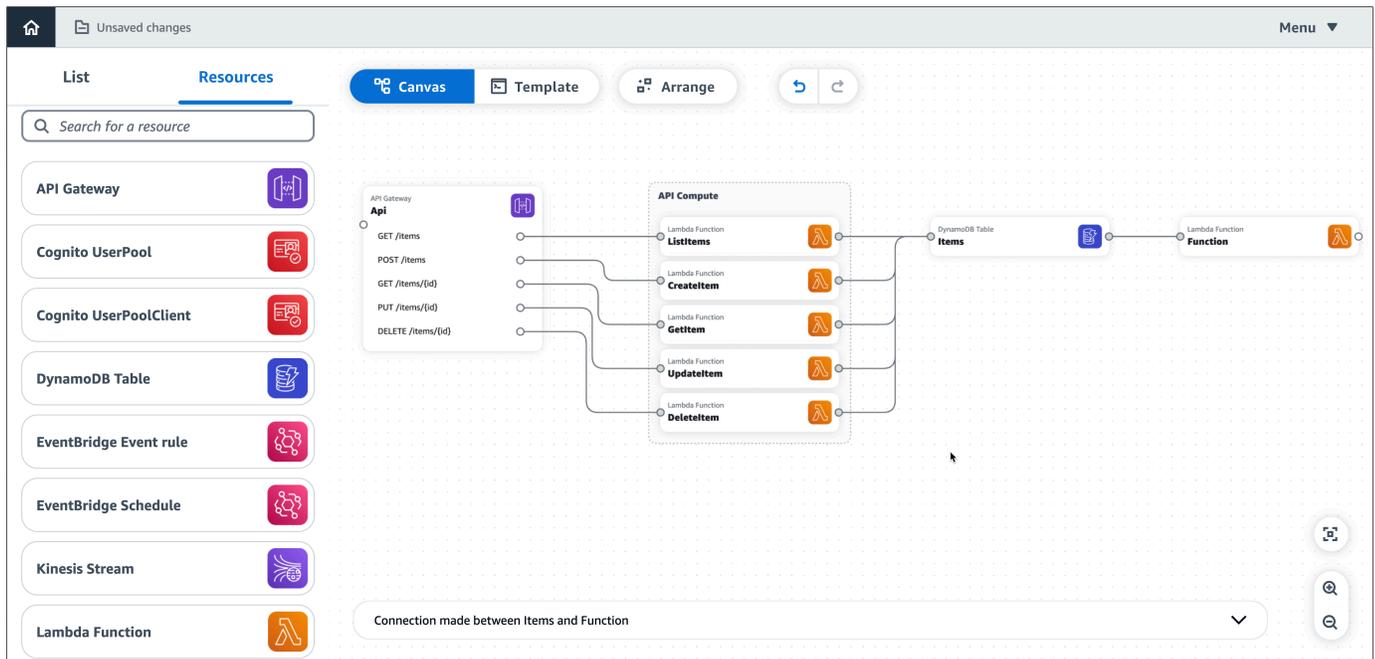
En este paso, ampliará la arquitectura de la aplicación añadiendo una función Lambda a la tabla de DynamoDB.

Para añadir una función Lambda a la tabla de DynamoDB

1. Desde la paleta de recursos (Recursos), arrastre la tarjeta del componente mejorado de la función Lambda al lienzo, a la derecha de la tarjeta de tabla de DynamoDB.



2. Connect la tabla de DynamoDB a la función Lambda. Para conectarlos, haga clic en el puerto derecho de la tarjeta de tabla de DynamoDB y arrástrelo hasta el puerto izquierdo de la tarjeta de función Lambda.
3. Seleccione Organizar para organizar las tarjetas en la vista de lienzo.



4. Configure la función Lambda. Para configurarla, realice una de las siguientes acciones:
  - En la vista de lienzo, modifique las propiedades de la función en el panel de propiedades del recurso. Para abrir el panel, haga doble clic en la tarjeta de Función Lambda. O bien, seleccione la tarjeta y, a continuación, elija Detalles. Para obtener más información sobre las propiedades de la función Lambda configurables que se muestran en el panel de propiedades del recurso, consulte la Guía para [AWS Lambda desarrolladores](#).
  - En la vista de plantilla, modifique el código de la función (`AWS::Serverless::Function`). Infrastructure Composer sincroniza automáticamente los cambios con el lienzo. Para obtener más información sobre el recurso de función de una AWS SAM plantilla, consulte [AWS::Serverless::Function](#) en la referencia de AWS SAM recursos y propiedades.

## Paso 4: Guarde la aplicación

Guarde la aplicación guardando manualmente la plantilla de la aplicación en su máquina local o activando la sincronización local.

Para guardar manualmente la plantilla de la aplicación

1. En el menú, seleccione Guardar > Guardar archivo de plantilla.
2. Proporcione un nombre para la plantilla y elija una ubicación en su máquina local para guardar la plantilla. Presiona Guardar.

Para obtener instrucciones sobre cómo activar la sincronización local, consulte [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#).

## Pasos a seguir a continuación

Para empezar a crear su primera aplicación, consulte [Cree su primera aplicación con Infrastructure Composer](#).

## Cree su primera aplicación con Infrastructure Composer

En este tutorial, se utiliza AWS Infrastructure Composer para crear, leer, actualizar y eliminar (CRUD) una aplicación sin servidor que gestione los usuarios de una base de datos.

Para este tutorial, utilizamos Infrastructure Composer en AWS Management Console. Le recomendamos que utilice Google Chrome o Microsoft Edge, y una ventana de navegador a pantalla completa.

 ¿Es la primera vez que utiliza la tecnología sin servidor?

Se recomienda tener conocimientos básicos de los siguientes temas:

- [Arquitectura basada en eventos](#)
- [Infraestructura como código \(IaC\)](#)
- [Tecnologías sin servidor](#)

Para obtener más información, consulte [Conceptos sin servidor para AWS Infrastructure Composer](#).

### Temas

- [Referencia de propiedades del recurso](#)
- [Paso 1: Crea tu proyecto](#)
- [Paso 2: Añade cartas al lienzo](#)
- [Paso 3: Configure su API puerta de enlace REST API](#)
- [Paso 4: Configure las funciones de Lambda](#)

- [Paso 5: Conecta tus tarjetas](#)
- [Paso 6: Organice el lienzo](#)
- [Paso 7: Agregar y conectar una tabla de DynamoDB](#)
- [Paso 8: Revise la plantilla AWS CloudFormation](#)
- [Paso 9: Intégrelo en sus flujos de trabajo de desarrollo](#)
- [Pasos a seguir a continuación](#)

## Referencia de propiedades del recurso

Al crear la aplicación, utilice esta tabla como referencia para configurar las propiedades de Amazon API Gateway y sus AWS Lambda recursos.

Método	Ruta	Nombre de la función
GET	/artículos	getItems
GET	/elementos/ {id}	getItem
PUT	/items/ {id}	updateItem
POST	/artículo	addItem
DELETE	/items/ {id}	deleteItem

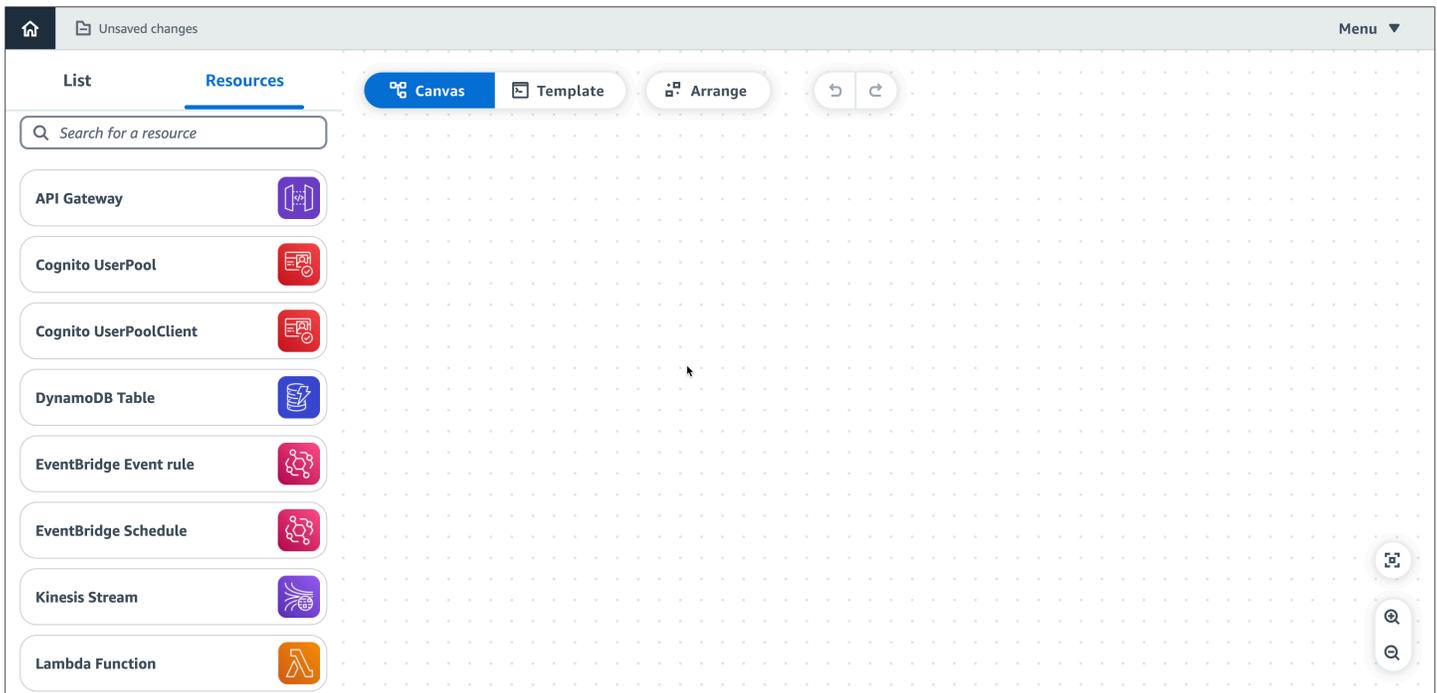
## Paso 1: Crea tu proyecto

Para empezar a utilizar su aplicación CRUD sin servidor, cree un proyecto nuevo en Infrastructure Composer y active la sincronización local.

Para crear un nuevo proyecto en blanco

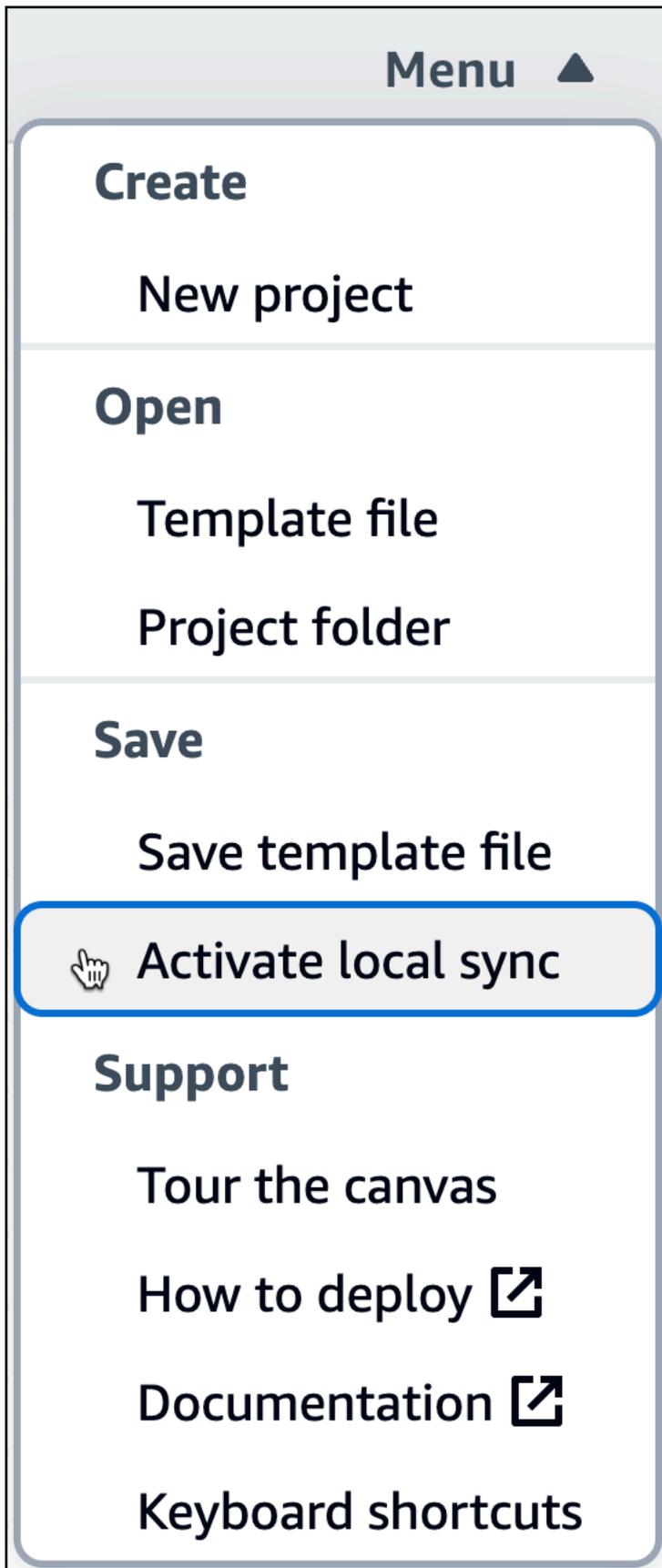
1. Inicie sesión en la [consola de Infrastructure Composer](#).
2. En la página de inicio, elija Crear proyecto.

Como se muestra en la imagen siguiente, Infrastructure Composer abre el lienzo visual y carga una plantilla de aplicación inicial (en blanco).



Para activar la sincronización local

1. En el menú de Infrastructure Composer, seleccione Guardar > Activar la sincronización local.



2. Para la ubicación del proyecto, pulse **Seleccionar carpeta** y elija un directorio. Aquí es donde Infrastructure Composer guardará y sincronizará los archivos y carpetas de la plantilla a medida que vaya diseñando.

La ubicación del proyecto no debe contener una plantilla de aplicación existente.

 **Note**

La sincronización local requiere un navegador compatible con el acceso al sistema de archivosAPI. Para obtener más información, consulte [Data Infrastructure Composer obtiene acceso a](#).

3. Cuando se te pida que permitas el acceso, selecciona **Ver archivos**.
4. Presiona **Activar** para activar la sincronización local. Cuando se te pida que guardes los cambios, selecciona **Guardar cambios**.

Cuando esté activado, el indicador de guardado automático se mostrará en el área superior izquierda del lienzo.

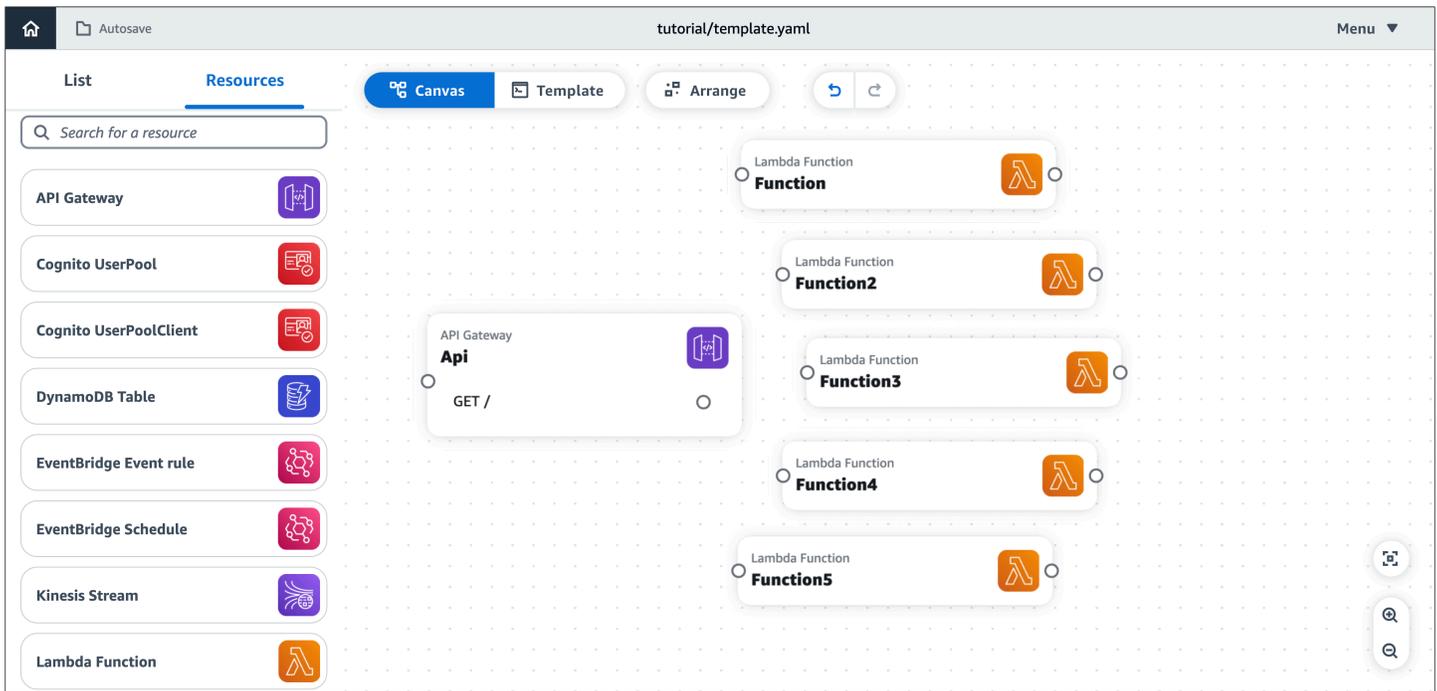
## Paso 2: Añade cartas al lienzo

Comience a diseñar la arquitectura de su aplicación con tarjetas de componentes mejoradas, empezando por una API puerta de enlace REST API y cinco funciones Lambda.

Para añadir tarjetas API Gateway y Lambda al lienzo

En la paleta Recursos, en la sección Componentes mejorados, haga lo siguiente:

1. Arrastra una carta APIGateway al lienzo.
2. Arrastre una tarjeta de Función Lambda al lienzo. Repita el proceso hasta que haya agregado cinco tarjetas de Función Lambda al lienzo.



## Paso 3: Configure su API puerta de enlace REST API

A continuación, añada cinco rutas en su tarjeta API Gateway.

Para añadir rutas a la tarjeta API Gateway

1. Abra el panel de propiedades del recurso de la tarjeta APIGateway. Para abrir el panel, haga doble clic en la tarjeta. O bien, selecciona la tarjeta y, a continuación, selecciona Detalles.
2. En el panel de propiedades del recurso, en Rutas, haga lo siguiente:

### Note

Para cada una de las siguientes rutas, utilice los valores de HTTP método y ruta especificados en la [tabla de referencia de propiedades del recurso](#).

- a. En Método, elija el HTTP método especificado. Por ejemplo, GET.
  - b. En Ruta, introduzca la ruta especificada. Por ejemplo, **/items**.
  - c. Seleccione Añadir ruta.
  - d. Repita los pasos anteriores hasta que haya agregado las cinco rutas especificadas.
3. Seleccione Guardar.

The screenshot displays the AWS Infrastructure Composer interface for a template named 'tutorial/template.yaml'. On the left, a 'Resources' list includes API Gateway, Cognito UserPool, Cognito UserPoolClient, DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, and Lambda Function. The central canvas shows an API Gateway resource named 'Api' with five routes: GET /items, GET /items/{id}, PUT /items/{id}, POST /item, and DELETE /items/{id}. Each route is connected to a corresponding Lambda Function (Function1 to Function5). The right-hand 'Resource properties' panel is open for a selected route, showing the Method (GET, PUT, POST) and Path (/items/{id}) configuration, along with a 'Remove route' button.

## Paso 4: Configure las funciones de Lambda

Asigne un nombre a cada una de las cinco funciones Lambda tal y como se especifica en la tabla de [referencia de propiedades del recurso](#).

Para asignar un nombre a las funciones Lambda

1. Abra el panel de propiedades del recurso de una tarjeta de función Lambda. Para abrir el panel, haga doble clic en la tarjeta. O bien, selecciona la tarjeta y, a continuación, selecciona Detalles.
2. En el panel de propiedades del recurso, en Logical ID, introduzca el nombre de una función específica. Por ejemplo, **getItems**.
3. Seleccione Guardar.
4. Repita los pasos anteriores hasta que haya asignado un nombre a las cinco funciones.

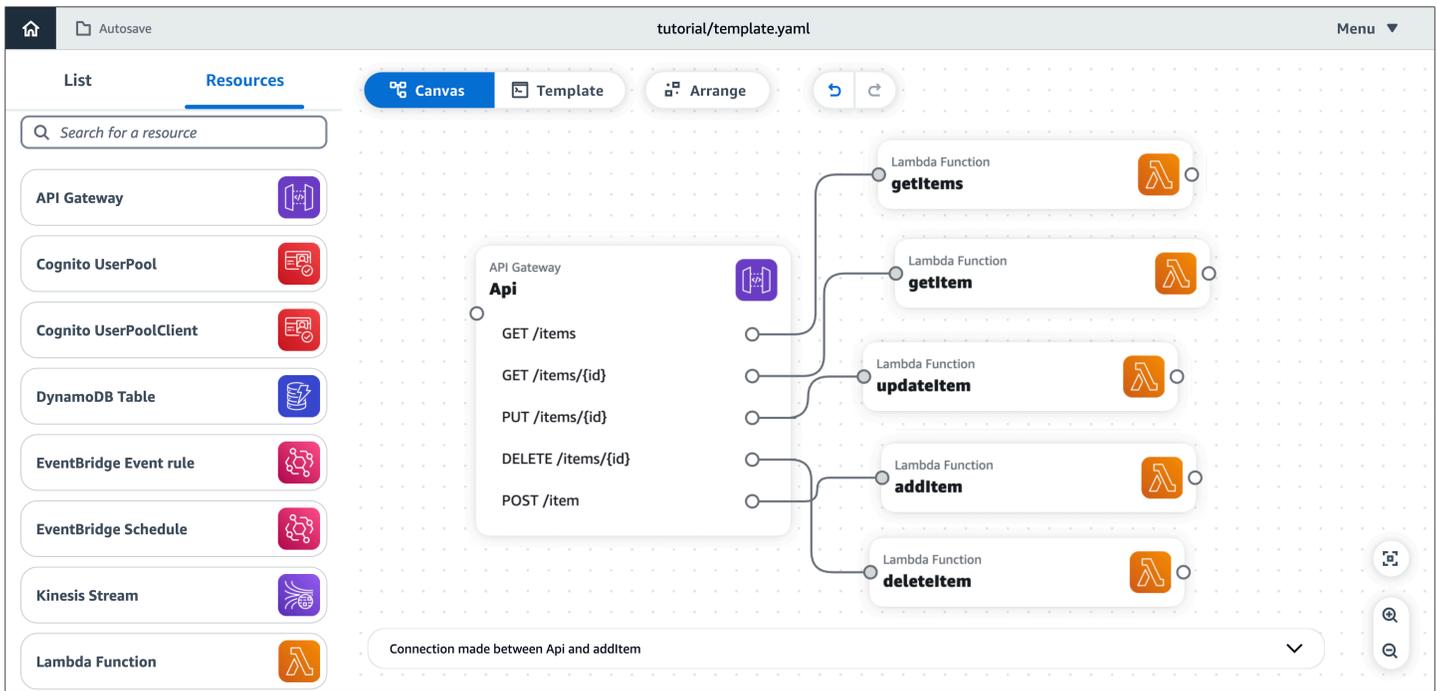
The screenshot displays the AWS Infrastructure Composer interface for a template named 'tutorial/template.yaml'. On the left, a 'Resources' panel lists various AWS services. The central canvas shows an 'API Gateway' resource named 'Api' with five endpoints. These endpoints are connected to five 'Lambda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

## Paso 5: Conecta tus tarjetas

Conecte cada ruta de la tarjeta APIGateway a su tarjeta de función Lambda relacionada, tal y como se especifica en la tabla de [referencia de propiedades del recurso](#).

Para conectar sus tarjetas

1. Haga clic en un puerto derecho de la tarjeta APIGateway y arrástrelo hasta el puerto izquierdo de la tarjeta de función Lambda especificada. Por ejemplo, haga clic en el puerto GET/items y arrástrelo hasta el puerto izquierdo de. getItems
2. Repita el paso anterior hasta conectar las cinco rutas de la tarjeta APIGateway a las tarjetas de función Lambda correspondientes.



## Paso 6: Organice el lienzo

Organice el lienzo visual agrupando las funciones de Lambda y organizando todas las tarjetas.

Para agrupar sus funciones

1. Mantenga pulsada la tecla Mayús y, a continuación, seleccione cada tarjeta de Función Lambda del lienzo.
2. Seleccione Agrupar.

Para asignar un nombre a su grupo

1. Haga doble clic en la parte superior del grupo, cerca del nombre del grupo (Grupo).

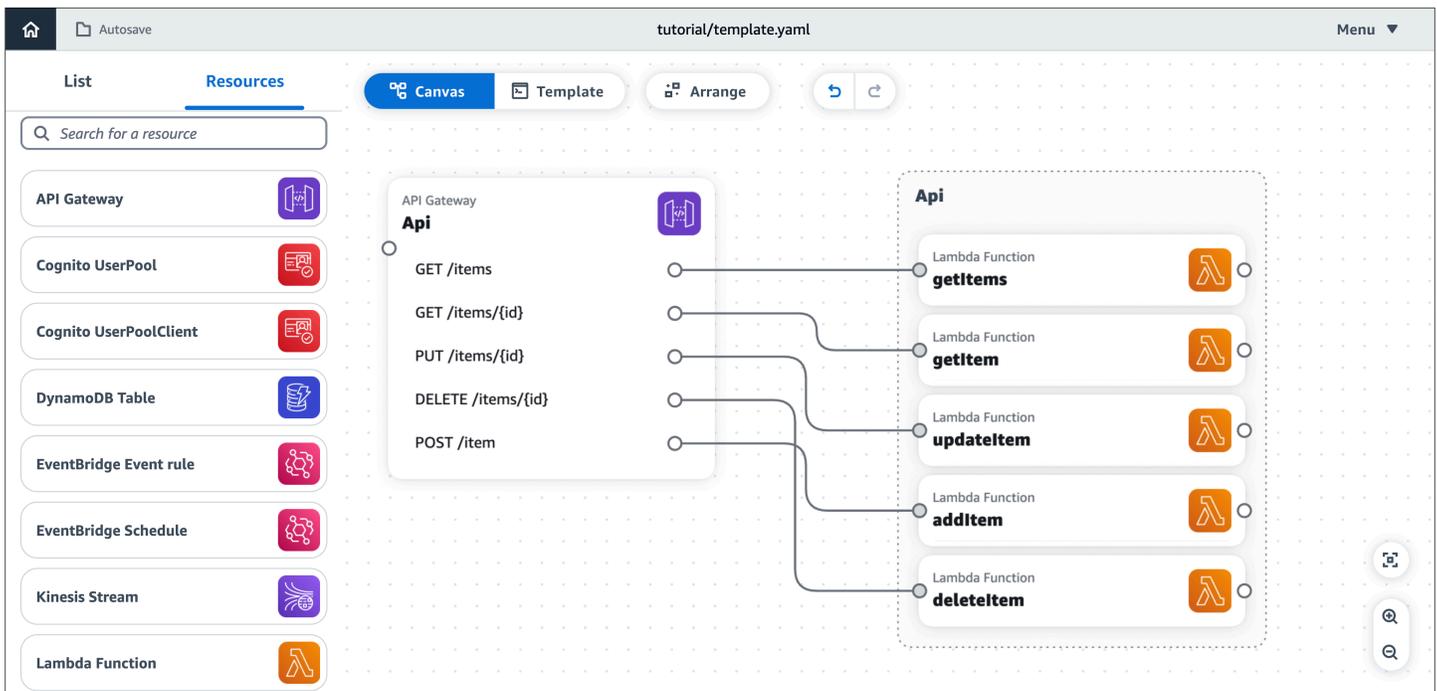
Se abre el panel de propiedades del grupo.

2. En el panel de propiedades del grupo, introduzca el nombre del grupo **API**.
3. Seleccione Guardar.

Para organizar las tarjetas

En el lienzo, sobre el área de visualización principal, selecciona Organizar.

Infrastructure Composer organiza y alinea todas las cartas en el lienzo visual, incluido el nuevo grupo (API), como se muestra a continuación:

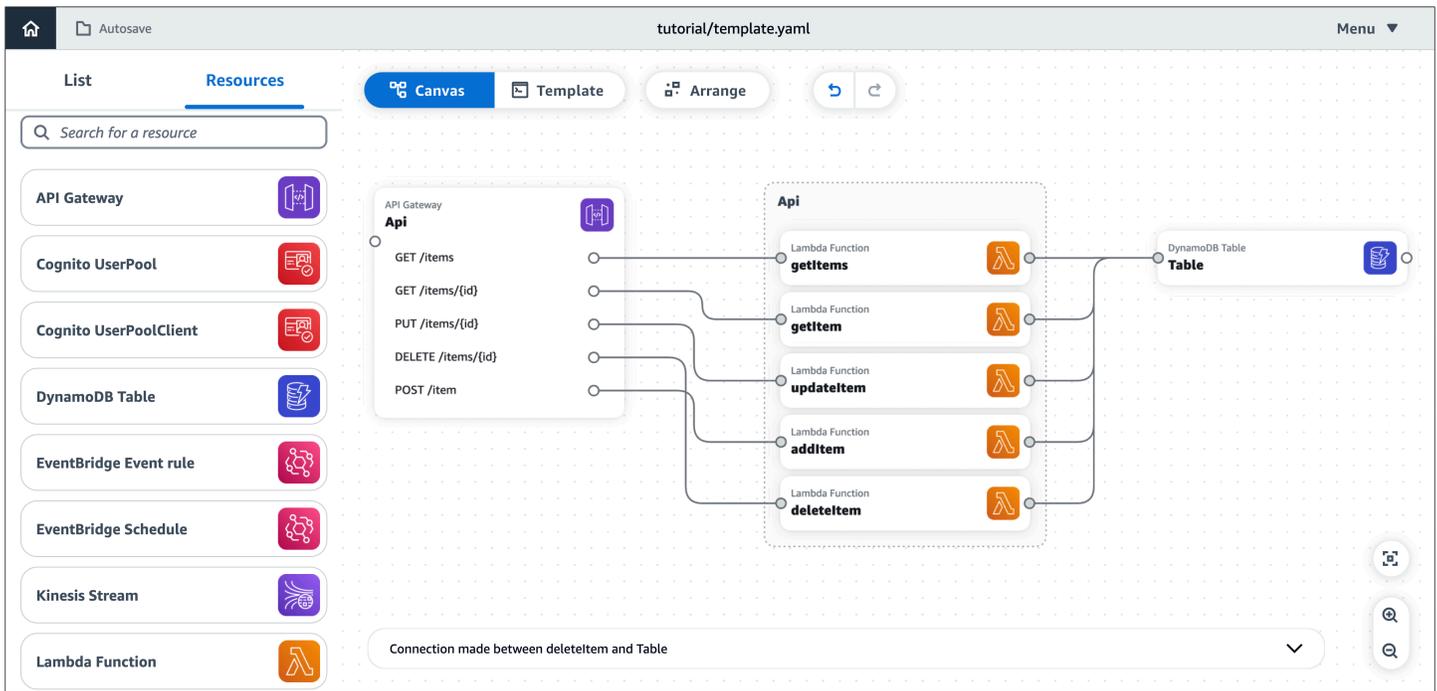


## Paso 7: Agregar y conectar una tabla de DynamoDB

Ahora, añada una tabla de DynamoDB a la arquitectura de la aplicación y conéctela a las funciones de Lambda.

Para añadir y conectar una tabla de DynamoDB

1. Desde la paleta de recursos (Recursos), en la sección Componentes mejorados, arrastre una tarjeta de DynamoDB Table al lienzo.
2. Haga clic en el puerto derecho de una tarjeta de función Lambda y arrástrelo hasta el puerto izquierdo de la tarjeta de tabla de DynamoDB.
3. Repita el paso anterior hasta conectar las cinco tarjetas de función Lambda a la tarjeta de tabla de DynamoDB.
4. (Opcional) Para reorganizar y realinear las cartas en el lienzo, elija Organizar.



## Paso 8: Revise la plantilla AWS CloudFormation

¡Enhorabuena! Diseñó correctamente una aplicación sin servidor que está lista para su implementación. Por último, elija Plantilla para revisar la AWS CloudFormation plantilla que Infrastructure Composer ha generado automáticamente para usted.

En la plantilla, Infrastructure Composer ha definido lo siguiente:

- La Transform declaración, que especifica la plantilla como una plantilla AWS Serverless Application Model (AWS SAM). Para obtener más información, consulta la [anatomía AWS SAM de la plantilla](#) en la Guía para AWS Serverless Application Model desarrolladores.
- Un `AWS::Serverless::Api` recurso que especifica su API puerta de enlace REST API con sus cinco rutas.
- Cinco `AWS::Serverless::Function` recursos, que especifican las configuraciones de las funciones de Lambda, incluidas sus variables de entorno y políticas de permisos.
- Un `AWS::DynamoDB::Table` recurso que especifica la tabla de DynamoDB y sus propiedades.
- La Metadata sección, que contiene información sobre el grupo de recursos () API. Para obtener más información sobre esta sección, consulte [Metadatos](#) en la Guía del AWS CloudFormation usuario.

The screenshot shows the AWS Infrastructure Composer IDE interface. On the left, there is a 'List' sidebar with various AWS services like API Gateway, Cognito UserPool, and Lambda Function. The main area is titled 'tutorial/template.yaml' and shows a YAML configuration for an API Gateway resource. The configuration includes details like 'Type: AWS::Serverless::Api', 'Name: !Sub', and 'DefinitionBody' with an 'openapi' version of '3.0' and a 'paths' section for a GET endpoint. The status bar at the bottom indicates 'YAML Ln 1, Col 1' with 0 errors and 0 warnings.

```

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:${AWS::Region}:lambda:path/2015-03-31/functions
20                /${getItems.Arn}/invocations
21            responses: {}
22        /items/{id}:
23          get:
24            x-amazon-apigateway-integration:

```

## Paso 9: Intégrelo en sus flujos de trabajo de desarrollo

Utilice el archivo de plantilla y los directorios de proyectos que creó Infrastructure Composer para realizar más pruebas e implementar.

- Con la sincronización local, puede conectar Infrastructure Composer IDE a su máquina local para acelerar el desarrollo. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su consola local IDE](#).
- Con la sincronización local, puede utilizar la interfaz de línea de AWS Serverless Application Model comandos (AWS SAM CLI) de su máquina local para probar e implementar la aplicación. Para obtener más información, consulte [Implemente su aplicación sin servidor Infrastructure Composer en la nube AWS](#).

## Pasos a seguir a continuación

Ahora está listo para crear sus propias aplicaciones con Infrastructure Composer. Para obtener información detallada sobre el uso de Infrastructure Composer, consulte [Cómo componer en AWS Infrastructure Composer](#). Cuando esté listo para implementar la aplicación, consulte [Implemente su aplicación sin servidor Infrastructure Composer en la nube AWS](#).

# Dónde puede usar Infrastructure Composer

Puede usar Infrastructure Composer desde su consola AWS Toolkit for Visual Studio Code, desde Infrastructure Composer y en Infrastructure Composer en modo CloudFormation consola. Si bien cada uno varía según los casos de uso ligeramente diferentes, en general se trata de experiencias similares. En esta sección se proporcionan detalles de cada experiencia.

El tema [Uso de la AWS Infrastructure Composer consola](#) es una descripción general completa de la experiencia de consola predeterminada. En este tema se [CloudFormation modo consola](#) proporcionan detalles sobre una versión de Infrastructure Composer que está integrada con el flujo de trabajo de la AWS CloudFormation pila. [AWS Toolkit for Visual Studio Code](#) proporciona información sobre el acceso y el uso de Infrastructure Composer en VS Code.

## Temas

- [Uso de la AWS Infrastructure Composer consola](#)
- [Uso de Infrastructure Composer en modo CloudFormation consola](#)
- [Uso de Infrastructure Composer desde AWS Toolkit for Visual Studio Code](#)

# Uso de la AWS Infrastructure Composer consola

En esta sección se proporcionan detalles sobre el acceso y el uso AWS Infrastructure Composer desde la consola de Infrastructure Composer. Esta es la experiencia predeterminada de Infrastructure Composer y es una buena forma de familiarizarse con Infrastructure Composer. También puede integrar la consola de Infrastructure Composer con su consola local IDE. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su consola local IDE](#).

También puede [acceder a Infrastructure Composer desde el AWS kit de herramientas de VS Code](#) y puede usar un [modo de Infrastructure Composer diseñado específicamente para su uso en AWS CloudFormation](#).

Para obtener documentación general sobre el uso de Infrastructure Composer, consulte [¿Cómo componer](#).

## Temas

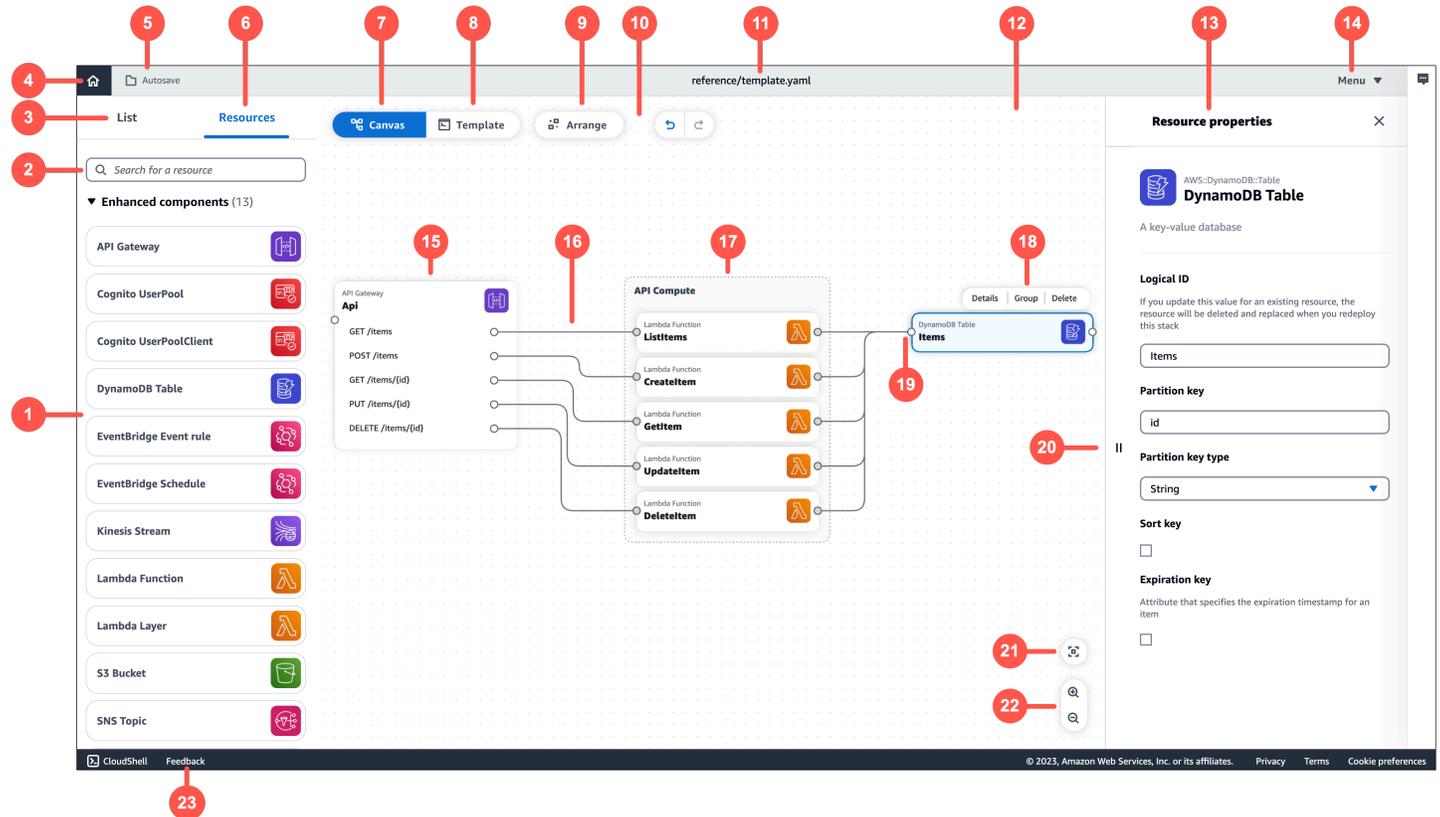
- [AWS Infrastructure Composer descripción visual de la consola](#)
- [Administre su proyecto desde la consola de Infrastructure Composer](#)



4. Crear proyecto: cree o cargue un proyecto.
5. Comenzar a crear: enlaces rápidos para empezar a crear una aplicación.
6. Comentarios: haga clic aquí para enviar comentarios.

## Diseñador visual y lienzo visual

La siguiente imagen es del diseñador visual y el lienzo visual de Infrastructure Composer:



1. Paleta de recursos: muestra tarjetas con las que puede diseñar.
2. Barra de búsqueda de recursos: busca tarjetas que puedas añadir al lienzo.
3. Lista: muestra una vista en árbol de los recursos de la aplicación.
4. Inicio: seleccione aquí para ir a la página de inicio de Infrastructure Composer.
5. Estado de almacenamiento: indica si los cambios de Infrastructure Composer se guardan en el equipo local. Los estados incluyen:
  - Guardado automático: la sincronización local está activada y el proyecto se sincroniza y guarda automáticamente.
  - Cambios guardados: la plantilla de la aplicación se guarda en el equipo local.

- Cambios no guardados: la plantilla de la aplicación contiene cambios que no se guardan en el equipo local.
6. Recursos: muestra la paleta de recursos.
  7. Lienzo: muestra la vista en lienzo de la aplicación en el área de visualización principal.
  8. Plantilla: muestra la vista de plantilla de la aplicación en el área de visualización principal.
  9. Organizar: organiza la arquitectura de la aplicación en el lienzo.
  10. Deshacer y rehacer: realiza acciones de deshacer y rehacer cuando sea posible.
  11. Nombre de la plantilla: indica el nombre de la plantilla que está diseñando.
  12. Área de visualización principal: muestra el lienzo o la plantilla en función de su selección.
  13. Panel de propiedades del recurso: muestra las propiedades relevantes de la tarjeta que se ha seleccionado en el lienzo. Este panel es dinámico. Las propiedades que se muestran cambiarán a medida que configure la tarjeta.
  14. Menú: proporciona opciones generales como las siguientes:
    - Crear un proyecto
    - Abra un archivo de plantilla o un proyecto
    - Guarda un archivo de plantilla
    - [Activa la sincronización local](#)
    - [Exportar lienzo](#)
    - Obtenga soporte
    - Métodos abreviados del teclado
  15. Tarjeta: muestra una vista de la tarjeta en el lienzo.
  16. Línea: representa una conexión entre tarjetas.
  17. Agrupar: agrupa las cartas seleccionadas para organizarlas visualmente.
  18. Acciones con las cartas: proporciona las acciones que puedes realizar con tu carta.
    - a. Detalles: abre el panel de propiedades del recurso.
    - b. Agrupar: agrupa las cartas seleccionadas.
    - c. Eliminar: elimina la tarjeta del lienzo.
  19. Puerto: puntos de conexión a otras tarjetas.
  20. Campos de propiedades de recursos: un conjunto seleccionado de campos de propiedades que puedes configurar para tus tarjetas.
  21. Vuelva a centrar: vuelva a centrar el diagrama de la aplicación en el lienzo visual.

22.Zoom: acerque y aleje el lienzo.

23.Comentarios: haga clic aquí para enviar comentarios.

## Administre su proyecto desde la consola de Infrastructure Composer

En este tema se proporcionan instrucciones sobre las tareas básicas que se realizan para administrar el proyecto desde la consola de Infrastructure Composer. Esto incluye tareas comunes, como crear un proyecto nuevo, guardar un proyecto e importar un proyecto o una plantilla. También puedes cargar un proyecto existente si activas el [modo de sincronización local](#). Tras activar el modo de sincronización local, puedes hacer lo siguiente:

- Cree un nuevo proyecto que consista en una plantilla inicial y una estructura de carpetas.
- Cargue un proyecto existente eligiendo una carpeta principal que contenga la plantilla y los archivos del proyecto.
- Use Infrastructure Composer para administrar sus plantillas y carpetas

Con el modo de sincronización local, Infrastructure Composer guarda automáticamente los cambios de plantilla y carpeta del proyecto en el equipo local. Si su navegador no admite el modo de sincronización local o si prefiere utilizar Infrastructure Composer sin el modo de sincronización local activado, puede crear una plantilla nueva o cargar una plantilla existente. Para guardar los cambios, debe exportar la plantilla a su máquina local.

### Note

Infrastructure Composer admite aplicaciones que constan de lo siguiente:

- Una AWS Serverless Application Model plantilla AWS CloudFormation o que defina el código de infraestructura.
- Estructura de carpetas que organiza los archivos del proyecto, como el código de la función Lambda, los archivos de configuración y las carpetas de compilación.

## Temas

- [Cree un proyecto nuevo en la consola de Infrastructure Composer](#)
- [Importe una carpeta de proyecto existente en la consola de Infrastructure Composer](#)
- [Importe una plantilla de proyecto existente en la consola de Infrastructure Composer](#)

- [Guarde una plantilla de proyecto existente en la consola de Infrastructure Composer](#)

## Cree un proyecto nuevo en la consola de Infrastructure Composer

Al crear un proyecto nuevo, Infrastructure Composer genera una plantilla inicial. A medida que diseña la aplicación en el lienzo, la plantilla se modifica. Para guardar el trabajo, debe exportar la plantilla o activar el modo de sincronización local.

Para crear un nuevo proyecto

1. Inicie sesión en la [consola de Infrastructure Composer](#).
2. En la página de inicio, elija Crear proyecto.

### Note

También puede cargar uno existente en Infrastructure Composer, pero primero debe [activar el modo de sincronización local](#). Una vez activado, consulte [Cargue un proyecto de Infrastructure Composer existente con la sincronización local activada](#) para cargar un proyecto existente.

## Importe una carpeta de proyecto existente en la consola de Infrastructure Composer

Con el modo de sincronización local, puede importar la carpeta principal de un proyecto existente. Si tu proyecto contiene varias plantillas, puedes elegir la plantilla que quieres cargar.

Para importar un proyecto existente desde la página de inicio

1. Inicie sesión en la [consola de Infrastructure Composer](#).
2. En la página de inicio, elija Cargar una CloudFormation plantilla.
3. En Ubicación del proyecto, selecciona Seleccionar carpeta. Seleccione la carpeta principal del proyecto y pulse Seleccionar.

### Note

Si no recibes este mensaje, es posible que tu navegador no admita el acceso al sistema de archivosAPI, que es necesario para el modo de sincronización local. Para obtener

más información, consulte [Permitir el acceso de la página web a los archivos locales en Infrastructure Composer](#).

4. Cuando el navegador se lo pida, selecciona Ver archivos.
5. En Archivo de plantilla, elige tu plantilla en la lista desplegable. Si su proyecto contiene una sola plantilla, Infrastructure Composer la seleccionará automáticamente.
6. Seleccione Crear.

Para importar un proyecto existente desde el lienzo

1. Desde el lienzo, selecciona Menú para abrir el menú.
2. En la sección Abrir, selecciona la carpeta del proyecto.

 Note

Si la opción de carpeta de proyectos no está disponible, es posible que su navegador no admita el acceso al sistema de archivosAPI, que es necesario para el modo de sincronización local. Para obtener más información, consulte [Permitir el acceso de la página web a los archivos locales en Infrastructure Composer](#).

3. Para la ubicación del proyecto, elija Seleccionar carpeta. Seleccione la carpeta principal del proyecto y pulse Seleccionar.
4. Cuando el navegador se lo pida, selecciona Ver archivos.
5. En Archivo de plantilla, elige tu plantilla en la lista desplegable. Si su proyecto contiene una sola plantilla, Infrastructure Composer la seleccionará automáticamente.
6. Seleccione Crear.

Al importar una carpeta de proyecto existente, Infrastructure Composer activa el modo de sincronización local. Los cambios realizados en la plantilla o los archivos del proyecto se guardan automáticamente en el equipo local.

Importe una plantilla de proyecto existente en la consola de Infrastructure Composer

Al importar una AWS SAM plantilla AWS CloudFormation o una existente, Infrastructure Composer genera automáticamente una visualización de la arquitectura de la aplicación en el lienzo.

Puede importar una plantilla de proyecto desde su máquina local.

## Para importar una plantilla de proyecto existente

1. Inicie sesión en la [consola de Infrastructure Composer](#).
2. Seleccione Crear proyecto para abrir un lienzo en blanco.
3. Seleccione Menú para abrir el menú.
4. En la sección Abrir, selecciona Archivo de plantilla.
5. Seleccione tu plantilla y selecciona Abrir.

Para guardar los cambios en la plantilla, debe exportarla o activar el modo de sincronización local.

## Guarde una plantilla de proyecto existente en la consola de Infrastructure Composer

Si no utiliza el modo de sincronización local, debe exportar la plantilla para guardar los cambios. Si tienes activado el modo de sincronización local, no es necesario guardar la plantilla manualmente. Los cambios se guardan automáticamente en su máquina local.

## Para guardar una plantilla de proyecto existente

1. En el lienzo de Infrastructure Composer, seleccione Menú para abrir el menú.
2. En la sección Guardar, elija Guardar archivo de plantilla.
3. Proporcione un nombre para la plantilla.
4. Seleccione una ubicación para guardar la plantilla.
5. Seleccione Guardar.

## Conecte la consola de Infrastructure Composer con su consola local IDE

Para conectar la consola de Infrastructure Composer con su entorno de desarrollo integrado local (IDE), utilice el modo de sincronización local. Este modo sincroniza y guarda automáticamente los datos en su máquina local. Para obtener más información sobre el modo de sincronización local, consulte [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#). Para obtener instrucciones sobre el uso del modo de sincronización local, consulte [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#).

**Note**

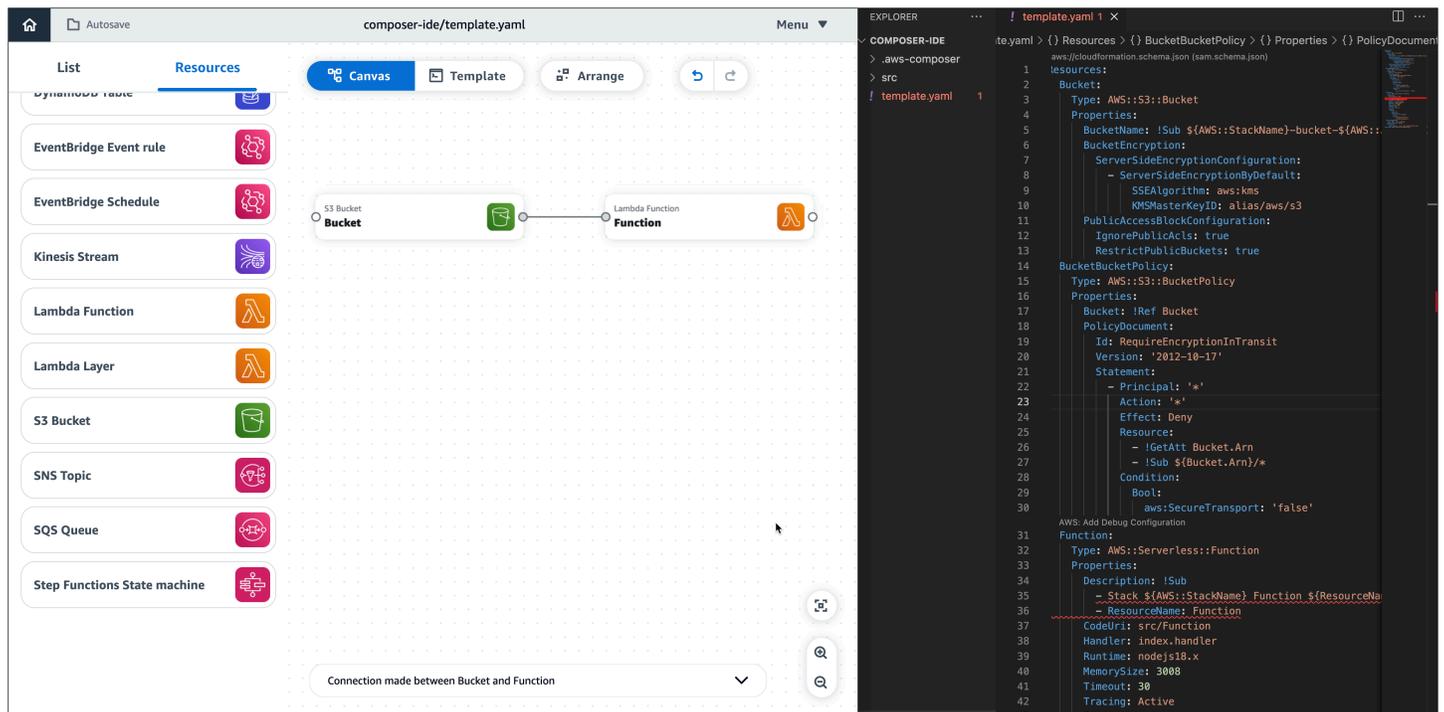
La opción Activar la sincronización local no está disponible en todos los navegadores. Está disponible en Google Chrome y Microsoft Edge.

## Ventajas de usar Infrastructure Composer con su empresa local IDE

Al diseñar en Infrastructure Composer, la plantilla local y el directorio del proyecto se sincronizan y guardan automáticamente.

Puede usar su versión local IDE para ver los cambios y modificar las plantillas. Los cambios que realice localmente se sincronizan automáticamente con Infrastructure Composer.

Puede utilizar herramientas locales, como la interfaz de línea de AWS Serverless Application Model comandos (AWS SAM CLI), para crear, probar e implementar la aplicación, etc. El siguiente ejemplo muestra cómo arrastrar y soltar recursos en el lienzo visual de Infrastructure Composer, lo que, a su vez, crea marcas en la AWS SAM plantilla de su entorno local IDE.



The screenshot displays the AWS Infrastructure Composer interface. On the left, a 'Resources' panel lists various AWS services such as DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, Lambda Function, Lambda Layer, S3 Bucket, SNS Topic, SQS Queue, and Step Functions State machine. The central canvas shows a visual diagram with an 'S3 Bucket' resource connected to a 'Lambda Function' resource. On the right, a code editor displays the corresponding AWS SAM template in YAML format. The code defines an S3 bucket with server-side encryption and a bucket policy, and a Lambda function that triggers on S3 events.

```
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
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 Infrastructure Composer con su local IDE

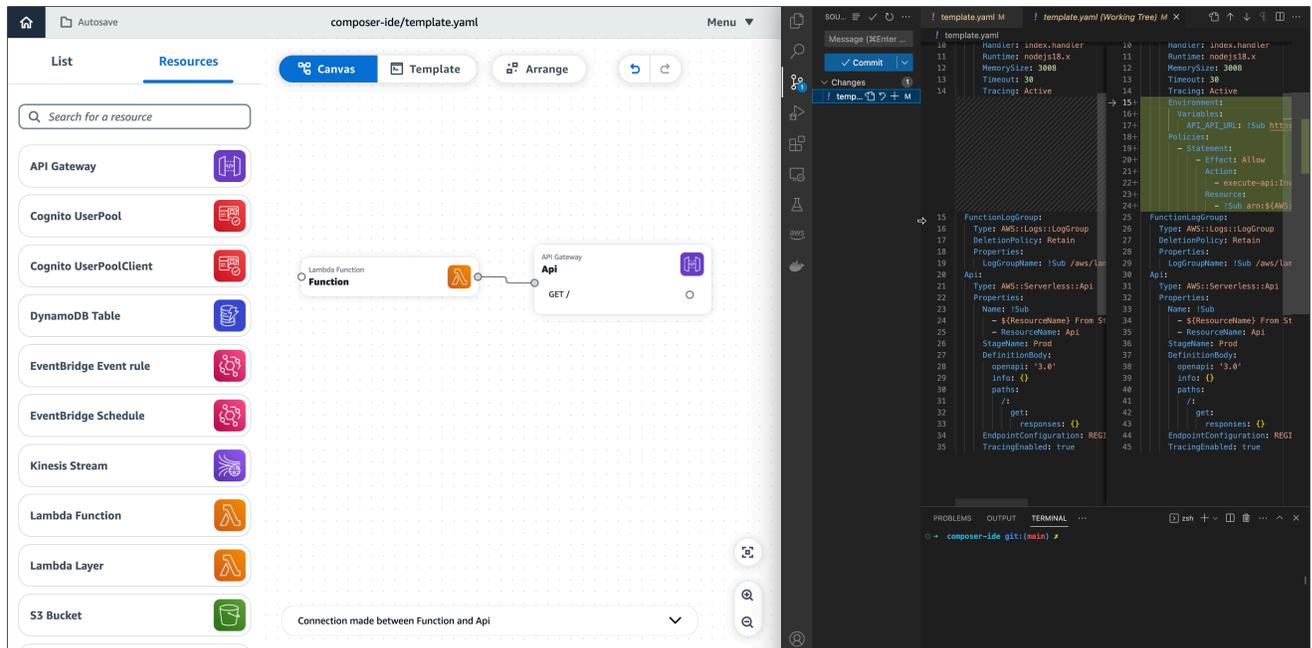
### Para integrar Infrastructure Composer con su local IDE

1. En Infrastructure Composer, cree o cargue un proyecto y active la sincronización local seleccionando el botón de menú en la parte superior derecha de la pantalla y seleccionando Activar la sincronización local.

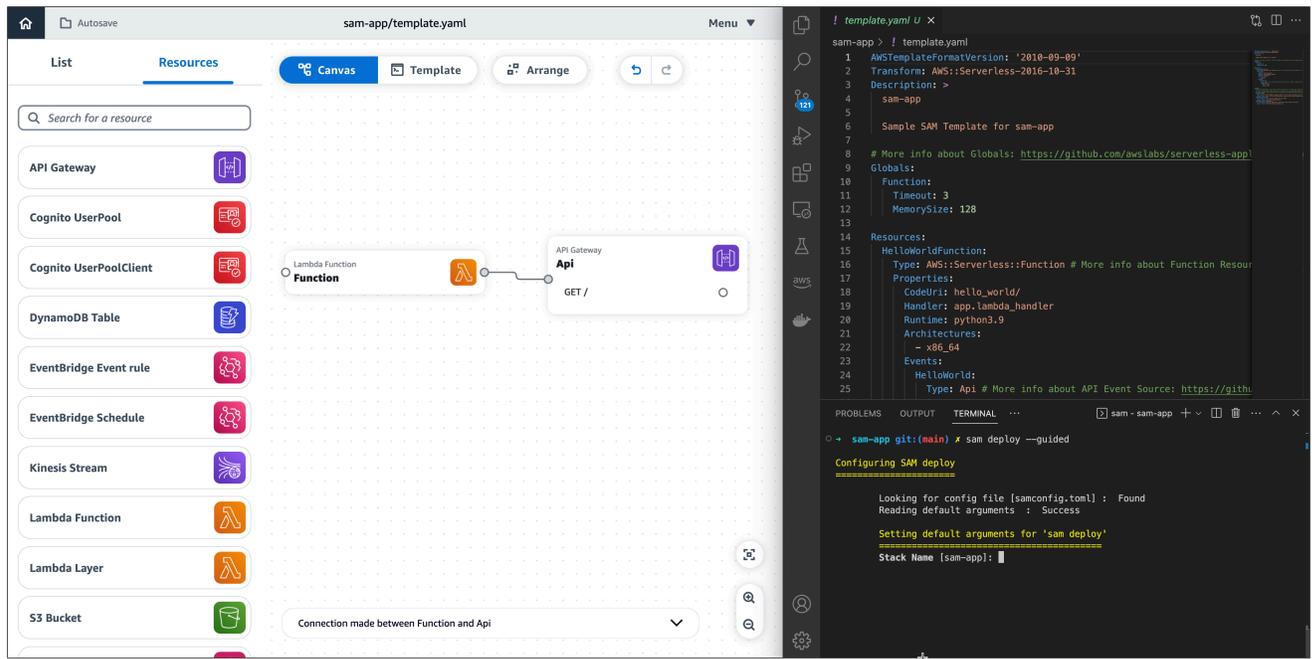
#### Note

La opción Activar la sincronización local no está disponible en todos los navegadores. Está disponible en Google Chrome y Microsoft Edge.

2. En tu localIDE, abre la misma carpeta de proyectos que Infrastructure Composer.
3. Utilice Infrastructure Composer con su localIDE. Las actualizaciones realizadas en Infrastructure Composer se sincronizarán automáticamente con su máquina local. Estos son algunos ejemplos de lo que puede hacer:
  - a. Utilice el sistema de control de versiones que prefiera para realizar un seguimiento de las actualizaciones que realiza Infrastructure Composer.



- b. Úselo de AWS SAM CLI forma local para crear, probar e implementar su aplicación y mucho más. Para obtener más información, consulte [Implemente su aplicación sin servidor Infrastructure Composer en la nube AWS](#).



## Permitir el acceso de la página web a los archivos locales en Infrastructure Composer

La consola de Infrastructure Composer admite el [modo de sincronización local](#) y [las funciones de importación desde la consola Lambda](#). Para utilizar estas funciones, se necesita un navegador web que admita el acceso al API sistema de archivos. Todas las versiones recientes de Google Chrome y Microsoft Edge admiten todas las funciones del sistema de acceso al sistema de archivos API y se pueden utilizar con el modo de sincronización local en Infrastructure Composer.

El acceso al sistema de archivos API permite a las páginas web acceder a su sistema de archivos local para leer, escribir o guardar archivos. Esta función está desactivada de forma predeterminada y requiere su permiso mediante un mensaje visual para habilitarla. Una vez concedido, este acceso permanece durante toda la sesión de navegación de la página web.

Para obtener más información sobre el acceso al sistema de archivos API, consulte:

- [Acceso al sistema de archivos API](#) en los documentos web de mdn.
- [El acceso al sistema de archivos API: simplifica el acceso a los archivos locales](#) en el sitio web [web.dev](#).

## modo de sincronización local

El modo de sincronización local le permite sincronizar y guardar automáticamente los archivos de plantilla y las carpetas de proyectos de forma local mientras diseña en Infrastructure Composer. Para utilizar esta función, se necesita un navegador web compatible con el acceso al API sistema de archivos.

### Data Infrastructure Composer obtiene acceso a

Infrastructure Composer obtiene acceso de lectura y escritura a la carpeta del proyecto que usted permita, así como a las carpetas secundarias de esa carpeta del proyecto. Este acceso se utiliza para crear, actualizar y guardar todos los archivos de plantilla, carpetas de proyectos y directorios de copias de seguridad que se generen a medida que diseña. Los datos a los que accede Infrastructure Composer no se utilizan para ningún otro propósito y no se almacenan fuera del sistema de archivos local.

#### Acceso a datos confidenciales

El acceso al sistema de archivos API excluye o limita el acceso a directorios específicos que pueden contener datos confidenciales. Se producirá un error si selecciona uno de estos directorios para usarlo con el modo de sincronización local de Infrastructure Composer. Puede elegir otro directorio local para conectarse o usar Infrastructure Composer en su modo predeterminado con la sincronización local desactivada.

Para obtener más información, incluidos ejemplos de directorios confidenciales, consulte [los usuarios que dan acceso a más o más archivos confidenciales de los previstos en el borrador del informe](#) del grupo comunitario del W3C sobre el acceso al sistema de archivos.

Si usas Windows Subsystem for Linux (WSL), el acceso al sistema de archivos API excluye el acceso a todo Linux debido a su ubicación dentro de su Windows sistema. Puede utilizar Infrastructure Composer con la sincronización local desactivada o configurar una solución para sincronizar los archivos del proyecto desde su WSL a un directorio de trabajo en Windows. A continuación, utilice el modo de sincronización local de Infrastructure Composer con su Windows directorio.

## Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer

En esta sección se proporciona información sobre el uso del modo de sincronización local de Infrastructure Composer para sincronizar y guardar automáticamente el proyecto en el equipo local.

Le recomendamos que utilice la sincronización local por los siguientes motivos:

Puedes activar la sincronización local para un proyecto nuevo o cargar un proyecto existente con la sincronización local activada.

- De forma predeterminada, debes guardar manualmente la plantilla de la aplicación a medida que diseñas. Utilice la sincronización local para guardar automáticamente la plantilla de la aplicación en el equipo local a medida que realiza cambios.
- La sincronización local administra y sincroniza automáticamente las carpetas de proyectos, la carpeta de respaldo y [los archivos externos compatibles con](#) su máquina local.
- Al utilizar la sincronización local, puede conectar Infrastructure Composer con la local IDE para acelerar el desarrollo. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su consola local IDE](#).

## ¿Qué guarda el modo de sincronización local

El modo de sincronización local sincroniza automáticamente y guarda lo siguiente en tu máquina local:

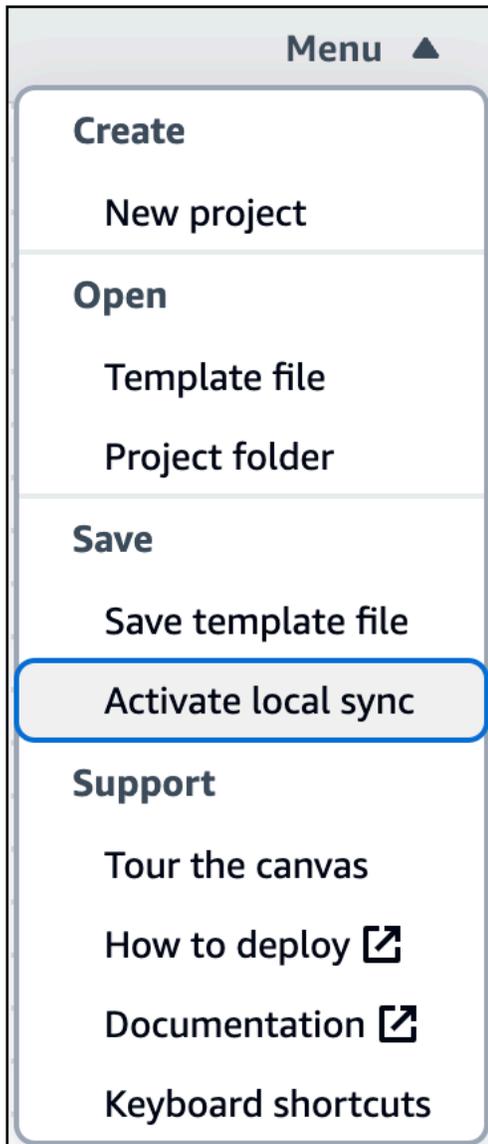
- Archivo de plantilla de aplicación: la plantilla AWS CloudFormation or AWS Serverless Application Model (AWS SAM) que contiene la infraestructura como código (IaC).
- Carpetas de proyectos: estructura de directorios general que organiza AWS Lambda las funciones.
- Directorio de respaldo: un directorio de respaldo con nombre `.aws-composer`, creado en la raíz de la ubicación del proyecto. Este directorio contiene una copia de seguridad del archivo de plantilla de la aplicación y de las carpetas del proyecto.
- Archivos externos: archivos externos compatibles que puede usar en Infrastructure Composer. Para obtener más información, consulte [Archivos externos de referencia en Infrastructure Composer](#).

## Requisitos del navegador

El modo de sincronización local requiere un navegador compatible con el acceso al sistema de archivosAPI. Para obtener más información, consulte [Permitir el acceso de la página web a los archivos locales en Infrastructure Composer](#).

## Activar el modo de sincronización local

El modo de sincronización local está desactivado de forma predeterminada. Puede activar el modo de sincronización local a través del menú Infrastructure Composer.



Para obtener instrucciones sobre cómo activar la sincronización local y los proyectos de carga existentes, consulte los siguientes temas:

- [Active la sincronización local en Infrastructure Composer](#)
- [Cargue un proyecto de Infrastructure Composer existente con la sincronización local activada](#)

## Active la sincronización local en Infrastructure Composer

Para activar la sincronización local, sigue estos pasos:

1. En la página de [inicio](#) de Infrastructure Composer, seleccione Crear proyecto.
2. En el menú de Infrastructure Composer, seleccione Activar la sincronización local.
3. Para la ubicación del proyecto, pulse Seleccionar carpeta y elija un directorio. Aquí es donde Infrastructure Composer guardará y sincronizará los archivos y carpetas de la plantilla a medida que vaya diseñando.

### Note

La ubicación del proyecto no debe contener una plantilla de aplicación existente.

4. Cuando se le pida que permita el acceso, seleccione Ver archivos.
5. Pulsa Activar. Cuando se te pida que guardes los cambios, selecciona Guardar cambios.

Cuando esté activado, el indicador de guardado automático se mostrará en el área superior izquierda del lienzo.

## Cargue un proyecto de Infrastructure Composer existente con la sincronización local activada

Para cargar un proyecto existente con la sincronización local activada, siga estos pasos:

1. En la página de [inicio](#) de Infrastructure Composer, seleccione Cargar una AWS CloudFormation plantilla.
2. En el menú de Infrastructure Composer, seleccione Abrir > Carpeta del proyecto.
3. Para la ubicación del proyecto, pulse Seleccionar carpeta y elija la carpeta raíz del proyecto.
4. Cuando se te pida que permitas el acceso, selecciona Ver archivos.
5. En Archivo de plantilla, seleccione la plantilla de la aplicación y pulse Crear.
6. Cuando se le pida que guarde los cambios, seleccione Guardar cambios.

Cuando esté activado, el indicador de guardado automático se mostrará en el área superior izquierda del lienzo.

## Importación de funciones a Infrastructure Composer desde la consola Lambda

Infrastructure Composer proporciona una integración con la AWS Lambda consola. Puede importar una función de Lambda de la consola de Lambda a la consola de Infrastructure Composer. A continuación, utilice el lienzo de Infrastructure Composer para seguir diseñando la arquitectura de la aplicación.

- Esta integración requiere un navegador compatible con el acceso al sistema de archivosAPI. Para obtener más información, consulte [Permitir el acceso de la página web a los archivos locales en Infrastructure Composer](#).
- Al importar la función Lambda a Infrastructure Composer, debe activar el modo de sincronización local para guardar los cambios. Para obtener más información, consulte [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#).

Para empezar a usar esta integración, consulte [Utilización AWS Lambda con AWS Infrastructure Composer](#) en la Guía para AWS Lambda desarrolladores.

## Exporte una imagen del lienzo visual de Infrastructure Composer

En este tema se describe la función de lienzo de exportación de la AWS Infrastructure Composer consola.

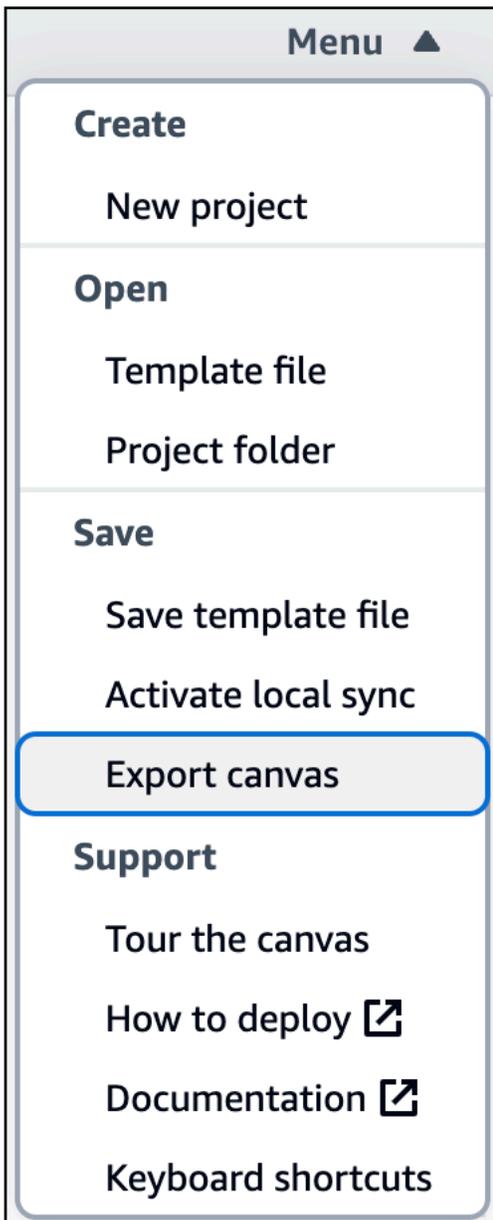
Para obtener una visión general visual de todas las funciones de Infrastructure Composer, consulte [AWS Infrastructure Composer descripción visual de la consola](#).

### Acerca de Export Canvas

La función de exportación de lienzo exporta el lienzo de la aplicación como una imagen a su máquina local.

- Infrastructure Composer elimina los elementos de la interfaz de usuario del diseñador visual y exporta solo el diagrama de la aplicación.
- El formato de archivo de imagen predeterminado espng.
- El archivo se exporta a la ubicación de descarga predeterminada de su máquina local.

Puede acceder a la función de exportación del lienzo desde el menú.



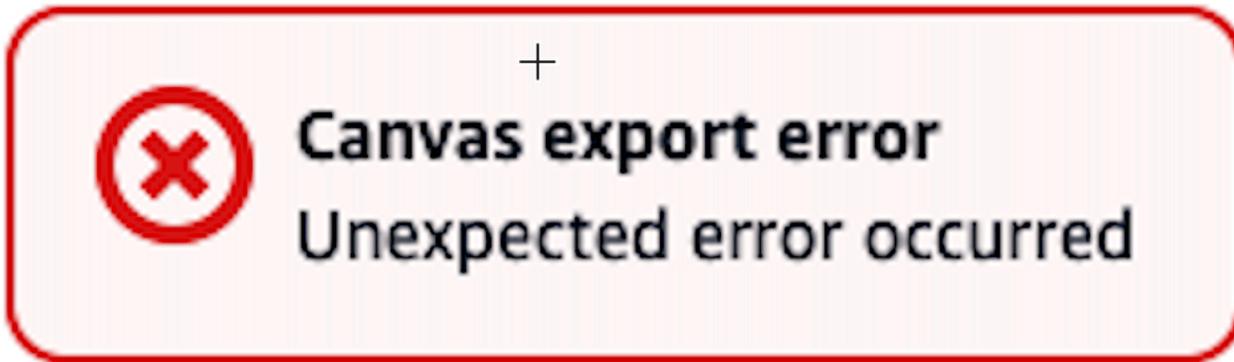
## Exportación de lienzo

Al exportar el lienzo, Infrastructure Composer muestra un mensaje de estado.

Si la exportación se realiza correctamente, verá el siguiente mensaje:



Si la exportación no se realizó correctamente, aparecerá un mensaje de error. Si recibe un error, intente exportar de nuevo.



## Uso de Infrastructure Composer en modo CloudFormation consola

Infrastructure Composer en modo CloudFormation consola es la herramienta recomendada para visualizar AWS CloudFormation las plantillas. También puede usar esta herramienta para crear y editar AWS CloudFormation plantillas.

### ¿En qué se diferencia este modo de la consola de Infrastructure Composer?

Por lo general, Infrastructure Composer en modo CloudFormation consola tiene la misma funcionalidad que la [consola Infrastructure Composer predeterminada](#), pero hay algunas diferencias que se deben tener en cuenta.

- Este modo está integrado con el flujo de trabajo apilado de la AWS CloudFormation consola. Esto le permite utilizar Infrastructure Composer directamente en ella AWS CloudFormation.
- [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#), una función que sincroniza y guarda automáticamente los datos en el equipo local, no es compatible.
- Las tarjetas relacionadas con Lambda (función Lambda y capa Lambda) requieren compilaciones de código y soluciones de empaquetado que no están disponibles en este modo.

#### Note

Estas tarjetas y la sincronización local se pueden utilizar en la consola de [Infrastructure Composer](#) o en la. AWS Toolkit for Visual Studio Code

Al abrir Infrastructure Composer desde la AWS CloudFormation consola, Infrastructure Composer se abre en modo CloudFormation consola. En este modo, puede usar Infrastructure Composer para visualizar, crear y actualizar las plantillas.

## ¿Cómo acceder a Infrastructure Composer en modo CloudFormation consola

Infrastructure Composer en modo CloudFormation consola es una actualización de AWS CloudFormation Designer. Recomendamos usar Infrastructure Composer para visualizar AWS CloudFormation las plantillas. También puede usar esta herramienta para crear y editar AWS CloudFormation plantillas.

1. Ve a la [consola de Cloudformation](#) e inicia sesión.
2. Seleccione Infrastructure Composer en el menú de navegación de la izquierda. Esto lo llevará a Infrastructure Composer en modo CloudFormation consola.

### Note

Para obtener información sobre el uso de Infrastructure Composer en modo CloudFormation consola, consulte [Uso de Infrastructure Composer en modo CloudFormation consola](#).

## Visualice una implementación en Infrastructure Composer en modo CloudFormation consola

Siga las instrucciones de este tema para visualizar una plantilla de AWS CloudFormation stack/ Infrastructure Composer implementada.

1. Ve a la [AWS CloudFormation consola](#) e inicia sesión.
2. Selecciona la pila que quieres editar.
3. Selecciona la pestaña Plantilla.
4. Seleccione Infrastructure Composer.

Infrastructure Composer visualizará su pila o plantilla. Aquí también se pueden realizar cambios.

## Cree una plantilla nueva en Infrastructure Composer en modo CloudFormation consola

Siga las instrucciones de este tema para crear una plantilla nueva.

1. Ve a la [AWS CloudFormation consola](#) e inicia sesión.
2. Seleccione Infrastructure Composer en el menú de navegación de la izquierda. Esto abrirá Infrastructure Composer en modo CloudFormation consola.
3. Arrastre, suelte, configure y conecte los recursos ([tarjetas](#)) que necesite de la paleta Recursos.

### Note

Consulte [¿Cómo componer](#) para obtener más información sobre el uso de Infrastructure Composer y tenga en cuenta que las tarjetas relacionadas con Lambda (función Lambda y capa Lambda) requieren compilaciones de código y soluciones de empaquetado que no están disponibles en Infrastructure Composer en modo consola. CloudFormation Estas tarjetas se pueden utilizar en la consola de [Infrastructure](#) Composer o en la AWS Toolkit for Visual Studio Code Para obtener información sobre el uso de estas herramientas, consulte [Dónde puede usar Infrastructure Composer](#).

4. Haga doble clic en las tarjetas para usar el panel de propiedades del recurso y especificar cómo están configuradas las tarjetas.
5. [Conecte sus tarjetas](#) para especificar el flujo de trabajo basado en eventos de su aplicación.
6. Seleccione Plantilla para ver y editar el código de infraestructura. Los cambios se sincronizan automáticamente con la vista del lienzo.
7. Una vez que la plantilla esté lista para exportarse a una pila, selecciona Crear plantilla.
8. Seleccione el CloudFormation botón Confirmar y exportar a. De este modo, volverás al flujo de trabajo de creación de pilas y aparecerá un mensaje confirmando que la plantilla se ha importado correctamente.

### Note

Solo se pueden exportar las plantillas que contengan recursos.

9. En el flujo de trabajo Crear pila, selecciona Siguiente.
10. Proporcione un nombre de pila, revise los parámetros de la lista y seleccione Siguiente.

**Note**

El nombre de la pila debe empezar con una letra y contener solo letras, números y guiones.

11. Seleccione **Siguiente** después de proporcionar la siguiente información:

- Etiquetas asociadas a la pila
- Permisos de pila
- Las opciones de error de la pila

**Note**

Para obtener orientación sobre la administración de pilas, consulta [las prácticas AWS CloudFormation recomendadas](#) en la Guía del AWS CloudFormation usuario.

12. Confirma que los detalles de tu pila son correctos, comprueba las confirmaciones en la parte inferior de la página y selecciona el botón **Enviar**.

AWS CloudFormation empezará a crear la pila en función de los datos de tu plantilla.

## Actualice una pila existente en Infrastructure Composer en modo CloudFormation consola

Siga las instrucciones de este tema para actualizar una AWS CloudFormation pila existente.

**Note**

Si el archivo está guardado localmente, le recomendamos que lo utilice [AWS Toolkit for Visual Studio Code](#).

1. Ve a la [AWS CloudFormation consola](#) e inicia sesión.
2. Selecciona la pila que quieres editar.
3. Seleccione el botón **Actualizar**. Si lo hace, accederá al asistente de actualización de pilas.

4. A la derecha, selecciona Editar en Infrastructure Composer.
5. Seleccione el botón de abajo denominado Editar en Infrastructure Composer. Esto lo llevará a Infrastructure Composer en modo CloudFormation consola.
6. Aquí puede arrastrar, soltar, configurar y conectar recursos ([tarjetas](#)) desde la paleta Recursos.

 Note

Consulte [¿Cómo componer](#) para obtener más información sobre el uso de Infrastructure Composer y tenga en cuenta que las tarjetas relacionadas con Lambda (función Lambda y capa Lambda) requieren compilaciones de código y soluciones de empaquetado que no están disponibles en Infrastructure Composer en modo consola. CloudFormation Estas tarjetas se pueden utilizar en la consola de [Infrastructure](#) Composer o en la. AWS Toolkit for Visual Studio Code Para obtener información sobre el uso de estas herramientas, consulte [Dónde puede usar Infrastructure Composer](#).

7. Cuando esté listo para exportar los cambios a AWS CloudFormation, seleccione Actualizar plantilla.
8. Seleccione Confirmar y continúa CloudFormation. Esto te llevará de vuelta al flujo de trabajo de actualización de la pila con un mensaje que confirma que la plantilla se ha importado correctamente.

 Note

Solo se pueden exportar las plantillas que contengan recursos.

9. En el flujo de trabajo Actualizar pila, selecciona Siguiente.
10. Revise todos los parámetros de la lista y seleccione Siguiente.
11. Seleccione Siguiente después de proporcionar la siguiente información:
  - Etiquetas asociadas a la pila
  - Permisos de pila
  - Las opciones de error de la pila

**Note**

Para obtener orientación sobre la administración de pilas, consulta [las prácticas AWS CloudFormation recomendadas](#) en la Guía del AWS CloudFormation usuario.

12. Confirma que los detalles de tu pila son correctos, comprueba las confirmaciones en la parte inferior de la página y selecciona el botón Enviar.

AWS CloudFormation empezará a actualizar la pila en función de las actualizaciones que hayas realizado en tu plantilla.

## Uso de Infrastructure Composer desde AWS Toolkit for Visual Studio Code

En esta sección se describe cómo puede utilizar AWS Infrastructure Composer desde [AWS Toolkit for Visual Studio Code](#). Incluye una descripción visual de Infrastructure Composer desde el AWS Toolkit for Visual Studio Code. También incluye instrucciones que muestran cómo puede acceder a esta experiencia y sincronizar su proyecto de VS Code a la AWS nube. Para sincronizarlo, usa el sam sync comando del AWS SAM CLI. Esta sección también proporciona orientación sobre el uso Amazon Q mientras esté en Infrastructure Composer desde AWS Toolkit for Visual Studio Code.

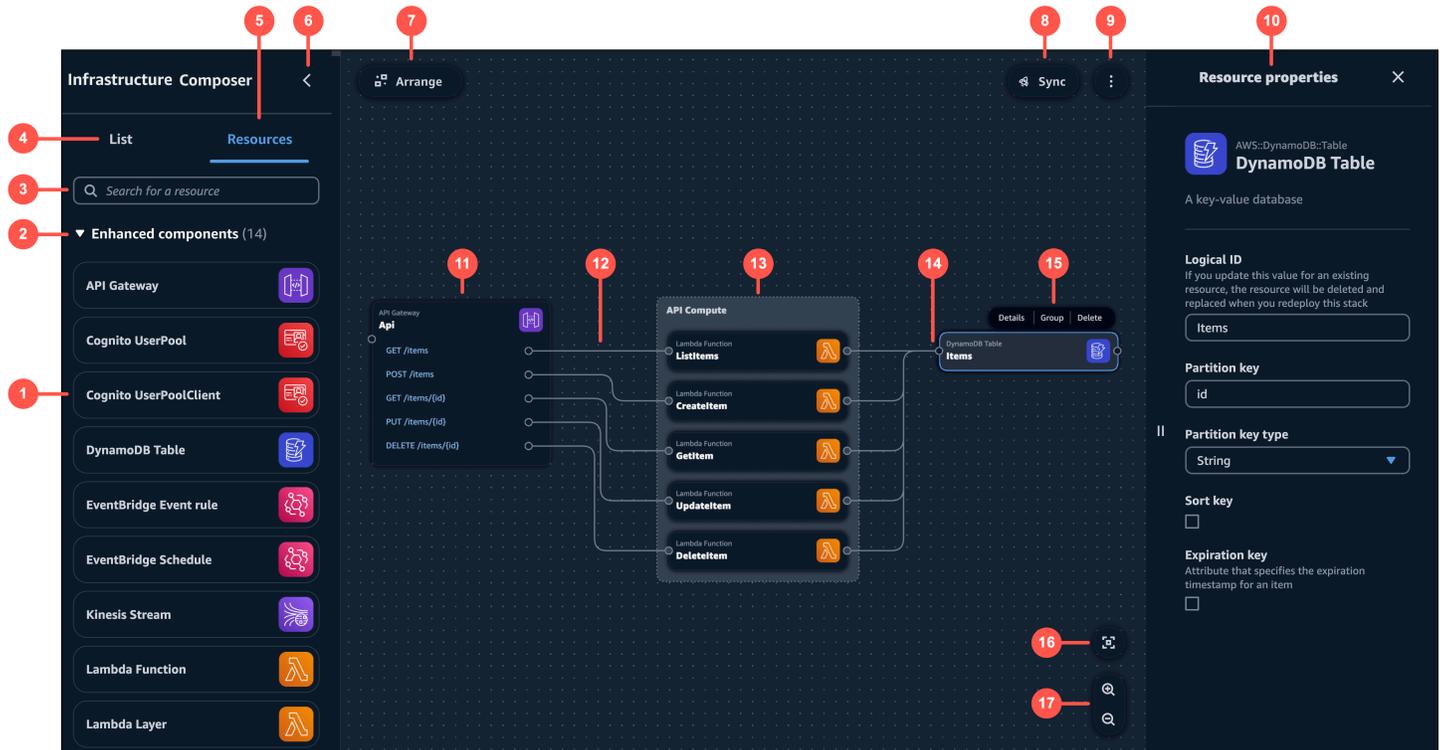
Para obtener más información sobre el uso de Infrastructure Composer desde el AWS Toolkit for Visual Studio Code, consulte [¿Cómo componer](#). El contenido de esta sección se aplica a esta experiencia, así como a la experiencia de la consola de Infrastructure Composer.

### Temas

- [Información general visual de Infrastructure Composer desde AWS Toolkit for Visual Studio Code](#)
- [Acceda a Infrastructure Composer desde AWS Toolkit for Visual Studio Code](#)
- [Sincronice Infrastructure Composer para implementarlo en Nube de AWS](#)
- [Uso AWS Infrastructure Composer con Amazon Q Developer](#)

# Información general visual de Infrastructure Composer desde AWS Toolkit for Visual Studio Code

El diseñador visual de Infrastructure Composer AWS Toolkit for Visual Studio Code incluye un lienzo visual, que incluye los componentes numerados en la siguiente imagen y que se enumeran a continuación.



1. Paleta de recursos: muestra tarjetas con las que puede diseñar.
2. Categorías de tarjetas: las tarjetas se organizan por categorías exclusivas de Infrastructure Composer.
3. Barra de búsqueda de recursos: busque tarjetas que pueda añadir al lienzo.
4. Lista: muestra una vista en árbol de los recursos de la aplicación.
5. Recursos: muestra la paleta de recursos.
6. Alternar entre el panel izquierdo: permite ocultar o mostrar el panel izquierdo.
7. Organizar: organiza la arquitectura de la aplicación en el lienzo.
8. Sincronizar: inicia el AWS Serverless Application Model (AWS SAM) CLI `aws sam sync` comando para implementar la aplicación.
9. Menú: proporciona opciones generales como las siguientes:
  - Exportación de lienzo

- Recorre el lienzo
- Enlaces a la documentación
- Métodos abreviados de teclado

10 Panel de propiedades del recurso: muestra las propiedades relevantes de la tarjeta que se ha seleccionado en el lienzo. Este panel es dinámico. Las propiedades que se muestran cambiarán a medida que configure la tarjeta.

11 Tarjeta: muestra una vista de la tarjeta en el lienzo.

12 Línea: representa una conexión entre tarjetas.

13 Grupo: grupo de cartas. Puede agrupar las tarjetas para organizarlas visualmente.

14 Puerto: puntos de conexión a otras tarjetas.

15 Acciones con las cartas: proporciona las acciones que puedes realizar con tu tarjeta.

- Detalles: abre el panel de propiedades del recurso.
- Agrupar: agrupa las cartas seleccionadas.
- Eliminar: elimina la tarjeta del lienzo y la plantilla.

16 Volver a centrar: vuelva a centrar el diagrama de la aplicación en el lienzo visual.

17 Zoom: acerque y aleje el lienzo.

## Acceda a Infrastructure Composer desde AWS Toolkit for Visual Studio Code

Siga las instrucciones de este tema para acceder a Infrastructure Composer desde AWS Toolkit for Visual Studio Code.

### Note

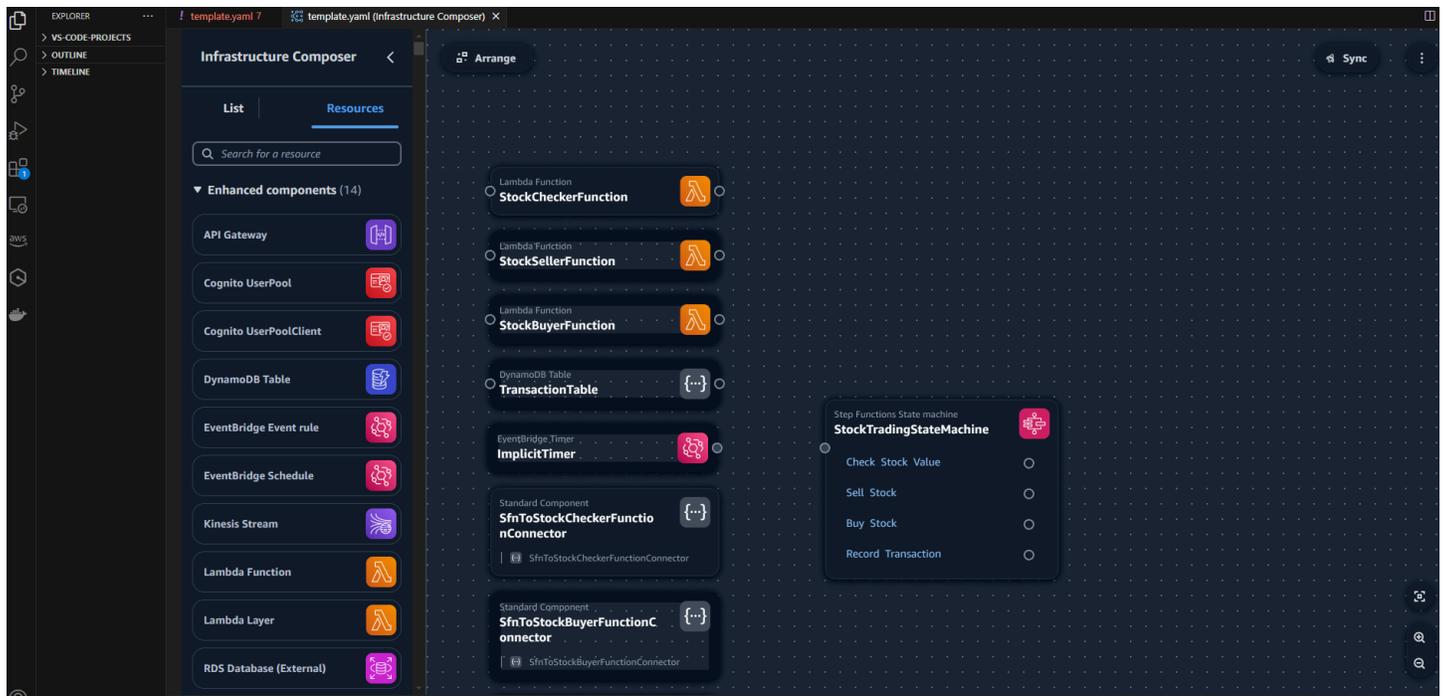
Para poder acceder a Infrastructure Composer desde AWS Toolkit for Visual Studio Code, primero debe descargar e instalar el Toolkit for VS Code. Para obtener instrucciones, consulte [Descarga del kit de herramientas para VS Code](#).

Para acceder a Infrastructure Composer desde el Toolkit for VS Code

Puede acceder a Infrastructure Composer de cualquiera de las siguientes maneras:

1. Al seleccionar el botón Infrastructure Composer desde cualquier AWS SAM plantilla AWS CloudFormation o plantilla.
2. A través del menú contextual haciendo clic con el botón derecho en su AWS SAM plantilla AWS CloudFormation o plantilla.
3. Desde la paleta de comandos de VS Code.

El siguiente es un ejemplo de cómo acceder a Infrastructure Composer desde el botón Infrastructure Composer:



Para obtener más información sobre cómo acceder a Infrastructure Composer, consulte [Acceder AWS Infrastructure Composer desde el kit de herramientas](#).

## Sincronice Infrastructure Composer para implementarlo en Nube de AWS

Utilice el botón de sincronización AWS Toolkit for Visual Studio Code que aparece AWS Infrastructure Composer desde para implementar la aplicación en Nube de AWS.

El botón de sincronización inicia el `sam sync` comando desde la interfaz de línea de AWS SAM comandos (CLI).

El `sam sync` comando puede implementar nuevas aplicaciones o sincronizar rápidamente los cambios que realice localmente en el Nube de AWS. La ejecución `sam sync` puede incluir lo siguiente:

- Cree la aplicación con el fin de `sam build` preparar los archivos de la aplicación local para su implementación mediante la creación o actualización de un `.aws-sam` directorio local.
- Para obtener recursos que respalden el AWS servicio APIs, el AWS SAM CLI utilizará el APIs para implementar sus cambios. El AWS SAM CLI lo hace para actualizar rápidamente sus recursos en la nube.
- Si es necesario, el AWS SAM CLI realiza una AWS CloudFormation implementación para actualizar toda la pila mediante un conjunto de cambios.

El `sam sync` comando es el más adecuado para entornos de desarrollo rápido, ya que actualizar rápidamente los recursos de la nube puede beneficiar a sus flujos de trabajo de desarrollo y pruebas.

Para obtener más información `sam sync`, consulta Cómo [usar sam sync](#) en la Guía para AWS Serverless Application Model desarrolladores.

## Configuración

Para utilizar la función de sincronización de Infrastructure Composer, debe disponer de AWS SAM CLI instalado en su máquina local. Para obtener instrucciones, consulte [Instalación del AWS SAM CLI](#) en la Guía para desarrolladores de AWS Serverless Application Model .

Al utilizar la función de sincronización en Infrastructure Composer, el AWS SAM CLI hace referencia al archivo de configuración para obtener la información que necesita para sincronizar la aplicación con Nube de AWS. Para obtener instrucciones sobre cómo crear, modificar y usar los archivos de configuración, consulte [Configurar los ajustes del proyecto](#) en la Guía para AWS Serverless Application Model desarrolladores.

## Sincronice e implemente su aplicación

Para sincronizar la aplicación con el Nube de AWS

1. Seleccione el botón de sincronización en el lienzo de Infrastructure Composer.
2. Es posible que reciba un mensaje para confirmar que está trabajando con una pila de desarrollo. Seleccione OK para continuar.
3. Es posible que Infrastructure Composer le pida que configure las siguientes opciones:
  - Región de AWS— La región con la que se sincronizará la aplicación.
  - AWS CloudFormation nombre de la pila: el nombre de la AWS CloudFormation pila. Puedes seleccionar un nombre de pila existente o crear uno nuevo.

- Depósito de Amazon Simple Storage Service (Amazon S3): el nombre del depósito de Amazon S3. El AWS SAM CLI empaquetará y almacenará los archivos de su aplicación y el código de función aquí. Puede seleccionar un depósito existente o crear uno nuevo.

Infrastructure Composer iniciará el AWS SAM CLI `sam sync` ejecute el comando y abra una ventana de terminal en su IDE interior para mostrar su progreso.

## Uso AWS Infrastructure Composer con Amazon Q Developer

AWS Infrastructure Composer from the AWS Toolkit for Visual Studio Code proporciona una integración con Amazon Q. Puedes usar Amazon Q dentro de Infrastructure Composer para generar el código de infraestructura para sus AWS recursos a medida que diseña la aplicación.

Amazon Q es un generador de código de uso general basado en el aprendizaje automático. Para obtener más información, consulta [¿Qué es Amazon Q?](#) en el Amazon Q Developer Guía del usuario.

Para tarjetas de recursos estándar y de componentes estándar, puede utilizar Amazon Q para generar sugerencias de código de infraestructura para sus recursos.

The screenshot displays the AWS Infrastructure Composer interface. On the left, a grid of resource cards is shown. One card, labeled 'Standard Component VPC', is highlighted with a blue border. Above the grid are 'Sync' and a menu icon. Below the grid are 'Details', 'Group', and 'Delete' buttons. On the right, the 'Resource properties' panel is open for the 'AWS::EC2::VPC' CFN Resource. It shows the 'Editing' dropdown set to 'VPC'. The 'Logical ID' section indicates that updating this value will generate a new resource. The 'Resource configuration' section notes that updating this value will change the resource's properties. A modal dialog is open, displaying a loading spinner and the text 'Generating can take up to 30 seconds.' with a 'Stop generating' button. A 'Resource reference' button is also visible at the bottom right of the properties panel.

Las tarjetas de recursos y componentes estándar pueden representar un AWS CloudFormation recurso o un conjunto de AWS CloudFormation recursos. Para obtener más información, consulte [Configurar y modificar tarjetas en Infrastructure Composer](#).

## Configuración

Para utilizar Amazon Q en Infrastructure Composer, debe autenticarse con Amazon Q en el kit de herramientas. Para obtener instrucciones, consulte [Cómo empezar con Amazon Q en VS Code y JetBrains](#) en Amazon Q Developer Guía del usuario.

## Utilización Amazon Q Developer en Infrastructure Composer

Puede usar... Amazon Q Developer desde el panel de propiedades del recurso de cualquier tarjeta de recursos o componentes estándar.

Para utilizar Amazon Q en Infrastructure Composer

1. Desde una tarjeta de recursos o componentes estándar, abra el panel de propiedades del recurso.
2. Localice el campo de configuración de recursos. Este campo contiene el código de infraestructura de la tarjeta.
3. Selecciona el botón Generar sugerencias. Amazon Q generará una sugerencia.

### Note

El código generado en esta etapa no sobrescribirá el código de infraestructura existente de la plantilla.

4. Para generar más sugerencias, selecciona Regenerar. Puede alternar entre las muestras para comparar los resultados.
5. Para seleccionar una opción, elija Seleccionar. Puede modificar el código aquí antes de guardarlo en la aplicación. Para salir sin guardar, seleccione el icono de salida (X).
6. Para guardar el código en la plantilla de la aplicación, seleccione Guardar en el panel de propiedades del recurso.

## Más información

Para obtener más información Amazon Q, consulte [¿Qué es Amazon Q?](#) en el Amazon Q Developer Guía del usuario.

# Cómo componer en AWS Infrastructure Composer

En esta sección se describen los aspectos básicos del uso de Infrastructure Composer desde [Consola Infrastructure ComposerCloudFormation modo consola](#), y desde [AWS Toolkit for Visual Studio Code](#). Más específicamente, los temas de esta sección proporcionan detalles clave sobre cómo crear una aplicación con Infrastructure Composer e incluyen detalles sobre funciones y atajos adicionales. Existen algunas variaciones de funcionalidad entre las experiencias de consola y VS Code, y en los temas de esta sección se identifican y describen estas variaciones cuando se producen.

Después de crear la aplicación, estará listo [Implemente su aplicación sin servidor Infrastructure Composer en la nube AWS](#) para revisar la información sobre su implementación.

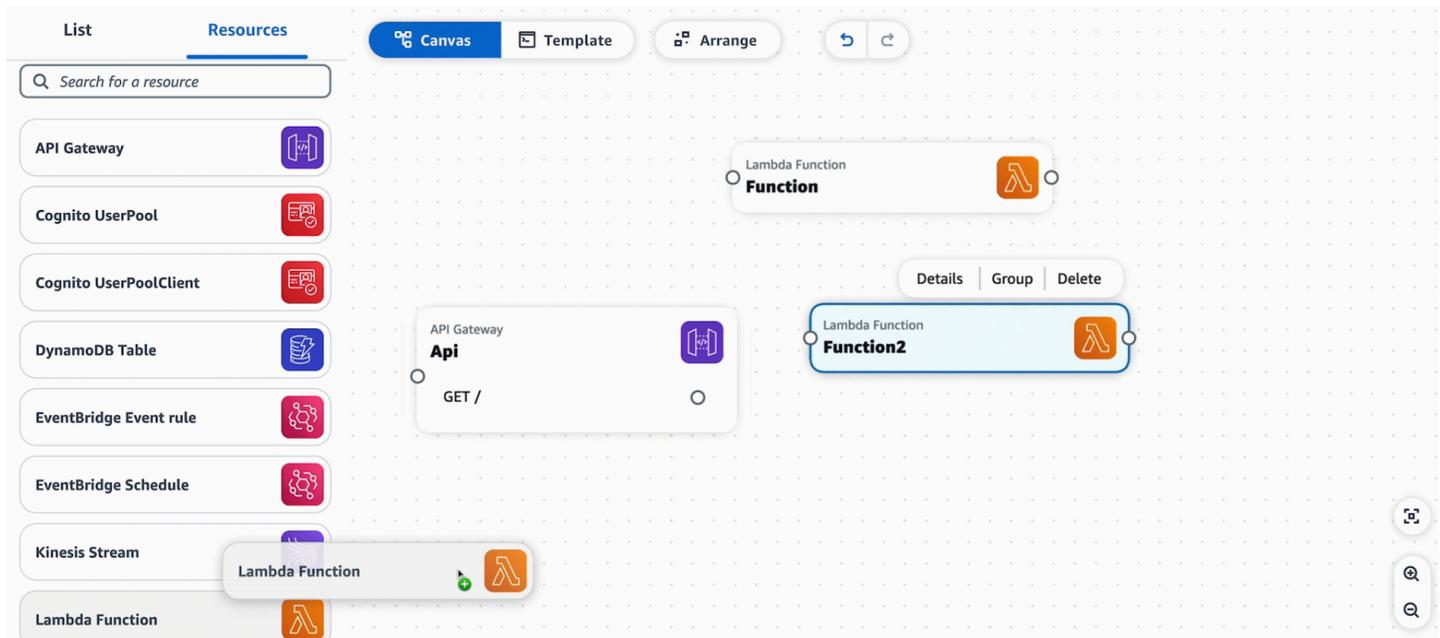
## Temas

- [Coloque las tarjetas en el lienzo visual de Infrastructure Composer](#)
- [Agrupa las tarjetas en el lienzo visual de Infrastructure Composer](#)
- [Connect tarjetas en el lienzo visual de Infrastructure Composer](#)
- [Desconectar tarjetas en Infrastructure Composer](#)
- [Organice las tarjetas en el lienzo visual de Infrastructure Composer](#)
- [Configurar y modificar tarjetas en Infrastructure Composer](#)
- [Eliminar tarjetas en Infrastructure Composer](#)
- [Vea las actualizaciones de código con el Inspector de cambios en Infrastructure Composer](#)
- [Archivos externos de referencia en Infrastructure Composer](#)
- [Integre Infrastructure Composer con Amazon Virtual Private Cloud \(AmazonVPC\)](#)

## Coloque las tarjetas en el lienzo visual de Infrastructure Composer

En esta sección se describe cómo se seleccionan y arrastran [las tarjetas](#) de Infrastructure Composer en su lienzo visual. Antes de empezar, identifique qué recursos necesita su aplicación y cómo deben interactuar. Para obtener consejos sobre cómo hacerlo, consulte [Cree su primera aplicación con Infrastructure Composer](#).

Para añadir una tarjeta a la aplicación, arrástrela desde la paleta de recursos y suéltela en el lienzo visual.



Puede elegir entre dos tipos de tarjetas: tarjetas con [componentes mejorados y tarjetas con recursos iAC estándar](#).

Tras colocar las cartas en el lienzo visual, estarás listo para agruparlas, conectarlas, organizarlas y configurarlas. Consulte los siguientes temas para obtener información sobre cómo hacerlo:

- [Agrupa las tarjetas en el lienzo visual de Infrastructure Composer](#)
- [Connect tarjetas en el lienzo visual de Infrastructure Composer](#)
- [Organice las tarjetas en el lienzo visual de Infrastructure Composer](#)
- [Configurar y modificar tarjetas en Infrastructure Composer](#)

## Agrupar las tarjetas en el lienzo visual de Infrastructure Composer

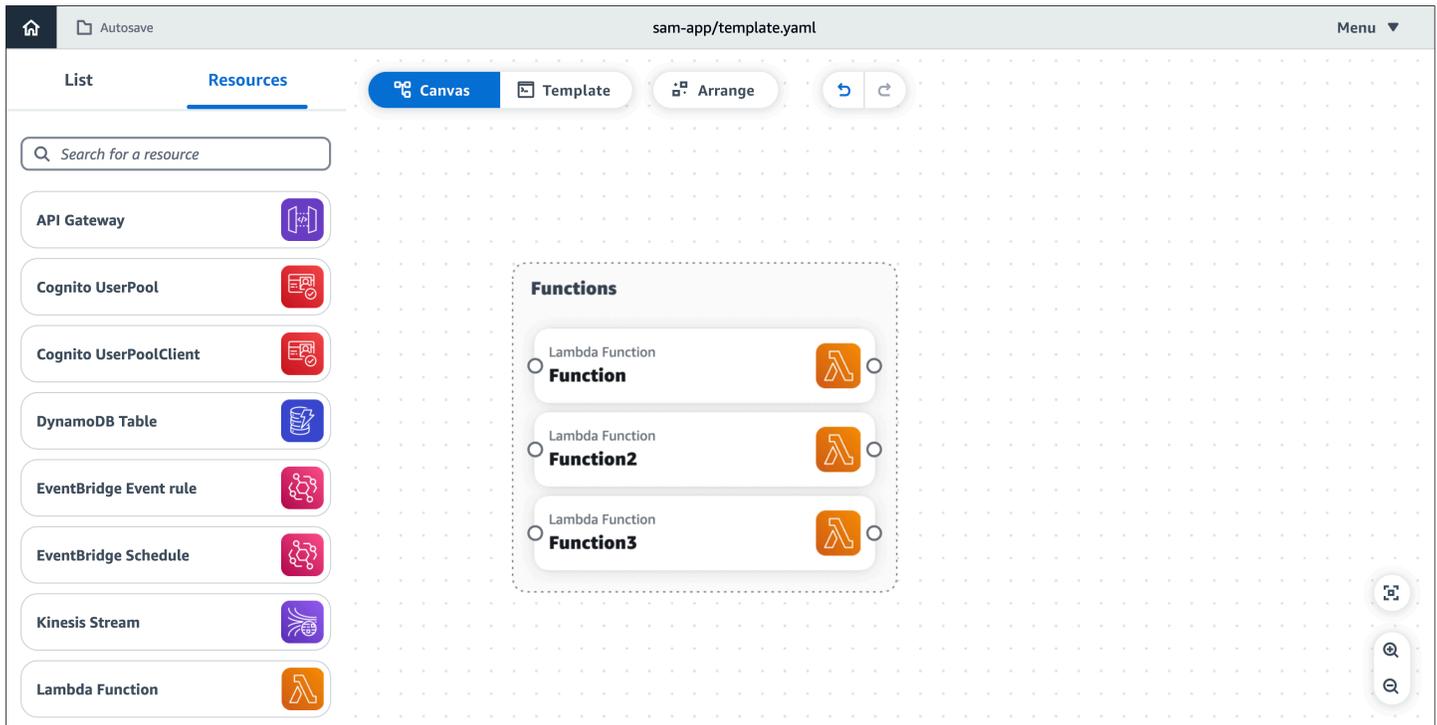
Este tema contiene detalles sobre cómo agrupar las tarjetas de componentes mejorados y las tarjetas de componentes estándar. Agrupar tarjetas le ayuda a clasificar y organizar sus recursos sin tener que pensar en el código o el marcado que debe escribir.

### Agrupar tarjetas de componentes mejorados

Hay dos formas de agrupar las tarjetas con componentes mejorados:

- Con la tecla Mayús pulsada, selecciona las cartas que deseas agrupar. A continuación, selecciona Agrupar en el menú de acciones del recurso.

- selecciona la carta que quieras incluir en un grupo. En el menú que aparece, selecciona Agrupar. Esto creará un grupo en el que podrás arrastrar y soltar otras cartas.



## Agrupar una tarjeta de componentes estándar en otra

El siguiente ejemplo muestra una forma de agrupar una tarjeta de un componente estándar en otra tarjeta desde el panel de propiedades del recurso:

The screenshot displays the AWS Infrastructure Composer interface. On the left, a canvas contains a 'Standard Component' card for 'Function'. The card includes a 'Role' component and a 'Function' component. On the right, the 'Resource properties' panel is open, showing the following details:

- Resource type:** AWS::Lambda::Function (CFN Resource)
- Editing:** Function
- Role:** (dropdown menu)
- Logical ID:** Function (with a checkmark icon)
- Logical ID description:** Updating this value will generate a new resource when this stack is updated.
- Resource configuration:** Updating this value will change the resource's properties. Replace all placeholder values before deploying.
- Code:** {}
- Role:** !Ref Role
- Resource reference:** (button with external link icon)

En el campo Configuración de recursos del panel de propiedades del recurso, se Role ha hecho referencia a ellos en la función Lambda. Esto hace que la carta de rol se agrupe en la tarjeta de función en el lienzo.

# Connect tarjetas en el lienzo visual de Infrastructure Composer

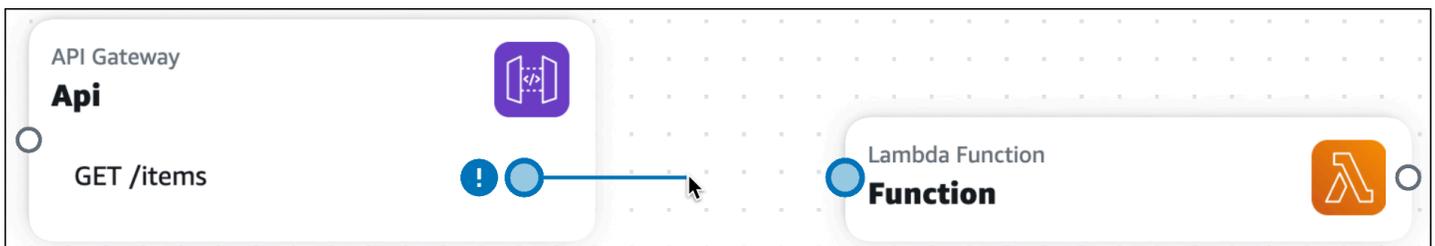
Utilice este tema para entender cómo conectar tarjetas en Infrastructure Composer. En esta sección se incluyen detalles sobre la conexión de tarjetas de componentes mejorados y tarjetas de componentes estándar. También proporciona algunos ejemplos que ilustran las diferentes formas en que se pueden conectar las tarjetas.

## Conexión de tarjetas de componentes mejorados

En las tarjetas de componentes mejoradas, los puertos identifican visualmente dónde se pueden realizar las conexiones.

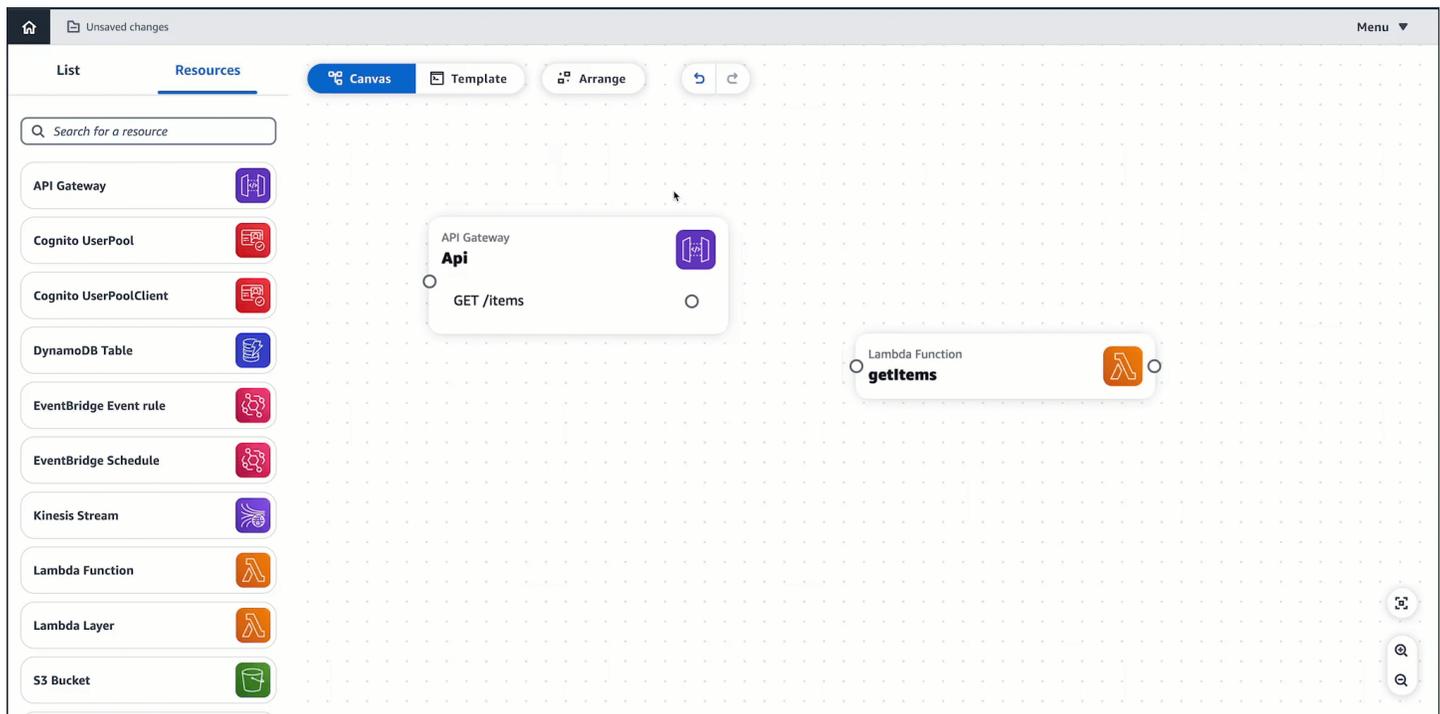
- Un puerto en el lado derecho de una tarjeta indica la posibilidad de que la tarjeta invoque a otra tarjeta.
- Un puerto en el lado izquierdo de una tarjeta indica la posibilidad de que otra tarjeta invoque la tarjeta.

Conecta tarjetas entre sí haciendo clic en el puerto derecho de una tarjeta y arrastrándolo hasta el puerto izquierdo de otra tarjeta.



Al crear una conexión, aparecerá un mensaje que le indicará si la conexión se ha realizado correctamente. Seleccione el mensaje para ver qué ha cambiado Infrastructure Composer para aprovisionar una conexión. Si la conexión no se realizó correctamente, puede seleccionar la vista de plantilla para actualizar manualmente el código de infraestructura y aprovisionar la conexión.

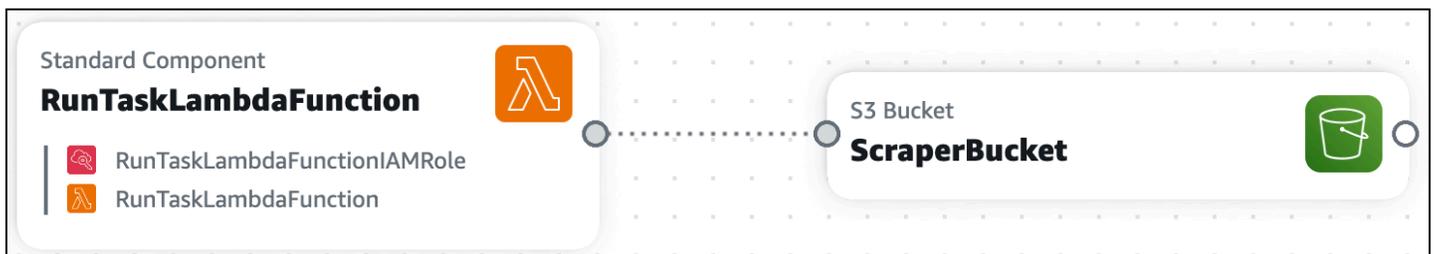
- Si se realiza correctamente, haga clic en el mensaje para ver el inspector de cambios. Aquí puede ver lo que Infrastructure Composer modificó para aprovisionar su conexión.
- Si no lo consigue, aparecerá un mensaje. Puede seleccionar la vista de plantilla y actualizar manualmente el código de infraestructura para aprovisionar la conexión.



Al conectar tarjetas de componentes mejoradas, Infrastructure Composer crea automáticamente el código de infraestructura en la plantilla para garantizar la relación entre los recursos basada en eventos.

## Conexión de tarjetas de componentes estándar (tarjetas de recursos iAC estándar)

Las tarjetas de recursos iAC estándar no incluyen puertos para crear conexiones con otros recursos. Durante la [configuración de la tarjeta](#), si especifica las relaciones basadas en eventos en la plantilla de la aplicación, Infrastructure Composer detectará automáticamente estas conexiones y las visualizará con una línea de puntos entre las tarjetas. El siguiente es un ejemplo de conexión entre una tarjeta de componentes estándar y una tarjeta de componentes mejorada:



El siguiente ejemplo muestra cómo se puede conectar una función Lambda con un resto de Amazon API Gateway: 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

```

En el ejemplo anterior, el fragmento de código que aparece `ApiGatewayMethod:` debajo `Integration:` especifica la relación basada en eventos que conecta las dos tarjetas.

## Ejemplos de conexión de tarjetas en Infrastructure Composer

Utilice los ejemplos de esta sección para entender cómo se pueden conectar las tarjetas en Infrastructure Composer.

### Invoca una AWS Lambda función cuando un artículo se coloca en un bucket de Amazon Simple Storage Service (Amazon S3)

En este ejemplo, una tarjeta bucket de Amazon S3 está conectada a una tarjeta de función Lambda. Cuando se coloca un artículo en el bucket de Amazon S3, se invoca la función. Luego, la función se puede usar para procesar el elemento o activar otros eventos en su aplicación.



Esta interacción requiere que se defina un evento para la función. Esto es lo que proporciona Infrastructure Composer:

```
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 un bucket de Amazon S3 desde una función Lambda

En este ejemplo, una tarjeta de función Lambda invoca una tarjeta de bucket de Amazon S3. La función Lambda se puede utilizar para realizar CRUD operaciones en los elementos del bucket de Amazon S3.



Esta interacción requiere lo siguiente, que es aprovisionado por Infrastructure Composer:

- IAM políticas que permiten que la función Lambda interactúe con el bucket de Amazon S3.
- Variables de entorno que influyen en el comportamiento de la función 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

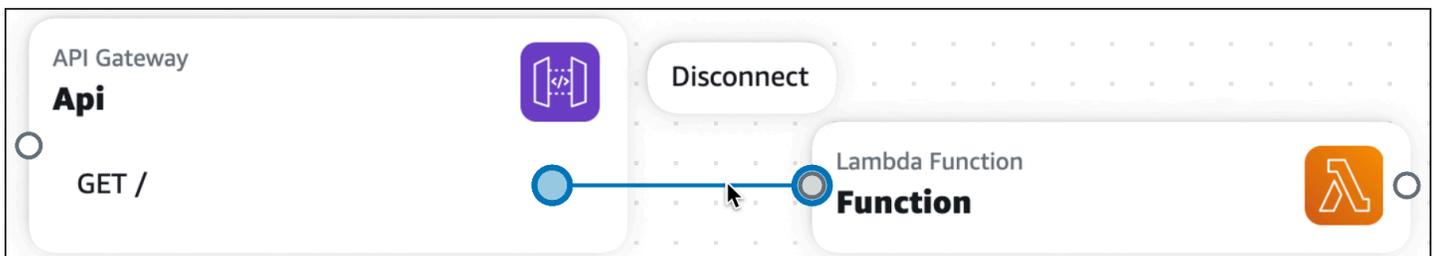
```

## Desconectar tarjetas en Infrastructure Composer

En Infrastructure Composer, AWS los recursos se conectan y desconectan mediante tarjetas de componentes mejorados y tarjetas de componentes estándar. En esta sección se describe cómo desconectar ambos tipos de tarjetas.

### Tarjetas con componentes mejorados

Para desconectar las tarjetas de componentes mejorados, seleccione la línea y elija Desconectar.



Infrastructure Composer modificará automáticamente la plantilla para eliminar de la aplicación la relación basada en eventos.

### Tarjetas de componentes estándar

Las tarjetas de componentes estándar no incluyen puertos para crear conexiones con otros recursos. Durante la [configuración de la tarjeta](#), si especifica las relaciones basadas en eventos en la plantilla

de la aplicación, Infrastructure Composer detectará automáticamente estas conexiones y las visualizará con una línea de puntos entre las tarjetas. Para desconectar una tarjeta de componentes estándar, elimine la relación basada en eventos de la plantilla de la aplicación.

El siguiente ejemplo muestra una función Lambda que está conectada a un resto de Amazon API Gateway: 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

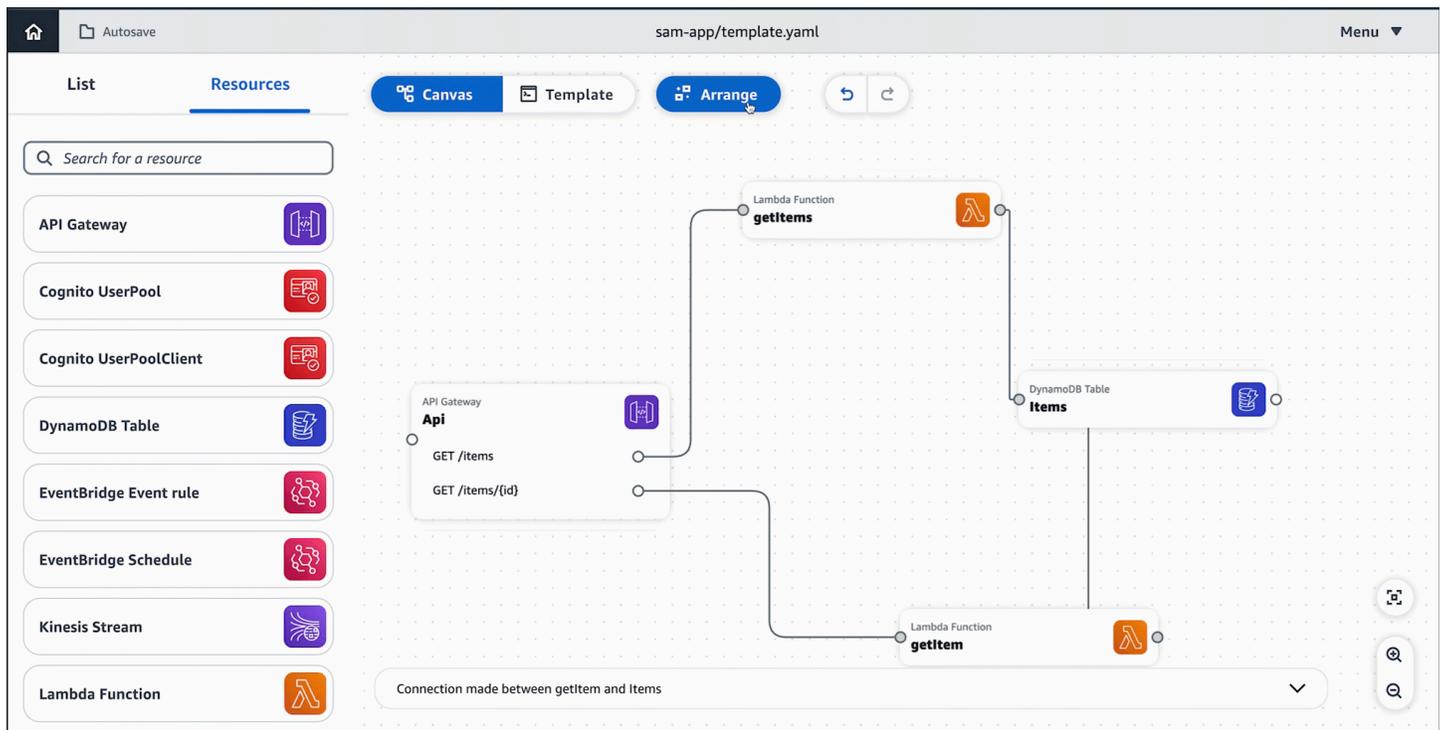
  LambdaExecutionRole:
```

```
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 eliminar la conexión entre las dos tarjetas, elimine las referencias que `MyLambdaFunction` aparecen a continuación. `ApiGatewayMethod: Integration`

## Organice las tarjetas en el lienzo visual de Infrastructure Composer

Seleccione Organizar para disponer y organizar visualmente las tarjetas en el lienzo. Utilizar el botón Organizar resulta especialmente útil cuando hay muchas tarjetas y conexiones en el lienzo.



## Configurar y modificar tarjetas en Infrastructure Composer

En Infrastructure Composer, las tarjetas representan los recursos que se utilizan para diseñar la arquitectura de la aplicación. Al configurar una tarjeta en Infrastructure Composer, se definen los detalles de los recursos de la aplicación. Esto incluye detalles como el identificador lógico y la clave de partición de la tarjeta. La forma en que se define esta información varía entre las tarjetas con componentes mejorados y las tarjetas estándar.

Una tarjeta con componentes mejorados es un conjunto de AWS CloudFormation recursos que se han combinado en una sola tarjeta seleccionada que mejora la facilidad de uso y la funcionalidad, y está diseñada para una amplia variedad de casos de uso. Una tarjeta de recursos iAC estándar representa un único AWS CloudFormation recurso. Cada tarjeta de recursos iAC estándar, una vez arrastrada al lienzo, se denomina Componente estándar.

En este tema se proporcionan detalles sobre la configuración de las tarjetas de componentes mejoradas y las tarjetas de componentes estándar.

### Note

Este tema se refiere al uso de tarjetas desde la consola de Infrastructure Composer, la AWS Toolkit for Visual Studio Code extensión y desde Infrastructure Composer en modo

de CloudFormation consola. Las tarjetas relacionadas con Lambda (función Lambda y capa Lambda) requieren compilaciones de código y soluciones de empaquetado que no están disponibles en Infrastructure Composer en modo consola. CloudFormation Para obtener más información, consulte [Uso de Infrastructure Composer en modo CloudFormation consola](#).

## Temas

- [Tarjetas de componentes mejoradas en Infrastructure Composer](#)
- [Tarjetas estándar en Infrastructure Composer](#)

## Tarjetas de componentes mejoradas en Infrastructure Composer

Para configurar las tarjetas de componentes mejoradas, Infrastructure Composer proporciona un formulario en el panel de propiedades del recurso. Este formulario está diseñado exclusivamente para guiarlo a través de la configuración de cada tarjeta de componentes mejorada. A medida que rellena el formulario, Infrastructure Composer modifica el código de infraestructura.

Algunas tarjetas de componentes mejoradas tienen funciones adicionales. En esta sección se analizan los aspectos básicos del uso de las cartas con componentes mejorados y se ofrecen detalles sobre las cartas con funciones adicionales.

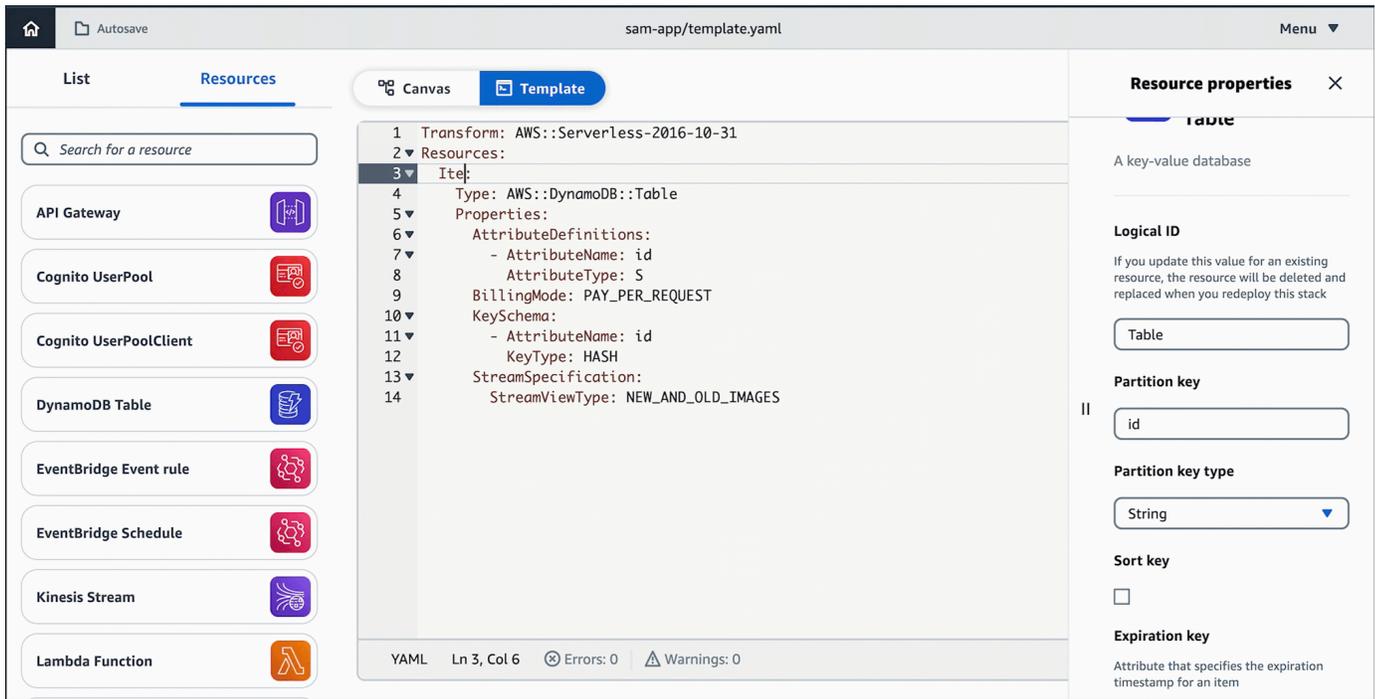
Para obtener más información sobre las tarjetas con componentes mejorados, consulte [Tarjetas de componentes mejoradas en Infrastructure Composer](#) y [Tarjetas de componentes mejoradas en Infrastructure Composer](#)

## Procedimiento

El panel de propiedades de los recursos agiliza la configuración y añade guías que simplifican la configuración de las tarjetas. Para usar este panel, lleve a cabo los siguientes pasos:

1. Haga doble clic en una tarjeta para abrir el panel de propiedades del recurso.
2. Haga clic en una tarjeta y seleccione Detalles para abrir el panel de propiedades del recurso.
3. En Infrastructure Composer AWS Management Console, seleccione Plantilla para mostrar el código de la aplicación. Configure directamente desde aquí.

La siguiente imagen muestra cómo se puede hacer esto:



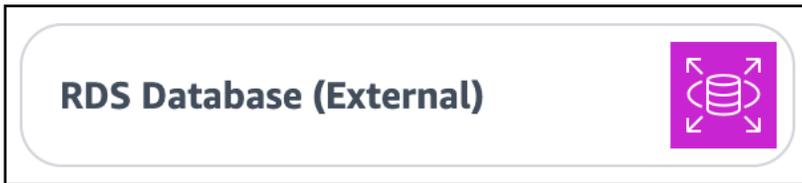
## Uso de Infrastructure Composer con Amazon Relational Database Service (AmazonRDS)

AWS Infrastructure Composer incluye una integración con Amazon Relational Database Service (RDSAmazon). Con la tarjeta de componentes mejorada de RDSbase de datos (externa) de Infrastructure Composer, puede conectar su aplicación a Amazon RDS DB clústeres, instancias y proxies que están definidos en otra plantilla AWS CloudFormation or AWS Serverless Application Model (AWS SAM).

La tarjeta de componentes mejorada de RDSbase de datos (externa) representa RDS los recursos de Amazon que están definidos en otra plantilla. Esto incluye:

- Amazon RDS DB clúster o instancia que está definido en otra plantilla
- Amazon RDS DB proxy

La tarjeta de componentes mejorada de la RDSbase de datos (externa) está disponible en la paleta de recursos.



Para usar esta tarjeta, arrástrela al lienzo de Infrastructure Composer, configúrela y conéctela a otros recursos.

Puedes conectar tu aplicación a la Amazon externa RDS DB clúster o instancia a través de una función Lambda.

### Requisitos

Para utilizar esta función, debe cumplir los siguientes requisitos:

1. Tu Amazon externo RDS DB Se debe utilizar un clúster, una instancia o un proxy AWS Secrets Manager para administrar la contraseña del usuario. Para obtener más información, consulta [Gestión de contraseñas con Amazon RDS y AWS Secrets Manager](#) en la Guía del RDS usuario de Amazon.
2. La aplicación en Infrastructure Composer debe ser un proyecto nuevo o debe haber sido creada originalmente en Infrastructure Composer.

### Procedimiento

#### Paso 1: Configurar la tarjeta de RDS base de datos externa

Desde la paleta Recursos, arrastre una tarjeta de componente mejorada de RDSbase de datos (externa) al lienzo.

Seleccione la tarjeta y elija Detalles o haga doble clic en la tarjeta para que aparezca el panel de propiedades del recurso. Aparecerá el panel de propiedades de los recursos de la tarjeta:

Details | Group | Delete

VPC  
 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}.

Aquí puede configurar lo siguiente:

- ID lógico: un nombre único para tu Amazon externo RDS DB clúster, instancia o proxy. Este ID no tiene que coincidir con el valor de ID lógico de tu Amazon externo RDS DB recurso.
- Secreto de base de datos: identificador del AWS Secrets Manager secreto asociado a tu Amazon RDS DB clúster, instancia o proxy. Este campo acepta los siguientes valores:
  - Valor estático: identificador único del secreto de la base de datos, como el secretoARN. A continuación se muestra un ejemplo: `arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-name-1a2b3c`. Para obtener más información, consulte [Conceptos de AWS Secrets Manager](#) en la Guía del usuario de la AWS Secrets Manager .

- Valor de salida: cuando se despliega un secreto de Secrets Manager AWS CloudFormation, se crea un valor de salida. Aquí puede especificar el valor de salida mediante la función [Fn::ImportValue](#) intrínseca. Por ejemplo, `!ImportValue MySecret`.
- Valor del almacén de SSM parámetros: puede almacenar su secreto en el almacén de SSM parámetros y especificar su valor mediante una referencia dinámica. Por ejemplo, `{{resolve:ssm:MySecret}}`. Para obtener más información, consulte [SSM los parámetros](#) en la Guía AWS CloudFormation del usuario.
- Nombre de host de la base de datos: el nombre de host que se puede usar para conectarse a Amazon RDS DB clúster, instancia o proxy. Este valor se especifica en la plantilla externa que define tu RDS recurso de Amazon. Se aceptan los siguientes valores:
  - Valor estático: identificador único del nombre de host de la base de datos, como la dirección del punto final. A continuación se muestra un ejemplo: `mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-2.rds.amazonaws.com`.
  - Valor de salida: el valor de salida de un Amazon desplegado RDS DB clúster, instancia o proxy. Puede especificar el valor de salida mediante la función [Fn::ImportValue](#) intrínseca. Por ejemplo, `!ImportValue myStack-myDatabase-abcd1234`.
  - Valor del almacén de SSM parámetros: puede almacenar el nombre de host de la base de datos en el almacén de SSM parámetros y especificar su valor mediante una referencia dinámica. Por ejemplo, `{{resolve:ssm:MyDatabase}}`.
- Puerto de base de datos: el número de puerto que se puede usar para conectarse a Amazon RDS DB clúster, instancia o proxy. Este valor se especifica en la plantilla externa que define tu RDS recurso de Amazon. Se aceptan los siguientes valores:
  - Valor estático: el puerto de la base de datos. Por ejemplo, `3306`.
  - Valor de salida: el valor de salida de un Amazon desplegado RDS DB clúster, instancia o proxy. Por ejemplo, `!ImportValue myStack-MyRDSInstancePort`.
  - Valor del almacén de SSM parámetros: puede almacenar el nombre de host de la base de datos en el almacén de SSM parámetros y especificar su valor mediante una referencia dinámica. Por ejemplo, `{{resolve:ssm:MyRDSInstancePort}}`.

#### Note

Aquí solo se debe configurar el valor de ID lógico. Si lo prefiere, puede configurar las demás propiedades en el momento de la implementación.

## Paso 2: Conectar una tarjeta de función Lambda

Desde la paleta Recursos, arrastre una tarjeta de componente mejorada de la función Lambda al lienzo.

Conecte el puerto izquierdo de la tarjeta de función Lambda al puerto derecho de la tarjeta de RDSbase de datos (externa).



Infrastructure Composer aprovisionará la plantilla para facilitar esta conexión.

¿Qué hace Infrastructure Composer para crear su conexión

Al completar el procedimiento indicado anteriormente, Infrastructure Composer realiza acciones específicas para conectar la función Lambda a la base de datos.

Al especificar el Amazon externo RDS DB clúster, instancia o proxy

Al arrastrar una tarjeta de RDSbase de datos (externa) al lienzo, Infrastructure Composer actualiza las Parameters secciones Metadata y secciones de la plantilla según sea necesario. A continuación, se muestra un ejemplo:

```
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
```

[Los metadatos](#) son una sección de la AWS CloudFormation plantilla que se utiliza para almacenar los detalles de la plantilla. Los metadatos específicos de Infrastructure Composer se almacenan en

la clave de `AWS::Composer::ExternalResources` metadatos. Aquí, Infrastructure Composer almacena los valores que especifique para su Amazon RDS DB clúster, instancia o proxy.

La sección de [parámetros](#) de una AWS CloudFormation plantilla se utiliza para almacenar valores personalizados que se pueden insertar en toda la plantilla durante la implementación. Según el tipo de valores que proporciones, Infrastructure Composer puede almacenar valores aquí para tu Amazon RDS DB agrupe, instancia o proxy y especifíquelos en toda la plantilla.

Los valores de cadena de la `Parameters` sección Metadata y utilizan el valor de identificador lógico que especifique en la tarjeta de RDSbase de datos (externa). Si actualiza el identificador lógico, los valores de la cadena cambiarán.

Al conectar la función Lambda a la base de datos

Al conectar una tarjeta de función Lambda a la tarjeta de RDSbase de datos (externa), Infrastructure Composer aprovisiona variables de entorno y AWS Identity and Access Management (IAM) políticas. A continuación, se muestra un ejemplo:

```
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
```

Las variables de [entorno](#) son variables que la función puede usar en tiempo de ejecución. Para obtener más información, consulte [Uso de variables de entorno Lambda](#) en la Guía para AWS Lambda desarrolladores.

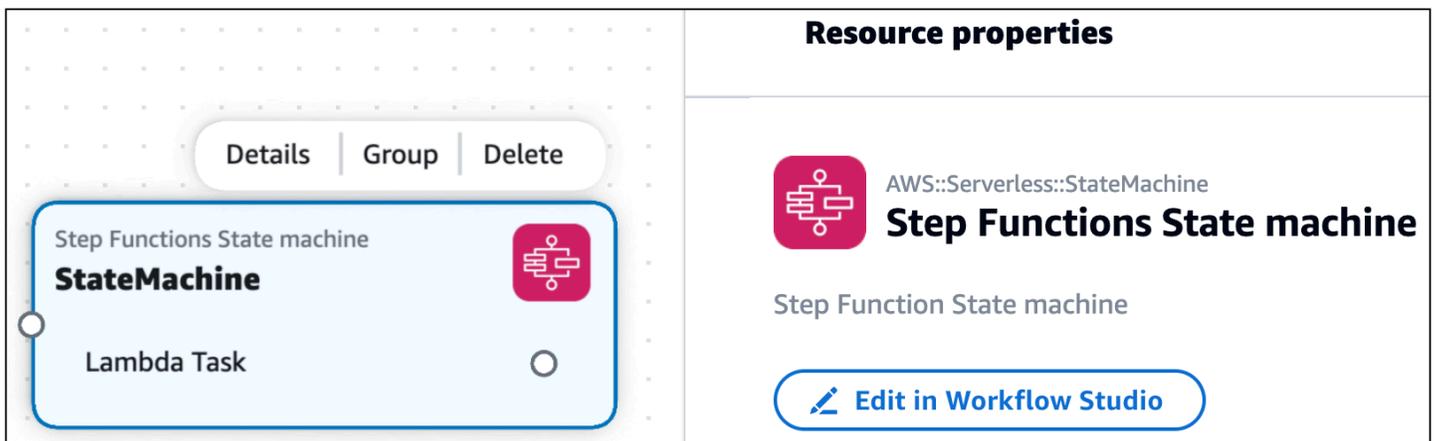
[Las políticas](#) proporcionan permisos para su función. Aquí, Infrastructure Composer crea una política para permitir el acceso de lectura desde tu función a Secrets Manager para obtener tu contraseña de acceso a Amazon RDS DB clúster, instancia o proxy.

## Uso AWS Infrastructure Composer con AWS Step Functions

AWS Infrastructure Composer presenta una integración con [AWS Step Functions Workflow Studio](#). Utilice Infrastructure Composer para hacer lo siguiente:

- Inicie Step Functions Workflow Studio directamente en Infrastructure Composer.
- Cree y gestione nuevos flujos de trabajo o importe los flujos de trabajo existentes a Infrastructure Composer.
- Integre sus flujos de trabajo con otros AWS recursos mediante el lienzo de Infrastructure Composer.

La siguiente imagen es de una tarjeta de máquina Step Functions State



Con Step Functions Workflow Studio en Infrastructure Composer, puede utilizar las ventajas de dos potentes diseñadores visuales en un solo lugar. Al diseñar el flujo de trabajo y la aplicación, Infrastructure Composer crea la infraestructura como código (IaC) para guiarlo hacia la implementación.

### Temas

- [Políticas de IAM](#)
- [Primeros pasos con Step Functions Workflow Studio en Infrastructure Composer](#)
- [Uso de Step Functions Workflow Studio en Infrastructure Composer](#)
- [Más información](#)

## Políticas de IAM

Al conectar las tareas de su flujo de trabajo con los recursos, Infrastructure Composer crea automáticamente las AWS Identity and Access Management (IAM) políticas necesarias para autorizar la interacción entre sus recursos. A continuación, se muestra un ejemplo:

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

Si es necesario, puede añadir más IAM políticas a la plantilla.

## Primeros pasos con Step Functions Workflow Studio en Infrastructure Composer

Para empezar, puede crear nuevos flujos de trabajo o importar los existentes.

Para crear un nuevo flujo de trabajo

1. Desde la paleta Recursos, arrastre una tarjeta de componentes mejorada de Step Functions State Machine al lienzo.



Al arrastrar una tarjeta de máquina Step Functions State al lienzo, Infrastructure Composer crea lo siguiente:

- Un [AWS::Serverless::StateMachine](#) recurso que define su máquina de estados. De forma predeterminada, Infrastructure Composer crea un flujo de trabajo estándar. Para crear un flujo de trabajo rápido, cambie el Type valor de la plantilla de STANDARD a EXPRESS.

- [AWS::Logs::LogGroup](#) Recurso que define un grupo de CloudWatch registros de Amazon para su máquina de estado.
2. Abre el panel de propiedades del recurso de la tarjeta y selecciona Editar en Workflow Studio para abrirlo Workflow Studio en Infrastructure Composer.

Step Functions Workflow Studio se abre en modo Diseño. Para obtener más información, consulte el [modo Diseño](#) en la Guía para AWS Step Functions desarrolladores.

 Note

Puede modificar Infrastructure Composer para guardar la definición de la máquina de estados en un archivo externo. Para obtener más información, consulte [Trabajar con archivos externos](#).

3. Cree su flujo de trabajo y seleccione Guardar. Para salir Workflow Studio, seleccione Volver a Infrastructure Composer.

Infrastructure Composer define el flujo de trabajo mediante la `Definition` propiedad del `AWS::Serverless::StateMachine` recurso.

4. Puede modificar el flujo de trabajo mediante cualquiera de las siguientes acciones:
  - Abra Workflow Studio de nuevo y modifique su flujo de trabajo.
  - Para Infrastructure Composer desde la consola, puede abrir la vista de plantilla de la aplicación y modificar la plantilla. Si utiliza la sincronización local, puede modificar el flujo de trabajo en su entorno local IDE. Infrastructure Composer detectará los cambios y actualizará el flujo de trabajo en Infrastructure Composer.
  - En el caso de Infrastructure Composer, del Toolkit for VS Code, puede modificar directamente la plantilla. Infrastructure Composer detectará los cambios y actualizará el flujo de trabajo en Infrastructure Composer.

Para importar los flujos de trabajo existentes

Puede importar flujos de trabajo desde aplicaciones definidas mediante plantillas AWS Serverless Application Model (AWS SAM). Use cualquier máquina de estados definida con el tipo de `AWS::Serverless::StateMachine` recurso y se visualizará como una tarjeta de componentes mejorada de Step Functions State Machine que puede usar para iniciar Workflow Studio.

El `AWS::Serverless::StateMachine` recurso puede definir flujos de trabajo mediante cualquiera de las siguientes propiedades:

- [Definition](#)— El flujo de trabajo se define en la AWS SAM plantilla como un objeto.
- [DefinitionUri](#)— El flujo de trabajo se define en un archivo externo en el [idioma de Amazon States](#). A continuación, se especifica la ruta local del archivo con esta propiedad.

### Propiedad de definición

#### Infrastructure Composer desde la consola

Para los flujos de trabajo definidos mediante la `Definition` propiedad, puede importar una sola plantilla o todo el proyecto.

- Plantilla: para obtener instrucciones sobre la importación de una plantilla, consulte [Importe una plantilla de proyecto existente en la consola de Infrastructure Composer](#). Para guardar los cambios que realice en Infrastructure Composer, debe exportar la plantilla.
- Proyecto: al importar un proyecto, debe activar la sincronización local. Los cambios que realices se guardan automáticamente en tu máquina local. Para obtener instrucciones sobre la importación de un proyecto, consulte [Importe una carpeta de proyecto existente en la consola de Infrastructure Composer](#).

#### Infrastructure Composer del kit de herramientas para VS Code

Para los flujos de trabajo definidos mediante la `Definition` propiedad, puede abrir Infrastructure Composer desde su plantilla. Para obtener instrucciones, consulte [Acceda a Infrastructure Composer desde AWS Toolkit for Visual Studio Code](#).

### DefinitionUri propiedad

#### Infrastructure Composer desde la consola

Para los flujos de trabajo definidos mediante la `DefinitionUri` propiedad, debe importar el proyecto y activar la sincronización local. Para obtener instrucciones sobre la importación de un proyecto, consulte [Importe una carpeta de proyecto existente en la consola de Infrastructure Composer](#).

## Infraestructure Composer del kit de herramientas para VS Code

Para los flujos de trabajo definidos mediante la `DefinitionUri` propiedad, puede abrir Infraestructure Composer desde su plantilla. Para obtener instrucciones, consulte [Acceda a Infraestructure Composer desde AWS Toolkit for Visual Studio Code](#).

## Uso de Step Functions Workflow Studio en Infraestructure Composer

### Cree flujos de trabajo

Infraestructure Composer utiliza sustituciones de definiciones para asignar las tareas del flujo de trabajo a los recursos de la aplicación. Para obtener más información sobre las sustituciones de definiciones, consulte [DefinitionSubstitutions](#) la Guía para AWS Serverless Application Model desarrolladores.

Al crear tareas en Workflow Studio, especifique una sustitución de definición para cada tarea. A continuación, puede conectar las tareas a los recursos del lienzo de Infraestructure Composer.

Para especificar una sustitución de definición en Workflow Studio

1. Abra la pestaña Configuración de la tarea y busque el campo APIParámetros.

**Check Stock Value** Definition

**Configuration** | Input | Output | Error handling

**State name**  
Check Stock Value

**API**  
Lambda: Invoke

**Integration type** [Info](#)  
The type of service integration to use. [Learn more](#)

Optimized

**API Parameters** Edit as JSON

**Function name**  
The Lambda function to invoke

Enter a Cloudformation substitution  
Substitutions can be used to parameterize your workflow definition which will be...

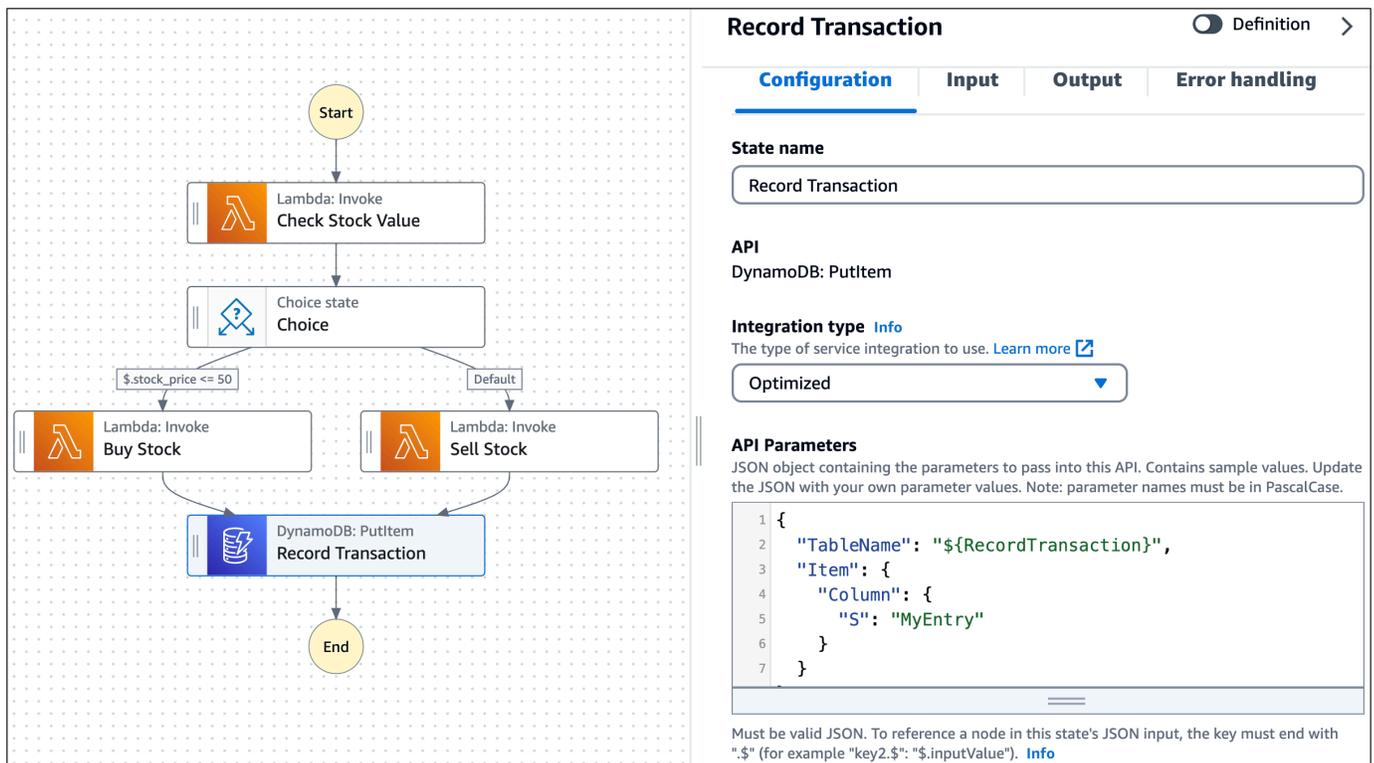
`${LambdaFunction1}`

Substitutions must be specified in `$(dollar_sign_brace)` notation. They will be mapped via the `DefinitionSubstitution` property inside your `StateMachine` resource in the Application Composer Canvas.

- Si el campo APIParámetros tiene una opción desplegable, seleccione Introducir una AWS CloudFormation sustitución. A continuación, proporcione un nombre exclusivo.

Para las tareas que se conectan al mismo recurso, especifique la misma sustitución de definición para cada tarea. Para usar una sustitución de definición existente, elija Seleccione una AWS CloudFormation sustitución y seleccione la sustitución que desee utilizar.

- Si el campo APIParámetros contiene un JSON objeto, modifique la entrada que especifica el nombre del recurso para utilizar una sustitución por definición. En el siguiente ejemplo, cambiamos "MyDynamoDBTable" a "\${RecordTransaction}".



**Record Transaction** Definition

**Configuration** | Input | Output | Error handling

**State name**  
Record Transaction

**API**  
DynamoDB: PutItem

**Integration type** [Info](#)  
The type of service integration to use. [Learn more](#)

Optimized

**API Parameters**  
JSON object containing the parameters to pass into this API. Contains sample values. Update the JSON with your own parameter values. Note: parameter names must be in PascalCase.

```

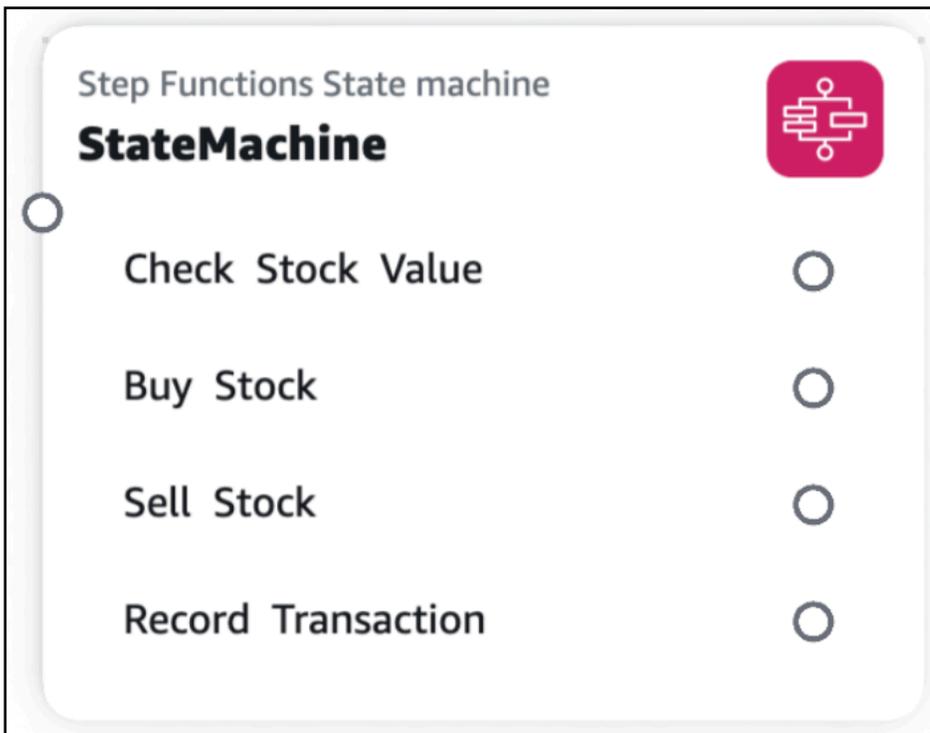
1 {
2   "TableName": "${RecordTransaction}",
3   "Item": {
4     "Column": {
5       "S": "MyEntry"
6     }
7   }

```

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

- Seleccione Guardar y volver a Infrastructure Composer.

Las tareas de su flujo de trabajo se visualizarán en la tarjeta de máquina Step Functions State.



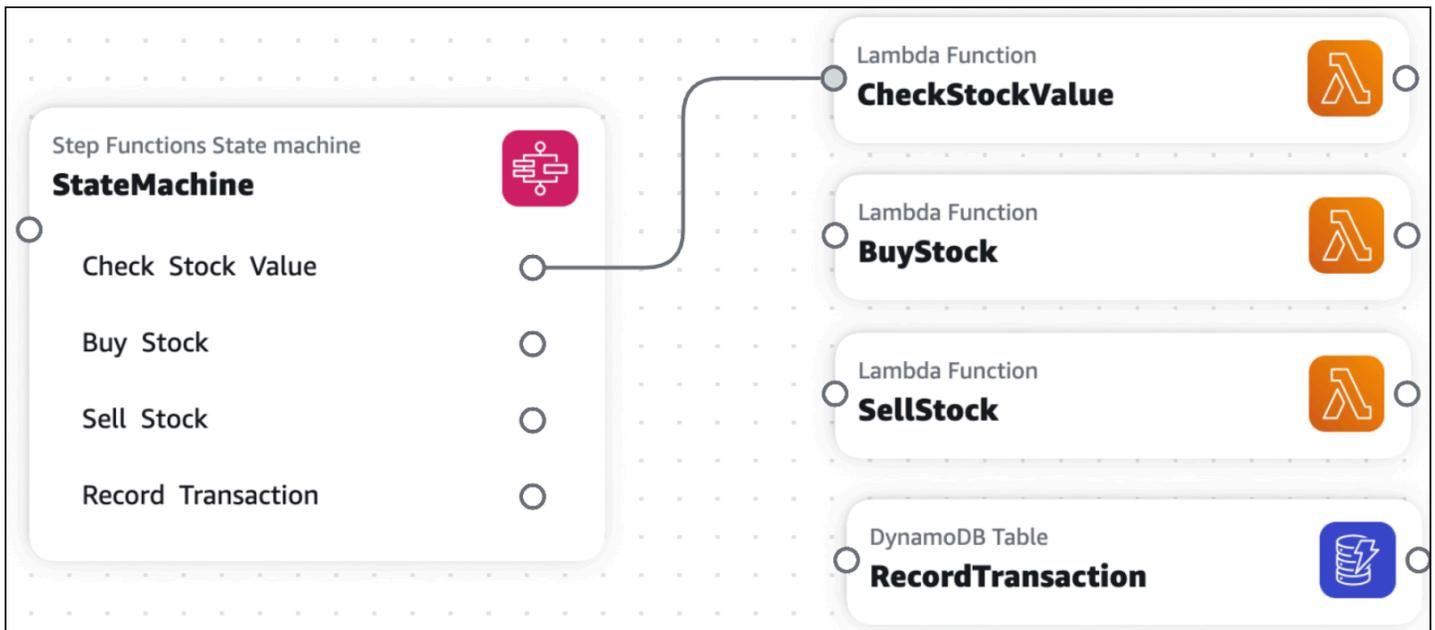
Connect los recursos a las tareas del flujo de trabajo

Puede crear conexiones en Infrastructure Composer entre las tareas de flujo de trabajo compatibles y las tarjetas de Infrastructure Composer compatibles.

- Tareas de flujo de trabajo compatibles: las tareas están optimizadas para Step Functions. Servicios de AWS Para obtener más información, consulte [Integraciones optimizadas para Step Functions](#) en la Guía para AWS Step Functions desarrolladores.
- Tarjetas Infrastructure Composer compatibles: se admiten tarjetas de componentes mejoradas. Para obtener más información sobre las tarjetas en Infrastructure Composer, consulte [Configurar y modificar tarjetas en Infrastructure Composer](#).

Al crear una conexión, Servicio de AWS la tarea y la tarjeta deben coincidir. Por ejemplo, puede conectar una tarea de flujo de trabajo que invoca una función Lambda a una tarjeta de componentes mejorada de la función Lambda.

Para crear una conexión, haga clic en el puerto de una tarea y arrástrelo hasta el puerto izquierdo de una tarjeta de componentes mejorada.



Infrastructure Composer actualizará automáticamente DefinitionSubstitution el valor para definir la conexión. A continuación, se muestra un ejemplo:

```

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:
        ...

```

## Trabajar con archivos externos

Al crear un flujo de trabajo a partir de la tarjeta de máquina de estado de Step Functions, Infrastructure Composer guarda la definición de la máquina de estados en la plantilla mediante la `Definition` propiedad. Puede configurar Infrastructure Composer para guardar la definición de la máquina de estados en un archivo externo.

### Note

Para utilizar esta función con Infrastructure Composer desde AWS Management Console, debe tener activada la sincronización local. Para obtener más información, consulte [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#).

Para guardar la definición de la máquina de estados en un archivo externo

1. Abra el panel de propiedades del recurso de su tarjeta de máquina Step Functions State.
2. Seleccione la opción Usar un archivo externo para la definición de la máquina de estados.
3. Proporcione una ruta y un nombre relativos para el archivo de definición de la máquina de estado.
4. Seleccione Guardar.

Infrastructure Composer hará lo siguiente:

1. Mueva la definición de la máquina de estados del `Definition` campo al archivo externo.
2. Guarda tu definición de máquina de estados en un archivo externo utilizando Amazon States Language.
3. Modifique su plantilla para hacer referencia al archivo externo mediante el `DefinitionUri` campo.

### Más información

Para obtener más información sobre Step Functions en Infrastructure Composer, consulte lo siguiente:

- [Utilizando Workflow Studio en Infrastructure Composer](#) en la Guía para AWS Step Functions desarrolladores.

- [DefinitionSubstitutions en AWS SAM las plantillas](#) de la Guía para AWS Step Functions desarrolladores.

## Tarjetas estándar en Infrastructure Composer

Todos los AWS CloudFormation recursos están disponibles para su uso como tarjetas de recursos estándar de laC en la paleta de recursos. Tras arrastrarla al lienzo visual, una tarjeta de recursos laC estándar se convierte en una tarjeta componente estándar. Esto simplemente significa que la tarjeta es uno o más recursos iAc estándar. Para obtener más ejemplos y detalles, consulte los temas de esta sección.

Puede modificar el código de infraestructura mediante la vista de plantillas y la ventana de propiedades del recurso. Por ejemplo, el siguiente es un ejemplo de plantilla inicial de un recurso de laC Alexa::ASK::Skill estándar:

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

La plantilla inicial de una tarjeta de recursos de laC estándar consta de lo siguiente:

- El tipo AWS CloudFormation de recurso.
- Propiedades obligatorias o de uso común.
- El tipo de valor que se debe proporcionar a cada propiedad.

**Note**

Puede usar... Amazon Q para generar sugerencias de códigos de infraestructura para las tarjetas de recursos estándar. Para obtener más información, consulte [Uso AWS Infrastructure Composer con Amazon Q Developer](#).

## Procedimiento

Puede modificar el código de infraestructura de cada recurso de una tarjeta de componentes estándar a través del panel de propiedades del recurso.

Para modificar una tarjeta de componentes estándar

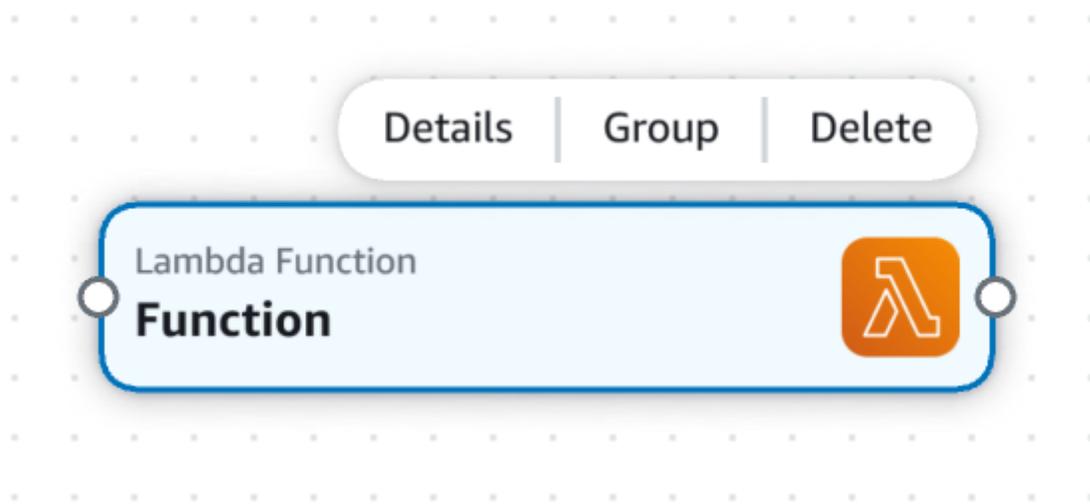
1. Abra el panel de propiedades del recurso de la tarjeta de componentes iAC estándar.
2. En el campo Edición, seleccione el recurso laC estándar que desee editar en la lista desplegable.
3. Modifique el código de infraestructura y guárdelo.

## Eliminar tarjetas en Infrastructure Composer

En esta sección se proporcionan instrucciones para eliminar tarjetas en AWS Infrastructure Composer.

### Tarjetas con componentes mejorados

Para eliminar una tarjeta de componentes mejorada, seleccione una tarjeta que haya colocado en el lienzo visual. En el menú de acciones de la tarjeta, selecciona Eliminar.



## Tarjetas con componentes estándar

Para eliminar las tarjetas de componentes estándar, debe eliminar manualmente el código de infraestructura de cada AWS CloudFormation recurso de la plantilla. La siguiente es una forma sencilla de hacerlo:

1. Tome nota del identificador lógico del recurso que se va a eliminar.
2. En la plantilla, localice el recurso por su identificador lógico en la Outputs sección Resources o.
3. Elimine el recurso de la plantilla. Esto incluye el identificador lógico del recurso y sus valores anidados, como Type y Properties.
4. Compruebe la vista de Canvas para comprobar que el recurso se ha eliminado de su lienzo.

## Vea las actualizaciones de código con el Inspector de cambios en Infrastructure Composer

A medida que diseña en la consola de Infrastructure Composer, el código de infraestructura se crea automáticamente. Utilice el Inspector de cambios para ver las actualizaciones del código de la plantilla y obtener información sobre lo que Infrastructure Composer está creando para usted.

En este tema se describe el uso de Infrastructure Composer desde la extensión AWS Management Console o la AWS Toolkit for Visual Studio Code extensión.

El Inspector de cambios es una herramienta visual de Infrastructure Composer que muestra las actualizaciones de código recientes.

- Al diseñar la aplicación, los mensajes aparecen en la parte inferior del lienzo visual. Estos mensajes proporcionan comentarios sobre las acciones que está realizando.
- Si lo admite, puede expandir un mensaje para ver el Inspector de cambios.
- El Inspector de cambios muestra los cambios de código de la interacción más reciente.

El siguiente ejemplo muestra cómo funciona el inspector de cambios:

The screenshot displays the AWS Infrastructure Composer interface. On the left, there is a 'Resources' panel with a search bar and a list of resources 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 shows a visual representation of a 'Lambda Function' connected to an 'S3 Bucket'. A 'Change Inspector' window is open, showing the following code snippet:

```

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 +

```

Below the code, it indicates '1 of 1 changes'. On the right side, the 'Resource properties' panel for the 'S3 Bucket' is visible, showing details like 'Stores files', 'Logical ID' (Bucket), and 'Block Public Access' (checked).

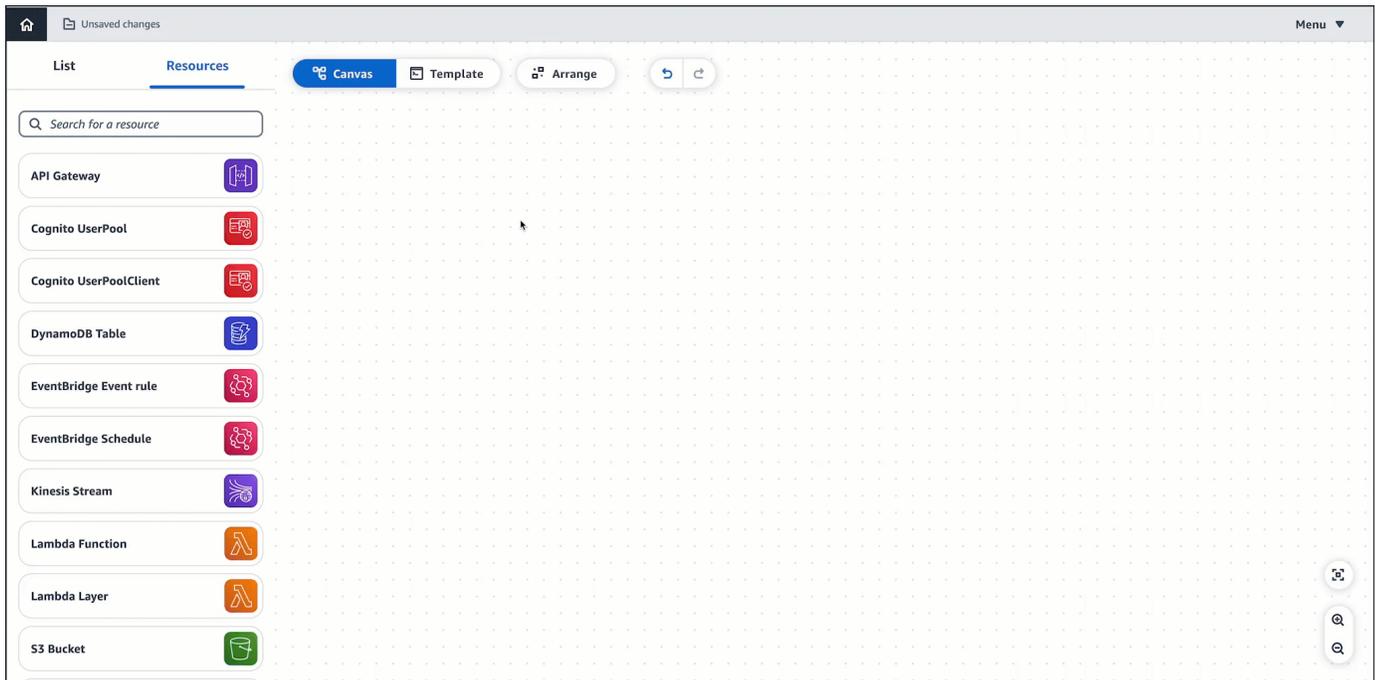
## Ventajas del Inspector de Cambios

El Inspector de cambios es una excelente forma de ver el código de plantilla que Infrastructure Composer crea para usted. También es una excelente forma de aprender a escribir código de infraestructura. Al diseñar aplicaciones en Infrastructure Composer, consulte las actualizaciones de código en el Inspector de cambios para obtener información sobre el código necesario para aprovisionar el diseño.

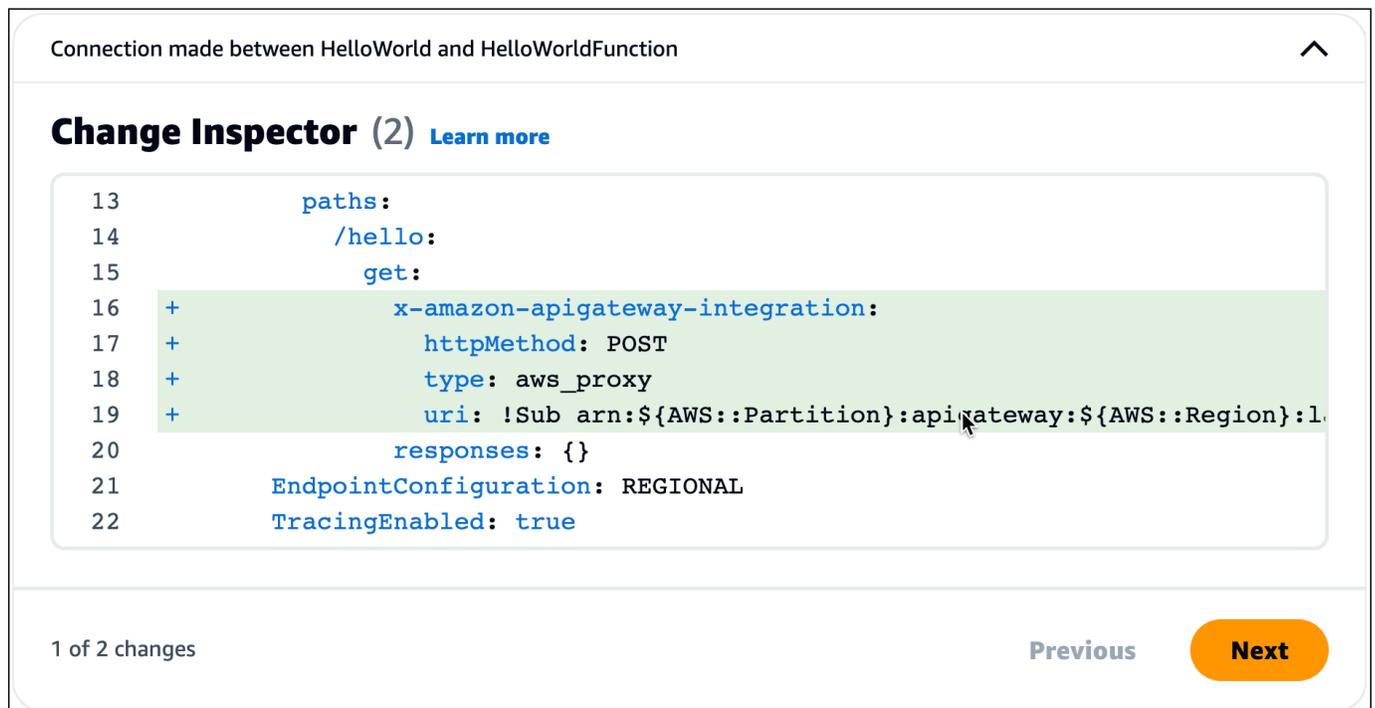
## Procedimiento

Para usar el Inspector de cambios

1. Amplíe un mensaje para que aparezca el Inspector de cambios.



2. Vea el código que se ha redactado automáticamente para usted.



- El código resaltado en verde indica el código recién agregado.
- El código resaltado en rojo indica el código recién eliminado.
- Los números de línea indican la ubicación dentro de la plantilla.

3. Cuando se han actualizado varias secciones de la plantilla, el Inspector de cambios las organiza. Selecciona los botones Anterior y Siguiente para ver todos los cambios.



### Note

En el caso de Infrastructure Composer, desde la consola, puede ver los cambios de código en el contexto de toda la plantilla mediante la vista de plantillas. También puede sincronizar Infrastructure Composer con una IDE versión local y ver la plantilla completa en su máquina local. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su consola local IDE](#).

## Más información

Para obtener más información sobre el código que crea Infrastructure Composer, consulte lo siguiente:

- [Conexiones de tarjetas en Infrastructure Composer](#).

# Archivos externos de referencia en Infrastructure Composer

Puede usar archivos externos con sus plantillas AWS Serverless Application Model (AWS SAM) para reutilizar el código repetido y organizar sus proyectos. Por ejemplo, es posible que tenga varios REST API recursos de Amazon API Gateway descritos en un OpenAPI especificación. En lugar de replicar el OpenAPI el código de especificación de la plantilla, puede crear un archivo externo y hacer referencia a él para cada uno de sus recursos.

AWS Infrastructure Composer admite los siguientes casos de uso de archivos externos:

- Gateway de API REST API recursos definidos por recursos externos OpenAPI archivos de especificaciones.
- AWS Step Functions recursos de máquina de estado definidos por archivos de definición de máquina de estado externos.

Para obtener más información sobre la configuración de archivos externos para los recursos compatibles, consulte lo siguiente:

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

## Note

Para hacer referencia a archivos externos con Infrastructure Composer desde la consola de Infrastructure Composer, debe usar Infrastructure Composer en el modo de sincronización local. Para obtener más información, consulte [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#).

## Temas

- [Prácticas recomendadas para los archivos de referencia externos de Infrastructure Composer](#)
- [Cree una referencia de archivo externo en Infrastructure Composer](#)
- [Cargue un proyecto con una referencia de archivo externa en Infrastructure Composer](#)
- [Cree una aplicación que haga referencia a un archivo externo en Infrastructure Composer](#)
- [Haga referencia a un OpenAPI archivo externo de especificaciones con Infrastructure Composer](#)

## Prácticas recomendadas para los archivos de referencia externos de Infrastructure Composer

### Utilice Infrastructure Composer con un dispositivo local IDE

Cuando usa Infrastructure Composer con un local IDE en modo de sincronización local, puede usar el local IDE para ver y modificar archivos externos. El contenido de los archivos externos compatibles a los que se hace referencia en la plantilla se actualizará automáticamente en el lienzo de Infrastructure Composer. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su consola local IDE](#).

### Mantenga los archivos externos en el directorio principal del proyecto

Puedes crear subdirectorios dentro del directorio principal de tu proyecto para organizar los archivos externos. Infrastructure Composer no puede acceder a los archivos externos que están almacenados en un directorio fuera del directorio principal del proyecto.

### Implemente la aplicación mediante el AWS SAM CLI

Al implementar la aplicación en Nube de AWS, los archivos externos locales deben cargarse primero en una ubicación accesible, como Amazon Simple Storage Service (Amazon S3). Puede utilizar el AWS SAM CLI para facilitar este proceso de forma automática. Para obtener más información, consulte [Cargar archivos locales durante la implementación en](#) la Guía para AWS Serverless Application Model desarrolladores.

## Cree una referencia de archivo externo en Infrastructure Composer

Puede crear una referencia a un archivo externo desde el panel de propiedades del recurso de los recursos compatibles.

Para crear una referencia a un archivo externo

1. En una tarjeta de componentes mejorada de APIGateway o Step Functions, seleccione Detalles para que aparezca el panel de propiedades del recurso.
2. Localice y seleccione la opción Usar archivo externo.
3. Especifique la ruta relativa al archivo externo. Esta es la ruta del `template.yaml` archivo al archivo externo.

Por ejemplo, para hacer referencia al archivo `api-spec.yaml` externo desde la siguiente estructura del proyecto, especifique `./api-spec.yaml` la ruta relativa.

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

#### Note

Si el archivo externo y la ruta especificada no existen, Infrastructure Composer los creará.

4. Guarde los cambios.

## Cargue un proyecto con una referencia de archivo externa en Infrastructure Composer

Siga los pasos que se indican en esta página para cargar un proyecto de Infrastructure Composer con una referencia de archivo externa.

Desde la consola de Infrastructure Composer

1. Realice los pasos que se indican en [Importe una plantilla de proyecto existente en la consola de Infrastructure Composer](#).
2. Confirme que Infrastructure Composer le pida que se conecte a la carpeta raíz del proyecto

Si su navegador admite el acceso al sistema de archivos API, Infrastructure Composer le pedirá que se conecte a la carpeta raíz del proyecto. Infrastructure Composer abrirá el proyecto en modo de sincronización local para admitir el archivo externo. Si el archivo externo al que se hace referencia no es compatible, recibirá un mensaje de error. Para obtener más información sobre los mensajes de error, consulte [Solución de problemas](#).

## Del kit de herramientas para VS Code

1. Realice los pasos que se indican en [Acceda a Infrastructure Composer desde AWS Toolkit for Visual Studio Code](#).
2. Abra la plantilla que desee ver en Infrastructure Composer.

Al acceder a Infrastructure Composer desde una plantilla, Infrastructure Composer detectará automáticamente el archivo externo. Si el archivo externo al que se hace referencia no es compatible, recibirá un mensaje de error. Para obtener más información sobre los mensajes de error, consulte [Solución de problemas](#).

## Cree una aplicación que haga referencia a un archivo externo en Infrastructure Composer

En este ejemplo se utiliza el AWS SAM CLI para crear una aplicación que haga referencia a un archivo externo para su definición de máquina de estados. A continuación, cargue el proyecto en Infrastructure Composer con el archivo externo debidamente referenciado.

### Ejemplo

1. En primer lugar, utilice el AWS SAM CLI `sam init` comando para inicializar una nueva aplicación llamada `demo`. Durante el flujo interactivo, seleccione la plantilla de inicio rápido del flujo de trabajo de varios pasos.

```
$ 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
```

```
...
```

Esta aplicación hace referencia a un archivo externo para la definición de la máquina de estados.

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

El archivo externo se encuentra en el `statemachine` subdirectorio de nuestra aplicación.

```
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. A continuación, cargue la aplicación en Infrastructure Composer desde la consola. En la página de inicio de Infrastructure Composer, seleccione Cargar una CloudFormation plantilla.
3. Seleccione nuestra carpeta de demo proyectos y deje que aparezca el mensaje para ver los archivos. Seleccione nuestro `template.yaml` archivo y seleccione Crear. Cuando se te pida, seleccione Guardar cambios.

## 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

Infrastructure Composer detecta automáticamente el archivo de definición de la máquina de estado externa y lo carga. Seleccione nuestro `StockTradingStateMachinerecurso` y elija `Detalles` para mostrar el panel de propiedades del recurso. Aquí puede ver que Infrastructure Composer se ha conectado automáticamente a nuestro archivo de definición de máquina de estado externa.

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' list with various AWS services like API Gateway, Cognito UserPool, DynamoDB Table, etc. The main canvas displays a collection of resources including Lambda Functions (StockCheckerFunction, StockSellerFunction, StockBuyerFunction), a DynamoDB Table (TransactionTable), and an EventBridge Timer (ImplicitTimer). A 'StockTradingStateMachine' resource is highlighted, and its 'Details' panel is open on the right. This panel shows the state machine definition in JSON format, including a comment, startAt, states, and a retry configuration. The 'Use external file for state machine definition' checkbox is checked.

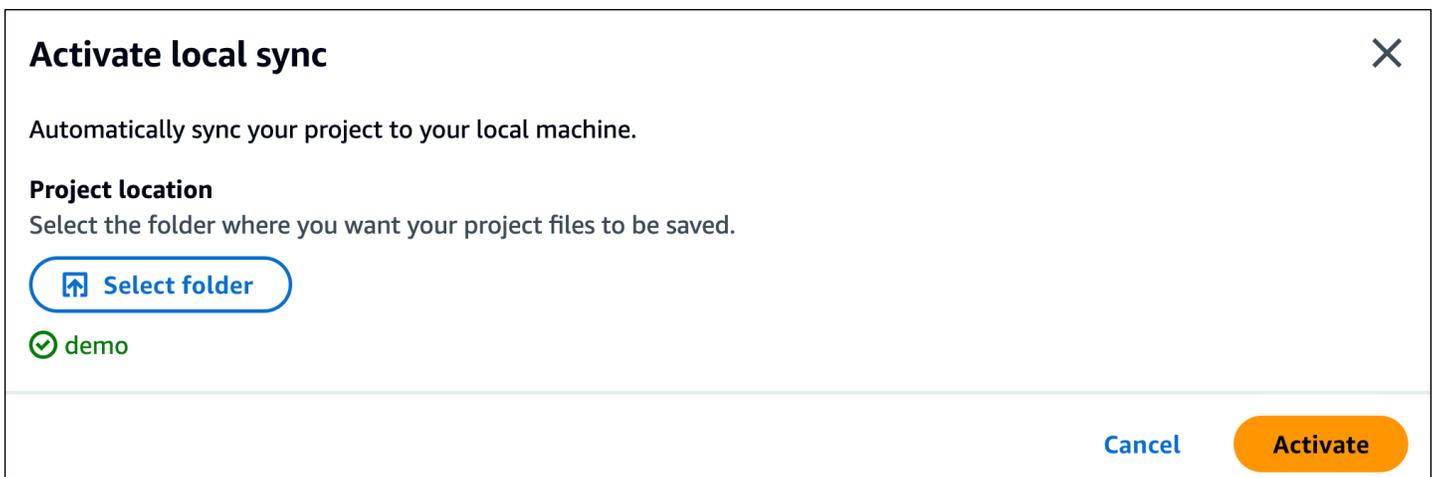
Cualquier cambio realizado en el archivo de definición de la máquina de estados se reflejará automáticamente en Infrastructure Composer.

## Haga referencia a un OpenAPI archivo externo de especificaciones con Infrastructure Composer

En este ejemplo, se utiliza Infrastructure Composer desde la consola para hacer referencia a un elemento externo OpenAPI archivo de especificaciones que define una API puerta de enlace REST API.

En primer lugar, cree un nuevo proyecto desde la página de inicio de Infrastructure Composer.

A continuación, active la sincronización local seleccionando Activar la sincronización local en el menú. Crea una nueva carpeta con un nombre demo, deja que el mensaje vea los archivos y selecciona Activar. Cuando se te pida, selecciona Guardar cambios.



A continuación, arrastra una tarjeta de Amazon API Gateway al lienzo. Selecciona Detalles para que aparezca el panel de propiedades del 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, 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. The right-hand '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.

En el panel de propiedades del recurso, configure lo siguiente y guárdelo.

- Seleccione la opción Usar un archivo externo para la definición de la API.
- `./api-spec.yaml` Introdúzcala como ruta relativa al archivo externo

## Use external file for api definition



## Relative path to external file

`./api-spec.yaml`

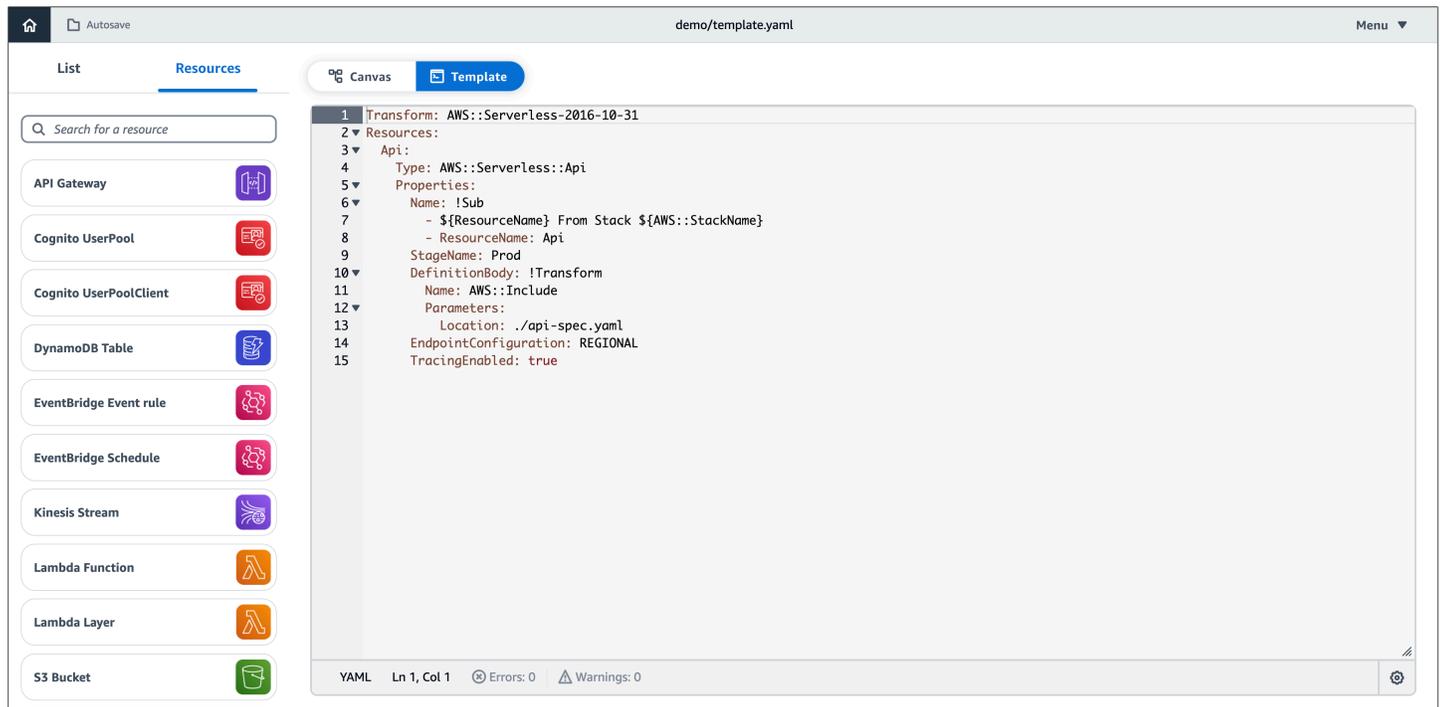
Esto crea el siguiente directorio en nuestra máquina local:

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

Ahora, puede configurar el archivo externo en nuestra máquina local. Con nuestro IDE, abra lo que se `api-spec.yaml` encuentra en la carpeta de su proyecto. Sustituya su contenido por lo siguiente:

```
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
```

En la vista de plantillas de Infrastructure Composer, puede ver que Infrastructure Composer ha actualizado automáticamente la plantilla para hacer referencia al archivo externo.

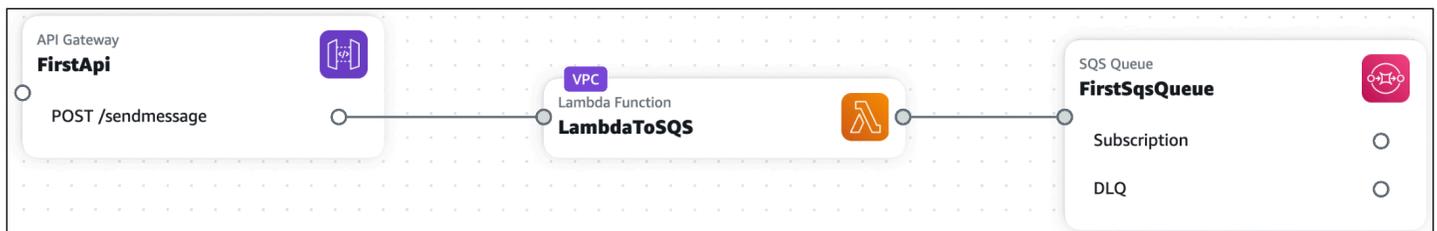


## Integre Infrastructure Composer con Amazon Virtual Private Cloud (AmazonVPC)

AWS Infrastructure Composer incluye una integración con el servicio Amazon Virtual Private Cloud (AmazonVPC). Con Infrastructure Composer, puede hacer lo siguiente:

- Identifique los recursos de su lienzo que se encuentran VPC en una VPCetiqueta visual.
- Configure AWS Lambda las funciones VPCs desde una plantilla externa.

La siguiente imagen muestra un ejemplo de una aplicación con una función Lambda configurada con un VPC



Para obtener más información sobre AmazonVPC, consulta [¿Qué es AmazonVPC?](#) en la Guía del VPC usuario de Amazon.

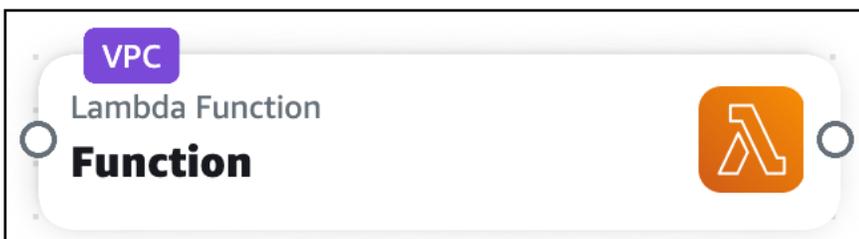
## Temas

- [Identifique los recursos de Infrastructure Composer y la información relacionada en un VPC](#)
- [Configurar funciones Lambda con funciones externas VPCs en Infrastructure Composer](#)
- [Parámetros en plantillas importadas para una externa VPC con Infrastructure Composer](#)
- [Agregar nuevos parámetros a las plantillas importadas con Infrastructure Composer](#)
- [Configure una función Lambda y una VPC definida en otra plantilla con Infrastructure Composer](#)

## Identifique los recursos de Infrastructure Composer y la información relacionada en un VPC

Para integrar Infrastructure Composer con AmazonVPC, primero debe identificar los recursos VPC y la información necesaria para completar una integración. También incluye información de configuración relacionada con los grupos de seguridad, los identificadores de subred, los tipos de parámetros, los tipos y los SSM tipos de valores estáticos.

Infrastructure Composer visualiza los recursos VPC mediante una etiqueta. VPC Esta etiqueta se aplica a las cartas del lienzo. A continuación se muestra un ejemplo de una función Lambda con una VPC etiqueta:



Las etiquetas se aplican a las cartas del lienzo cuando se hace lo siguiente:

- Configure una función Lambda con una de Infrastructure VPC Composer.
- Importe una plantilla que contenga recursos configurados con unVPC.

## Identificadores de subred y grupo de seguridad

Una función Lambda se puede configurar con varios grupos de seguridad y subredes. Para configurar un grupo de seguridad o una subred para una función Lambda, proporcione un valor y un tipo.

- Valor: identificador del grupo o la subred de seguridad. Los valores aceptados variarán según el tipo.
- Tipo: se permiten los siguientes tipos de valores:
  - Nombre del parámetro
  - AWS Systems Manager (SSM) Almacén de parámetros
  - Valor estático

## Tipo de parámetro

La `Parameters` sección de una AWS CloudFormation plantilla se puede utilizar para almacenar la información de los recursos en varias plantillas. Para obtener más información sobre los parámetros, consulte [Parámetros](#) en la Guía del AWS CloudFormation usuario.

Para el tipo de parámetro, puede proporcionar un nombre de parámetro. En el siguiente ejemplo, proporcionamos un valor de nombre de `PrivateSubnet1` parámetro:

**Subnet IDs**

List of VPC subnet identifiers

Value	Type
<input style="width: 90%; border: none;" type="text" value="PrivateSubnet1"/> <span style="float: right; color: blue; font-size: 1.2em;">✕</span>	Parameter ▼

Al proporcionar un nombre de parámetro, Infrastructure Composer lo define en la `Parameters` sección de la plantilla. A continuación, Infrastructure Composer hace referencia al parámetro en el recurso de la función Lambda. A continuación, se muestra un ejemplo:

```

...
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

El almacén de SSM parámetros proporciona un almacenamiento jerárquico y seguro para la gestión de los datos de configuración y los secretos. Para obtener más información, consulte [Almacén de parámetros de AWS Systems Manager](#) en la Guía del usuario de AWS Systems Manager .

Para el SSMtipo, puede proporcionar los siguientes valores:

- Referencia dinámica a un valor del almacén de SSM parámetros.
- ID lógico de un `AWS::SSM::Parameter` recurso definido en la plantilla.

### Referencia dinámica

Puede hacer referencia a un valor del almacén de SSM parámetros mediante una referencia dinámica en el siguiente formato: `{{resolve:ssm:reference-key}}`. Para obtener más información, consulte [SSMlos parámetros](#) en la Guía AWS CloudFormation del usuario.

Infrastructure Composer crea el código de infraestructura para configurar la función Lambda con el valor del almacén de SSM parámetros. A continuación, se muestra un ejemplo:

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

### ID lógico

Puede hacer referencia a un `AWS::SSM::Parameter` recurso de la misma plantilla mediante un ID lógico.

A continuación, se muestra un ejemplo de un `AWS::SSM::Parameter` recurso cuyo nombre `PrivateSubnet1Parameter` almacena el ID de subred para `PrivateSubnet1`:

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

A continuación, se muestra un ejemplo de este valor de recurso proporcionado por el identificador lógico de la función Lambda:

### Subnet IDs

List of VPC subnet identifiers

Value	Type
<input style="width: 90%; border: none;" type="text" value="PrivateSubnet1Parameter"/> <span style="float: right; border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px; color: blue;">×</span>	<input style="width: 90%; border: none;" type="text" value="SSM"/> <span style="float: right; border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px; color: blue;">▼</span>

Infrastructure Composer crea el código de infraestructura para configurar la función Lambda con el SSM parámetro:

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

## Tipo de valor estático

Cuando se implementa un grupo de seguridad o una subred en AWS CloudFormation, se crea un valor de ID. Puede proporcionar este ID como un valor estático.

Para el tipo de valor estático, los siguientes son valores válidos:

- Para los grupos de seguridad, proporcione el `GroupId`. Para obtener más información, consulte [Valores devueltos](#) en la Guía del AWS CloudFormation usuario. A continuación se muestra un ejemplo: `sg-0b61d5c742dc2c773`.
- Para las subredes, proporcione la `SubnetId`. Para obtener más información, consulte [Valores devueltos](#) en la Guía del AWS CloudFormation usuario. A continuación se muestra un ejemplo: `subnet-01234567890abcdef`.

Infrastructure Composer crea el código de infraestructura para configurar la función Lambda con el valor estático. A continuación, se muestra un ejemplo:

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

## Uso de varios tipos

Para los grupos de seguridad y las subredes, puede usar varios tipos juntos. El siguiente es un ejemplo que configura tres grupos de seguridad para una función Lambda proporcionando valores de diferentes tipos:

### Security group IDs

List of VPC security group identifiers

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

Infrastructure Composer hace referencia a los tres valores de la SecurityGroupIds propiedad:

```

...
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
```

## Configurar funciones Lambda con funciones externas VPCs en Infrastructure Composer

Para empezar a configurar una función de Lambda con una VPC que esté definida en otra plantilla, utilice la tarjeta de componentes mejorada de la función Lambda. Esta tarjeta representa una función Lambda que utiliza el tipo de `AWS::Serverless::Function` recurso AWS Serverless Application Model (AWS SAM).

Para configurar una función Lambda con a VPC desde una plantilla externa

1. En el panel de propiedades del recurso de la función Lambda, amplíe la sección desplegable de VPCconfiguración (avanzada).
2. Seleccione Asignar a externo. VPC
3. Proporcione valores para los grupos de seguridad y las subredes que desee configurar para la función Lambda. Para obtener más información, consulte [Identificadores de subred y grupo de seguridad](#).
4. Guarde los cambios.

## Parámetros en plantillas importadas para una externa VPC con Infrastructure Composer

Al importar una plantilla existente con los parámetros definidos para los grupos de seguridad y las subredes de una plantilla externaVPC, Infrastructure Composer proporciona una lista desplegable para seleccionar los parámetros.

A continuación, se muestra un ejemplo de la Parameters sección de una plantilla importada:

```
...
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
...

```

Al configurar una función Lambda externa VPC para una nueva función Lambda en el lienzo, estos parámetros estarán disponibles en una lista desplegable. A continuación, se muestra un ejemplo:

### Subnet IDs

List of VPC subnet identifiers

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

## Limitaciones a la hora de importar tipos de parámetros de lista

Normalmente, puede especificar varios identificadores de subred y grupos de seguridad para cada función de Lambda. Si la plantilla existente contiene tipos de parámetros de lista, como `List<AWS::EC2::SecurityGroup::Id>` o `List<AWS::EC2::Subnet::Id>`, solo puede especificar un identificador.

Para obtener más información sobre los tipos de listas de parámetros, consulte los [tipos AWS de parámetros específicos admitidos](#) en la Guía del AWS CloudFormation usuario.

A continuación se muestra un ejemplo de una plantilla que se define `VPCSecurityGroups` como un tipo de parámetro de lista:

```

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

```

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

```
...
```

En Infrastructure Composer, si selecciona el `VPCSecurityGroups` valor como identificador de grupo de seguridad para una función de Lambda, verá el siguiente mensaje:

### 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>	

Esta limitación se debe a que `SecurityGroupIds` tanto `SubnetIds` las propiedades como las propiedades de un `AWS::Lambda::Function VpcConfig` objeto solo aceptan una lista de valores de cadena. Como un único tipo de parámetro de lista contiene una lista de cadenas, puede ser el único objeto que se proporcione cuando se especifique.

Para los tipos de parámetros de lista, a continuación se muestra un ejemplo de cómo se definen en la plantilla cuando se configuran con una función 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
```

## Agregar nuevos parámetros a las plantillas importadas con Infrastructure Composer

Al importar una plantilla existente con los parámetros definidos, también puede crear parámetros nuevos. En lugar de seleccionar un parámetro existente de la lista desplegable, proporcione un tipo y un valor nuevos. A continuación, se muestra un ejemplo en el que se crea un nuevo parámetro denominado `MySecurityGroup`:

### Security group IDs

List of VPC security group identifiers

Value	Type
<input style="width: 100%;" type="text" value="MySecurityGroup"/>	<input style="width: 100%;" type="text" value="Parameter"/>
<div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">Use: "MySecurityGroup"</div>	
<div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">VPCSecurityGroups</div>	

Para todos los valores nuevos que proporcione en el panel de propiedades del recurso de la función Lambda, Infrastructure Composer los define en una lista en las `SubnetIds` propiedades `SecurityGroupIds` o de una función Lambda. A continuación, se muestra un ejemplo:

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

Si desea hacer referencia al ID lógico de un tipo de parámetro de lista desde una plantilla externa, le recomendamos que utilice la vista de plantillas y modifique directamente la plantilla. El identificador lógico de un tipo de parámetro de lista debe proporcionarse siempre como un valor único y como único valor.

```

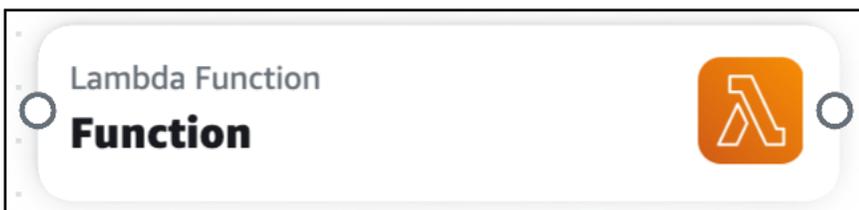
...
Parameters:
  VPCSecurityGroups:
    Description: Security group IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::SecurityGroup::Id>
  VPCSubnets:
    Description: Subnet IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::Subnet::Id>
Resources:
  ...
  MyFunction: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

```

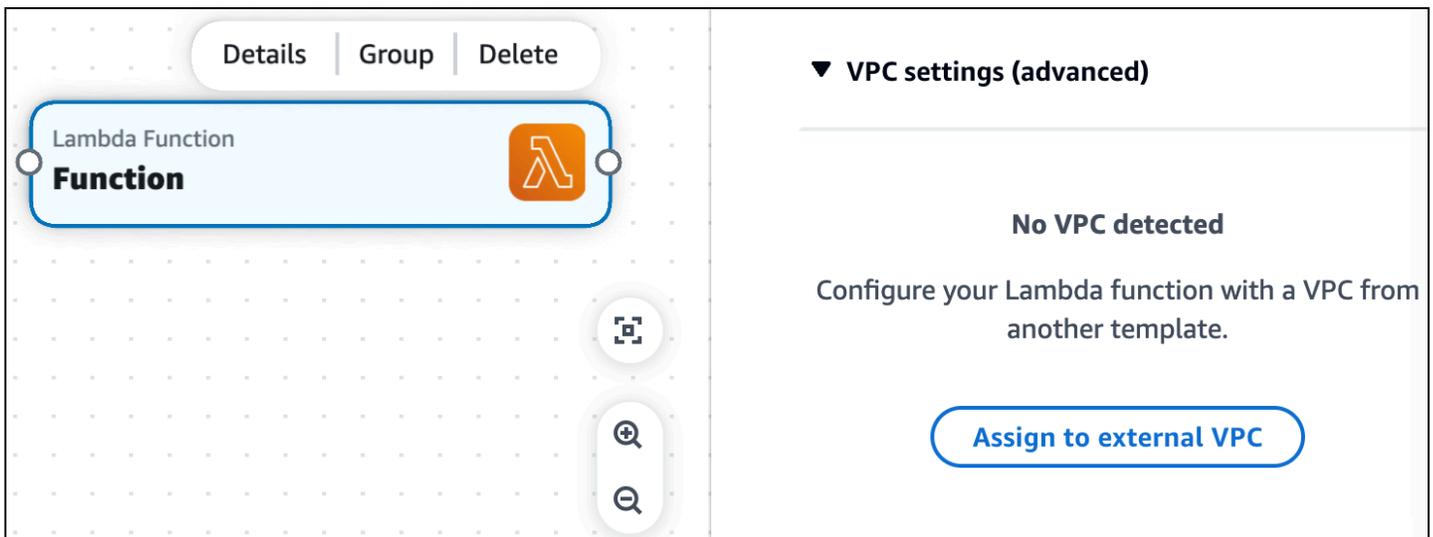
## Configure una función Lambda y una VPC definida en otra plantilla con Infrastructure Composer

En este ejemplo, configuramos una función Lambda en Infrastructure Composer con una VPC definida en otra plantilla.

Empezamos arrastrando al lienzo una tarjeta de componentes mejorada con la función Lambda.

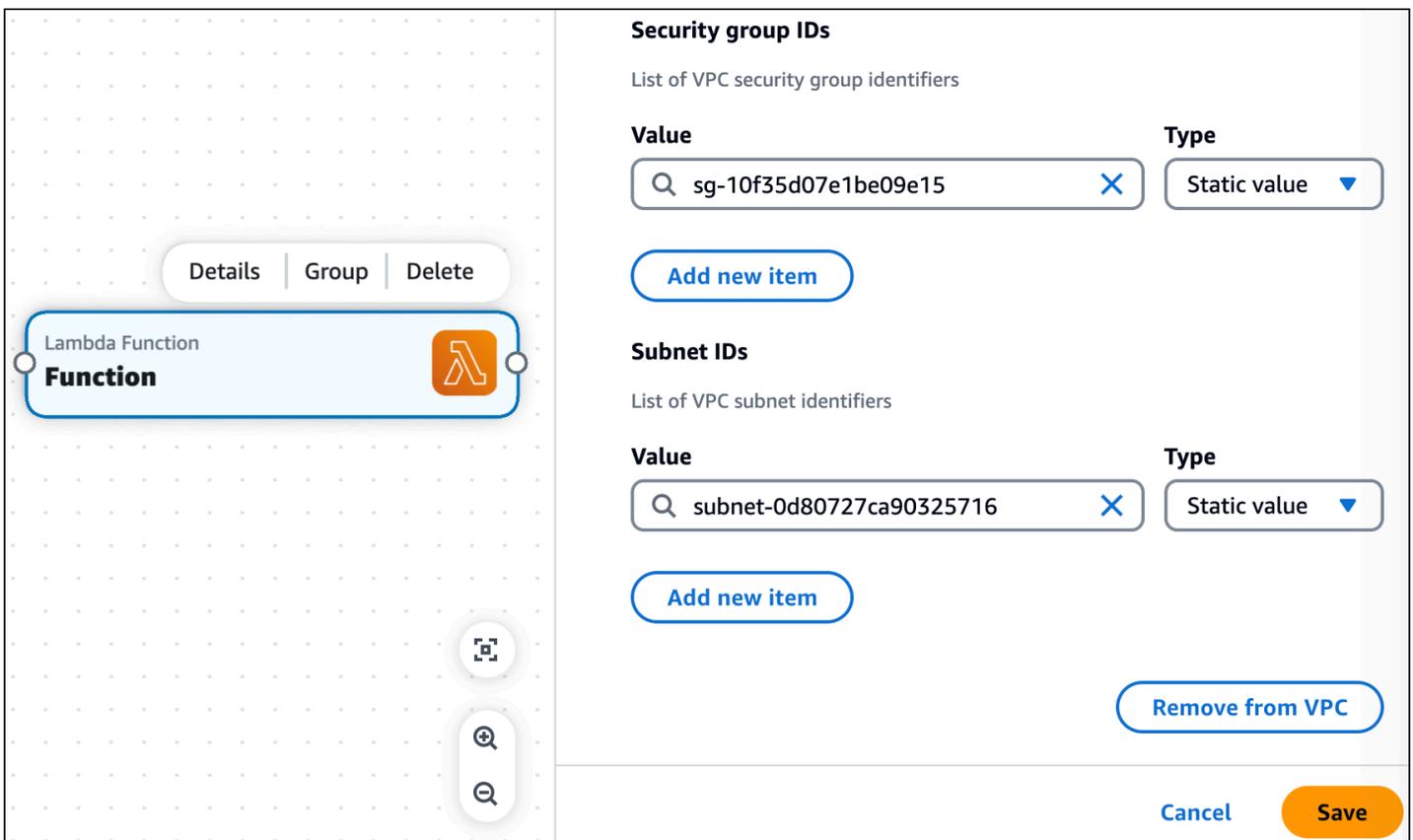


A continuación, abrimos el panel de propiedades de los recursos de la tarjeta y ampliamos la VPCsección desplegable de ajustes (avanzados).

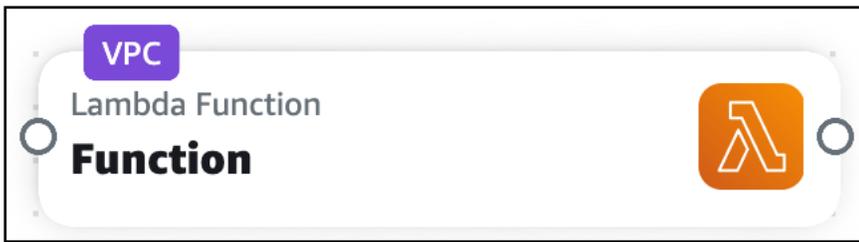


A continuación, seleccionamos Asignar a externo VPC para empezar a configurar a VPC partir de una plantilla externa.

En este ejemplo, hacemos referencia a un ID de grupo de seguridad y a un ID de subred. Estos valores se crean cuando se implementa la plantilla que VPC define el. Elegimos el tipo de valor estático e ingresamos el valor de nuestro IDs. Seleccionamos Guardar cuando terminamos.



Ahora que nuestra función Lambda está configurada con nuestra VPC, la VPC etiqueta se muestra en nuestra tarjeta.



Infrastructure Composer ha creado el código de infraestructura para configurar nuestra función Lambda con el grupo de seguridad y la subred del 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 su aplicación sin servidor Infrastructure Composer en la nube AWS

Se utiliza AWS Infrastructure Composer para diseñar aplicaciones sin servidor listas para su implementación. Para realizar la implementación, utilice cualquier servicio compatible. AWS CloudFormation Recomendamos usar el [AWS Serverless Application Model \(AWS SAM\)](#).

AWS SAM es un marco de código abierto que proporciona herramientas de desarrollo para crear y ejecutar aplicaciones sin servidor en ellas. AWS Con AWS SAM su sintaxis abreviada, los desarrolladores declaran los AWS CloudFormation recursos y los recursos especializados sin servidor que se transforman en infraestructura durante la implementación.

## Conceptos importantes AWS SAM

Antes de usarlo AWS SAM, es importante que te familiarices con algunos de sus conceptos fundamentales.

- [Cómo AWS SAM funciona](#): este tema, que se encuentra en la Guía para AWS Serverless Application Model desarrolladores, proporciona información importante sobre los componentes principales que se utilizan para crear una aplicación sin servidor: los AWS SAM CLI, el AWS SAM proyecto y la AWS SAM plantilla.
- [Cómo usar AWS Serverless Application Model \(AWS SAM\)](#): este tema, que se encuentra en la Guía para AWS Serverless Application Model desarrolladores, proporciona una descripción general de alto nivel de los pasos que debe completar AWS SAM para implementar su aplicación AWS en la nube.

Al diseñar la aplicación en Infrastructure Composer, puede usar el `sam sync` comando para tener AWS SAM CLI detecte automáticamente los cambios locales e impleméntelos en AWS CloudFormation. Para obtener más información, consulta [Cómo usar sam sync](#) en la Guía para AWS Serverless Application Model desarrolladores.

## Pasos a seguir a continuación

Consulte para [Prepárese para su implementación con el AWS SAM CLI e Infrastructure Composer](#) preparar la implementación de la aplicación.

# Prepárese para su implementación con el AWS SAM CLI e Infrastructure Composer

Para implementar su aplicación AWS SAM, primero debe instalar y acceder a AWS CLI y el AWS SAM CLI. En los temas de esta sección se proporcionan detalles sobre cómo hacerlo.

## Instale el AWS CLI

Recomendamos instalar y configurar el AWS CLI antes de instalar el AWS SAM CLI. Para obtener instrucciones, consulte [Instalar o actualizar a la última versión de AWS CLI en la Guía del AWS Command Line Interface usuario](#).

### Note

Tras instalar el AWS CLI, debe configurar AWS las credenciales. Para obtener más información, consulte [Configuración rápida](#) en la Guía del AWS Command Line Interface usuario.

## Instale el AWS SAM CLI

Para instalar el AWS SAM CLI, consulte [Instalación del AWS SAM CLI](#) en la Guía para desarrolladores de AWS Serverless Application Model .

## Acceda al AWS SAM CLI

Si usa Infrastructure Composer desde AWS Management Console, tiene las siguientes opciones para usar el AWS SAM CLI.

### Active el modo de sincronización local

Con el modo de sincronización local, la carpeta del proyecto, incluida la AWS SAM plantilla, se guarda automáticamente en el equipo local. Infrastructure Composer estructura el directorio del proyecto de forma que AWS SAM reconozca. Puede ejecutar el AWS SAM CLI desde el directorio raíz de su proyecto.

Para obtener más información sobre el modo de sincronización local, consulte [Sincronice y guarde el proyecto de forma local en la consola de Infrastructure Composer](#).

## Exporte su plantilla

Puede exportar la plantilla a su máquina local. A continuación, ejecute el AWS SAM CLI desde la carpeta principal que contiene la plantilla. También puede utilizar la `--template-file` opción con cualquier AWS SAM CLI comanda y proporciona la ruta a tu plantilla.

## Utilice Infrastructure Composer desde el AWS Toolkit for Visual Studio Code

Puede usar Infrastructure Composer del kit de Toolkit for VS Code para llevar Infrastructure Composer a su máquina local. A continuación, utilice Infrastructure Composer y AWS SAM CLI de VS Code.

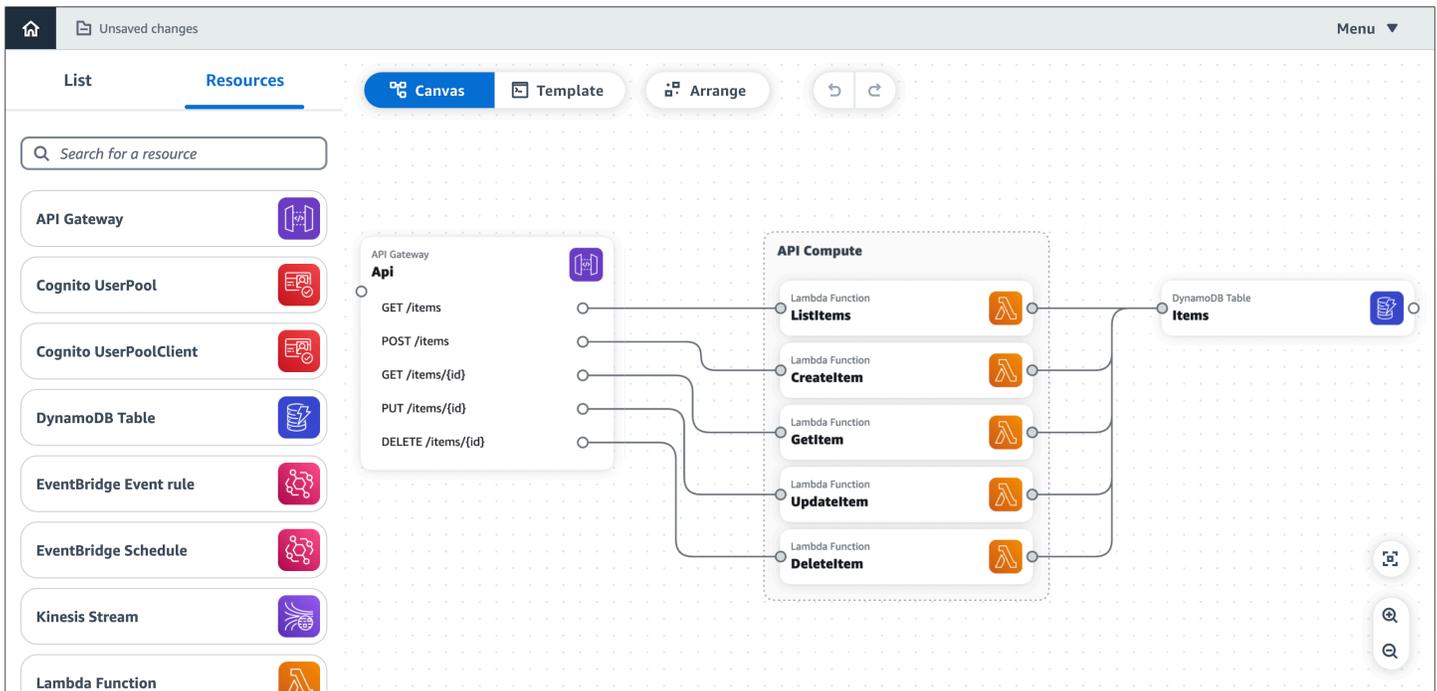
## Pasos a seguir a continuación

Para implementar su aplicación, consulte [Utilice Infrastructure Composer con AWS SAM para crear e implementar](#).

# Utilice Infrastructure Composer con AWS SAM para crear e implementar

Ahora que lo ha hecho [Prepárese para su implementación con el AWS SAM CLI e Infrastructure Composer](#), puede implementar la aplicación con AWS SAM Infrastructure Composer. En esta sección se proporciona un ejemplo en el que se detalla cómo puede hacerlo. También puede consultar la sección [Implemente su aplicación y sus recursos AWS SAM](#) en la Guía para AWS Serverless Application Model desarrolladores para obtener instrucciones sobre cómo implementar su aplicación con AWS SAM.

En este ejemplo, se muestra cómo crear e implementar la aplicación de demostración de Infrastructure Composer. La aplicación de demostración cuenta con los siguientes recursos:



### Note

- Para obtener más información sobre la aplicación de demostración, consulte [Cargue y modifique el proyecto de demostración de Infrastructure Composer](#).
- Para este ejemplo, utilizamos Infrastructure Composer con la sincronización local activada.

1. Utilice el `sam build` comando para crear la aplicación.

```
$ 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
```

El AWS SAM CLI crea el `./aws-sam` directorio en la carpeta del proyecto. Este directorio contiene artefactos de compilación para las funciones Lambda de la aplicación. Esta es una salida del directorio del proyecto:

```
.
### 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. Ahora, la aplicación está lista para su implementación. Usaremos `sam deploy --guided`. Esto prepara la aplicación para su implementación mediante una serie de instrucciones.

```
$ 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]:

```

El AWS SAM CLI muestra un resumen de lo que se desplegará:

```

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     : {}

```

El AWS SAM CLI despliega la aplicación, primero creando un conjunto de AWS CloudFormation cambios:

```

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
```

Luego, el AWS SAM CLI despliega la aplicación:

```
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            Resource creation Initiated
DeleteItemRole
CREATE_IN_PROGRESS  AWS::IAM::Role            Resource creation Initiated
ListItemsRole
CREATE_IN_PROGRESS  AWS::IAM::Role            GetItemRole
Resource creation Initiated
CREATE_IN_PROGRESS  AWS::IAM::Role            Resource creation Initiated
UpdateItemRole
CREATE_IN_PROGRESS  AWS::IAM::Role            Resource creation Initiated
CreateItemRole
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      -
-----

```

Por último, aparece un mensaje que le informa de que el despliegue se ha realizado correctamente:

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

## Utilice Infrastructure Composer con AWS SAM para eliminar una pila

En este ejemplo, se muestra cómo eliminar una AWS CloudFormation pila mediante el `aws sam delete` comando.

Introduzca el comando `aws sam delete` en el AWS SAM CLI y confirme si desea eliminar la pila y la plantilla:

```

$ 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 solución de problemas

Los temas de esta sección proporcionan orientación sobre la solución de problemas de los mensajes de error durante su uso AWS Infrastructure Composer.

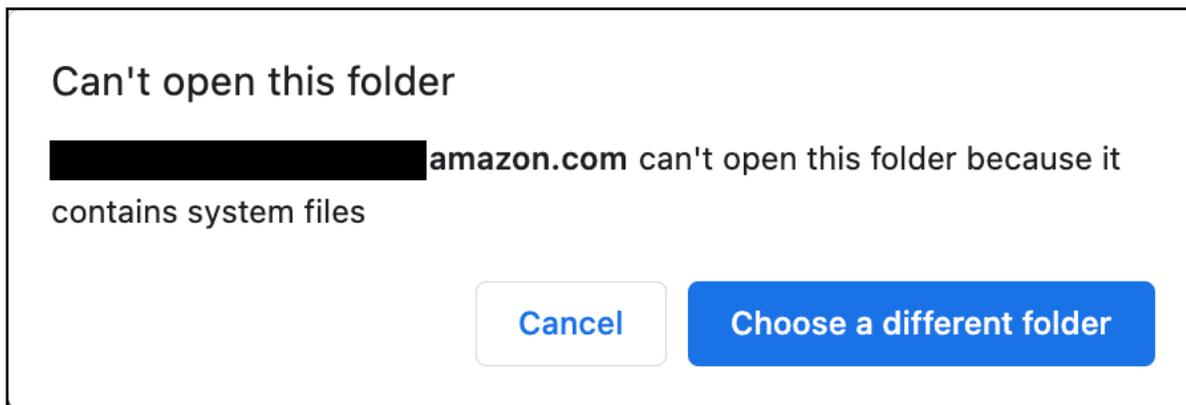
Temas

- [Mensajes de error](#)

## Mensajes de error

### «No se puede abrir esta carpeta»

Ejemplo de error:



Causa posible: Infrastructure Composer no puede acceder a un directorio confidencial mediante el modo de sincronización local.

Para obtener más información sobre este error, consulte [Data Infrastructure Composer obtiene acceso a](#).

Intente conectarse a un directorio local diferente o utilice Infrastructure Composer con la sincronización local desactivada.

### «Plantilla incompatible»

Ejemplo de error: al cargar un proyecto nuevo en Infrastructure Composer, aparece lo siguiente:

Causa posible: el proyecto contiene un archivo de referencia externa que Infrastructure Composer no admite.

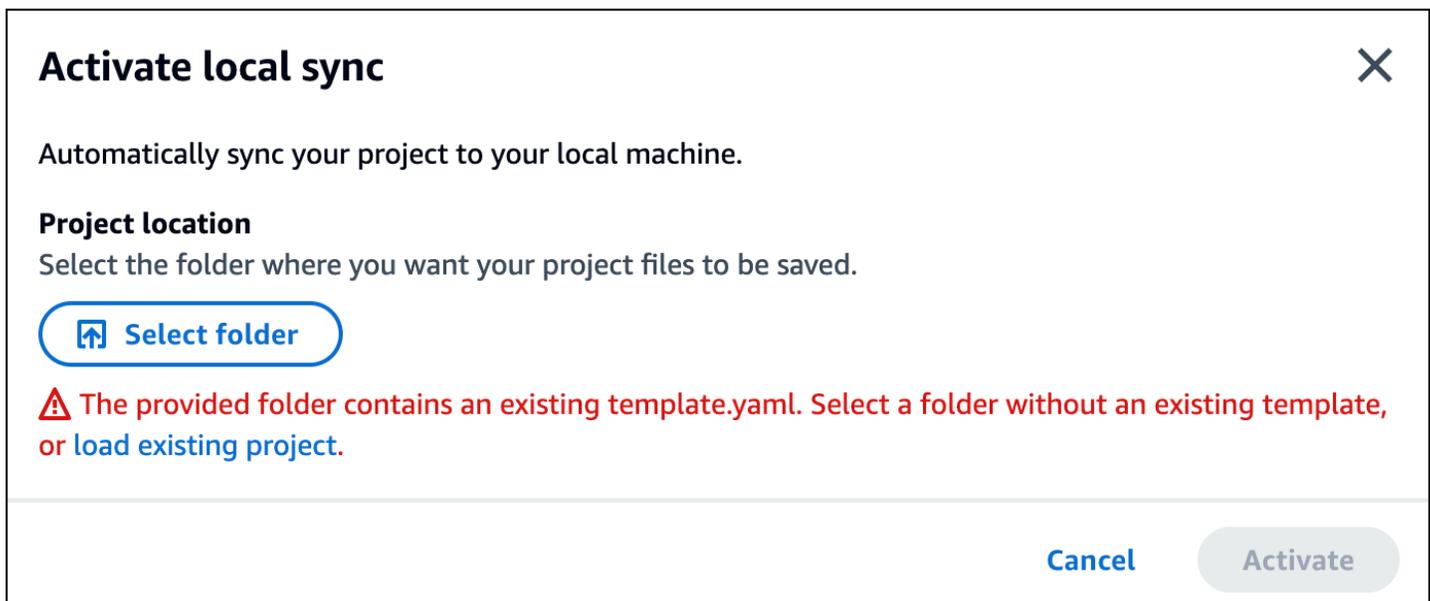
Para obtener información sobre los archivos externos compatibles en Infrastructure Composer, consulte [Haga referencia a archivos externos](#).

Causa posible: el proyecto se vincula a un archivo externo en un directorio local diferente.

Mueva el archivo al que se hace referencia externamente a un subdirectorio del directorio que haya seleccionado para usarlo con el modo de sincronización local de Infrastructure Composer.

## «La carpeta proporcionada contiene un template.yaml existente»

Al intentar activar la sincronización local, aparece el siguiente error:



Causa posible: la carpeta seleccionada ya contiene un archivo template.yaml.

Selecciona otro directorio que no contenga una plantilla de aplicación o crea uno nuevo.

«Su navegador no tiene permisos para guardar el proyecto en esa carpeta...»

Causa posible: Infrastructure Composer no puede acceder a un directorio confidencial mediante el modo de sincronización local.

Para obtener más información sobre este error, consulte [Data Infrastructure Composer obtiene acceso a](#).

Intente conectarse a un directorio local diferente o utilice Infrastructure Composer con la sincronización local desactivada.

# Seguridad en AWS Infrastructure Composer

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS Infrastructure Composer, consulte [AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos](#) .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida al utilizar Infrastructure Composer. En los temas siguientes, se muestra cómo configurar Infrastructure Composer para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger los recursos de Infrastructure Composer.

## Temas

- [Protección de datos en AWS Infrastructure Composer](#)
- [AWS Identity and Access Management para AWS Infrastructure Composer](#)
- [Validación de conformidad para AWS Infrastructure Composer](#)
- [Resiliencia en AWS Infrastructure Composer](#)

## Protección de datos en AWS Infrastructure Composer

El [modelo de](#) se aplica a protección de datos en AWS Infrastructure Composer. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en

esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte la sección [Privacidad de datos FAQ](#). Para obtener información sobre la protección de datos en Europa, consulte el [modelo de responsabilidad AWS compartida](#) y la entrada del GDPR blog sobre AWS seguridad.

Para proteger los datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- Use SSL/TLS para comunicarse con AWS los recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad del usuario con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Cómo trabajar con CloudTrail senderos](#) en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad gestionados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un FIPS terminal. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la [Norma federal de procesamiento de información \(\) FIPS 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con Infrastructure Composer u otro dispositivo Servicios de AWS mediante la consola, API AWS CLI, o. AWS SDKs Cualquier dato que ingrese en etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya información sobre las credenciales URL para validar la solicitud a ese servidor.

**Note**

Todos los datos que introduzca en Infrastructure Composer se utilizan con el único propósito de proporcionar funcionalidad a Infrastructure Composer y generar archivos y directorios de proyectos que se guardan localmente en su máquina. Infrastructure Composer no guarda, almacena ni transmite ninguno de estos datos.

## Cifrado de datos

Infrastructure Composer no cifra el contenido de los clientes, ya que los datos no se guardan, almacenan ni transmiten.

### Cifrado en reposo

Infrastructure Composer no cifra el contenido de los clientes porque los datos no se guardan, almacenan ni transmiten.

### Cifrado en tránsito

Infrastructure Composer no cifra el contenido de los clientes porque los datos no se guardan, almacenan ni transmiten.

## Administración de claves

Infrastructure Composer no admite la administración de claves, ya que el contenido del cliente no se guarda, almacena ni transmite.

## Privacidad del tráfico entre redes

Infrastructure Composer no genera tráfico con aplicaciones y clientes locales.

## AWS Identity and Access Management para AWS Infrastructure Composer

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. IAM los administradores controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos de Infrastructure Composer. IAM es un Servicio de AWS que puede utilizar sin coste adicional.

## Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración de acceso mediante políticas](#)
- [¿Cómo AWS Infrastructure Composer funciona con IAM](#)

## Público

Infrastructure Composer requiere, como mínimo, acceso de solo lectura al AWS Management Console. Cualquier usuario con esta autorización puede usar todas las funciones de Infrastructure Composer. No se admite el acceso granular a funciones específicas de Infrastructure Composer.

## Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como IAM usuario o asumiendo un IAM rol.

Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte [Cómo iniciar sesión Cuenta de AWS en su](#) Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS incluye un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte la [versión 4 de la AWS firma para ver API las solicitudes](#) en la Guía del IAM usuario.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación

multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte [Autenticación multifactorial](#) en la Guía del AWS IAM Identity Center usuario y [Autenticación AWS multifactorial IAM en](#) la Guía del IAM usuario.

## Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para obtener la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

## Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidad para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones Cuentas de AWS . Para obtener información sobre IAM Identity Center, consulte [¿Qué es IAM Identity Center?](#) en la Guía AWS IAM Identity Center del usuario.

## Usuarios y grupos de IAM

Un [IAMusuario](#) es una identidad dentro de ti Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a

largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte [Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdmins y concederle permisos para administrar IAM los recursos.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte [Casos de uso para IAM usuarios](#) en la Guía del IAM usuario.

## Roles de IAM

Un [IAMrol](#) es una identidad dentro de tu Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Para asumir temporalmente un IAM rol en el AWS Management Console, puede [cambiar de un IAM rol de usuario a uno \(consola\)](#). Puede asumir un rol llamando a una AWS API operación AWS CLI o utilizando una operación personalizadaURL. Para obtener más información sobre los métodos de uso de los roles, consulte [Métodos para asumir un rol](#) en la Guía del IAM usuario.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir los permisos para este. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información sobre las funciones de la federación, consulte [Crear una función para un proveedor de identidades externo \(federación\)](#) en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM. Para obtener información acerca de los conjuntos de permisos, consulta [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center .
- Permisos IAM de usuario temporales: un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.

- **Acceso entre cuentas:** puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta obtenga acceso a los recursos de su cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los [recursos entre cuentas IAM en](#) la Guía del IAM usuario.
- **Acceso entre servicios:** algunos Servicios de AWS utilizan funciones en otros. Servicios de AWS Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
- **Sesiones de acceso directo (FAS):** cuando utilizas un IAM usuario o un rol para realizar acciones en AWS ellas, se te considera director. Al utilizar algunos servicios, es posible que realice una acción que, a continuación, inicie otra acción en un servicio diferente. FAS utiliza los permisos de la persona principal que llama a un Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FAS las solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros recursos Servicios de AWS o para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte [Reenviar las sesiones de acceso](#).
- **Función de servicio:** una función de servicio es una [IAM función](#) que un servicio asume para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Crear un rol para delegar permisos a un Servicio de AWS](#) en la Guía del IAM usuario.
- **Función vinculada a un servicio:** una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.
- **Aplicaciones que se ejecutan en Amazon EC2:** puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y que realizan AWS CLI o AWS API solicitan. Es preferible hacerlo de este modo a almacenar claves de acceso dentro de la instancia EC2. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener

credenciales temporales. Para obtener más información, consulta [Usar un IAM rol para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon](#) en la Guía del IAM usuario.

## Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como JSON documentos. Para obtener más información sobre la estructura y el contenido de los documentos de JSON políticas, consulte [Descripción general de JSON las políticas](#) en la Guía del IAM usuario.

Los administradores pueden usar AWS JSON las políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede agregar las políticas de IAM a los roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción, independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de AWS Management Console AWS CLI, el o el AWS API.

### Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte [Definir IAM permisos personalizados con políticas administradas por el cliente](#) en la Guía del usuario. IAM

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas gestionadas. Las políticas insertadas se integran directamente en un único usuario, grupo

o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y funciones de su empresa. Cuenta de AWS Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para saber cómo elegir entre una política gestionada o una política integrada, consulte [Elegir entre políticas gestionadas y políticas integradas en la Guía del IAM](#) usuario.

## Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de buckets de Amazon S3. En los servicios que admiten políticas basadas en recursos, los gestores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puede usar políticas AWS administradas desde una política IAM basada en recursos.

## Listas de control de acceso ( ) ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3 AWS WAF y Amazon VPC son ejemplos de servicios compatibles ACLs. Para obtener más información ACLs, consulte la [descripción general de la lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

## Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAM usuario o rol). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites

de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.

- **Políticas de control de servicios (SCPs):** SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las [políticas de control de servicios](#) en la Guía del AWS Organizations usuario.
- **Políticas de control de recursos (RCPs):** RCPs son JSON políticas que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las IAM políticas asociadas a cada recurso que poseas. Este RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades Usuario raíz de la cuenta de AWS, incluidos los permisos, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte [Políticas de control de recursos \(RCPs\)](#) en la Guía del AWS Organizations usuario.
- **Políticas de sesión:** las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también puede proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

## Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del IAM usuario.

## ¿Cómo AWS Infrastructure Composer funciona con IAM

AWS Infrastructure Composer requiere, como mínimo, acceso de solo lectura al AWS Management Console. Cualquier usuario con esta autorización puede usar todas las funciones de Infrastructure Composer. No se admite el acceso granular a funciones específicas de Infrastructure Composer.

Al implementar la plantilla y los archivos del proyecto en AWS CloudFormation, necesitará disponer de los permisos necesarios. Para obtener más información, consulta [Control de acceso con AWS Identity and Access Management](#) en la Guía del usuario de AWS CloudFormation .

En la siguiente tabla se muestran IAM las funciones con las que se pueden utilizar AWS Infrastructure Composer.

Característica de IAM	Soporte para Infrastructure Composer
<a href="#">Políticas basadas en identidades</a>	No
<a href="#">Políticas basadas en recursos</a>	No
<a href="#">Acciones de políticas</a>	No
<a href="#">Recursos de políticas</a>	No
<a href="#">Claves de condición de política</a>	No
<a href="#">ACLs</a>	No
<a href="#">ABAC(etiquetas en las políticas)</a>	No
<a href="#">Credenciales temporales</a>	Sí
<a href="#">Permisos de entidades principales</a>	No
<a href="#">Roles de servicio</a>	No
<a href="#">Roles vinculados al servicio</a>	No

Para obtener una visión general de cómo funcionan Infrastructure Composer y otros AWS servicios con la mayoría de IAM las funciones, consulte [AWS los servicios con los que funcionan IAM](#) en la Guía del IAM usuario.

## Políticas basadas en la identidad para Infrastructure Composer

Admite políticas basadas en la identidad: no

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte [Definir IAM permisos personalizados con políticas administradas por el cliente](#) en la Guía del usuario. IAM

Con las políticas basadas en identidades de IAM, puede especificar las acciones permitidas o denegadas, así como los recursos y las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está asociada. Para obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la [referencia sobre los elementos de la IAM JSON política](#) en la Guía del IAM usuario.

## Políticas basadas en recursos en Infrastructure Composer

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son JSON documentos de políticas que se adjuntan a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de buckets de Amazon S3. En los servicios que admiten políticas basadas en recursos, los gestores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para hacer posible el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como entidad principal de la política basada en recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el IAM

administrador de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) el permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte el [tema Acceso a recursos entre cuentas IAM en](#) la Guía del IAM usuario.

## Acciones políticas para Infrastructure Composer

Apoya las acciones políticas: No

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de política suelen tener el mismo nombre que la AWS API operación asociada. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Infrastructure Composer, consulte las [acciones definidas por AWS Infrastructure Composer](#) en la Referencia de autorización de servicios.

## Recursos de políticas para Infrastructure Composer

Compatibilidad con recursos de políticas: no

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso mediante su [nombre de recurso de Amazon \(ARN\)](#). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (\*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de recursos de Infrastructure Composer y sus ARNs correspondientes, consulte [Recursos definidos por AWS Infrastructure Composer](#) en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar cada recurso, consulte [Acciones definidas por AWS Infrastructure Composer](#). ARN

## Claves de condición de política para Infrastructure Composer

Compatibilidad con claves de condición de políticas específicas del servicio: no

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento Condition (o bloque de Condition) permite especificar condiciones en las que entra en vigor una instrucción. El elemento Condition es opcional. Puedes crear expresiones condicionales que utilizan [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de Condition en una instrucción o varias claves en un único elemento de Condition, AWS las evalúa mediante una operación AND lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una OR operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para obtener más información, consulte [Elementos de la política de IAM: variables y etiquetas](#) en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del IAMusuario.

Para ver una lista de las claves de condición de Infrastructure Composer, consulte [las claves de condición de AWS Infrastructure Composer](#) en la Referencia de autorización de servicios. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte [Acciones definidas por AWS Infrastructure Composer](#).

## ACLs en Infrastructure Composer

Soportes ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

## ABAC con Infrastructure Composer

Soportes ABAC (etiquetas en las políticas): No

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchos AWS recursos. Etiquetar entidades y recursos es el primer paso de ABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABAC es útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información ABAC, consulte [Definir permisos con ABAC autorización](#) en la Guía del IAM usuario. Para ver un tutorial con los pasos de configuración ABAC, consulte [Usar el control de acceso basado en atributos \(ABAC\)](#) en la Guía del IAM usuario.

## Uso de credenciales temporales con Infrastructure Composer

Compatibilidad con credenciales temporales: sí

Algunos Servicios de AWS no funcionan cuando se inicia sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta la sección [Servicios de AWS Cómo trabajar con credenciales temporales IAM](#) en la Guía del IAM usuario.

Está utilizando credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de rol, consulte [Cambiar de un rol de usuario a un IAM rol \(consola\)](#) en la Guía del IAM usuario.

Puede crear credenciales temporales manualmente con la tecla AWS CLI o AWS API. A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte [Credenciales de seguridad temporales en IAM](#).

Puede usar credenciales temporales para acceder a Infrastructure Composer a través del AWS Management Console. Para ver un ejemplo, consulte [Habilitar el acceso personalizado de un agente de identidad a la AWS consola](#) en la Guía del IAM usuario.

## Permisos principales entre servicios para Infrastructure Composer

Admite sesiones de acceso directo (FAS): No

Cuando utilizas un IAM usuario o un rol para realizar acciones en AWSél, se te considera director. Al utilizar algunos servicios, es posible que realice una acción que, a continuación, inicie otra acción en un servicio diferente. FASutiliza los permisos de la persona principal que llama a un Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros recursos Servicios de AWS o para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte [Reenviar las sesiones de acceso](#).

## Funciones de servicio para Infrastructure Composer

Compatible con roles de servicio: No

Un rol de servicio es un [rol de IAM](#) que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Crear un rol para delegar permisos Servicio de AWS en un rol](#) en la Guía del IAM usuario.

#### Warning

Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de Infrastructure Composer. Edite las funciones de servicio solo cuando Infrastructure Composer proporcione instrucciones para hacerlo.

## Funciones vinculadas al servicio para Infrastructure Composer

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para obtener más información acerca de cómo crear o administrar roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#). Busque un servicio en la tabla que incluya Yes en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

## Validación de conformidad para AWS Infrastructure Composer

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte [Servicios de AWS Alcance por programa de cumplimiento](#) [Servicios de AWS](#) de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de [AWS cumplimiento > Programas AWS](#) .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#) .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- [Cumplimiento de seguridad y gobernanza](#): en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- [Referencia de servicios válidos de HIPAA](#): muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la HIPAA.
- [AWS Recursos de](#) de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- [AWS Guías de cumplimiento para clientes](#): comprenda el modelo de responsabilidad compartida desde la perspectiva del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- [Evaluación de los recursos con reglas](#) en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- [AWS Security Hub](#)— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulta la [Referencia de controles de Security Hub](#).
- [Amazon GuardDuty](#): Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- [AWS Audit Manager](#)— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

# Resiliencia en AWS Infrastructure Composer

La infraestructura AWS global se basa en distintas zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

[Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS](#)

Todos los datos que ingresa en Infrastructure Composer se utilizan con el único propósito de proporcionar funcionalidad a Infrastructure Composer y generar archivos y directorios de proyectos que se guardan localmente en su máquina. Infrastructure Composer no guarda ni almacena ninguno de estos datos.

# Historial de documentos de Infrastructure Composer

En la siguiente tabla se describen las versiones de documentación importantes de Infrastructure Composer. Para recibir notificaciones sobre las actualizaciones de esta documentación, puede suscribirse a un RSS feed.

- Última actualización de la documentación: 30 de noviembre de 2023

Cambio	Descripción	Fecha
<a href="#">Se reestructuró y actualizó contenido en toda la guía para desarrolladores</a>	Se reorganizó y reestructuró la guía para mejorar la visibilidad y la usabilidad. Se actualizaron y mejoraron los títulos. Se aportaron detalles adicionales al presentar los temas y conceptos.	1 de agosto de 2024
<a href="#">Se agregó documentación sobre el uso de Infrastructure Composer en modo CloudFormation consola y se reestructuró la Guía del desarrollador de Infrastructure Composer.</a>	AWS Infrastructure Composer ahora se puede usar en modo AWS CloudFormation consola. Para obtener más información, consulte <a href="#">Uso de Infrastructure Composer en modo CloudFormation consola</a> . Además, gran parte del contenido de la guía del usuario se ha reorganizado para crear una experiencia optimizada.	28 de marzo de 2024
<a href="#">Se agregó documentación para la integración de Infrastructure Composer con CodeWhisperer</a>	AWS Infrastructure Composer del Toolkit for VS Code proporciona una integración con Amazon CodeWhisperer. Para obtener más informaci	30 de noviembre de 2023

<a href="#">Se agregó documentación para implementar la aplicación con Infrastructure Composer desde el AWS Toolkit for Visual Studio Code</a>	ón, consulta <a href="#">Cómo usar AWS Infrastructure Composer con Amazon CodeWhisperer</a> .	30 de noviembre de 2023
<a href="#">Se agregó documentación sobre Infrastructure Composer desde la AWS Toolkit for Visual Studio Code</a>	Utilice el botón de sincronización del lienzo de Infrastructure Composer para implementar su aplicación en Nube de AWS. Para obtener más información, consulte <a href="#">Implementar la aplicación con sam sync</a> .	30 de noviembre de 2023
<a href="#">Se agregó la integración de Step Functions con Workflow Studio</a>	Ahora puede usar Infrastructure Composer de VS Code con AWS Toolkit for Visual Studio Code. Para obtener más información, consulte <a href="#">Uso AWS Infrastructure Composer desde AWS Toolkit for Visual Studio Code</a> .	27 de noviembre de 2023
<a href="#">Se agregó la integración de la consola Lambda y de Infrastructure Composer</a>	Inicie Step Functions Workflow Studio desde el lienzo de Infrastructure Composer. Para obtener más información, consulte <a href="#">Utilización AWS Infrastructure Composer con AWS Step Functions</a> .	14 de noviembre de 2023

[Se agregó Amazon VPC como servicio destacado con Infrastructure Composer](#)

Infrastructure Composer presenta una VPC etiqueta para visualizar los recursos configurados con unVPC. También puede configurar las funciones de Lambda VPCs definidas en una plantilla externa. Para obtener más información, consulte [Uso de Infrastructure Composer con Amazon VPC](#).

17 de octubre de 2023

[Se agregó Amazon RDS como servicio destacado con Infrastructure Composer](#)

Connect su aplicación Infrastru cture Composer a un clúster o instancia de Amazon RDS DB que esté definido en una plantilla externa. Para obtener más información, consulte [Uso de Infrastructure Composer con Amazon RDS](#).

17 de octubre de 2023

[Se agregó compatibilidad con Infrastructure Composer para diseñar con todos los AWS CloudFormation recursos](#)

Seleccione cualquier AWS CloudFormation recurso de la paleta de recursos para diseñar sus aplicaciones. Para obtener más informaci ón, consulte [Trabajar con cualquier AWS CloudForm ation recurso](#).

26 de septiembre de 2023

[Se agregó documentación para tarjetas en Infrastructure Composer](#)

Infrastructure Composer admite varios tipos de tarjetas que puede usar para diseñar y crear su aplicación. Para obtener más información, consulte [Diseñar con tarjetas en Infrastructure Composer](#).

20 de septiembre de 2023

[Se agregó documentación para la función de deshacer y rehacer](#)

Utilice los botones de deshacer y rehacer del lienzo de Infrastructure Composer. Para obtener más información, consulte [Deshacer y rehacer](#).

1 de agosto de 2023

[Se agregó documentación para el modo de sincronización local](#)

Usa el modo de sincronización local para sincronizar y guardar automáticamente el proyecto en tu máquina local. Para obtener más información, consulta [Modo de sincronización local](#).

1 de agosto de 2023

[Se agregó documentación para la función de exportación de lienzo](#)

Utilice la función de exportación de lienzo para exportar el lienzo de su aplicación como una imagen a su máquina local. Para obtener más información, consulte [Exportar lienzo](#).

1 de agosto de 2023

[Soporte de Infrastructure Composer para referencias a archivos externos](#)

Consulte los archivos externos para obtener los recursos compatibles en Infrastructure Composer. Para obtener más información, consulte [Trabajar con plantillas que hacen referencia a archivos externos](#).

17 de mayo de 2023

[Nueva documentación sobre la conexión de recursos](#)

Conecte los recursos para definir relaciones impulsadas por eventos entre los recursos de su aplicación. Para obtener más información, consulte [Conectar recursos entre sí mediante el lienzo visual de Infrastructure Composer](#).

7 de marzo de 2023

[Nueva función del Inspector de cambios](#)

Utilice el Inspector de cambios para ver las actualizaciones del código de la plantilla y obtener información sobre lo que Infrastructure Composer está creando para usted. Para obtener más información, consulte [Ver actualizaciones de código con el Inspector de cambios](#).

7 de marzo de 2023

[Infrastructure Composer ahora está disponible de forma general](#)

AWS Infrastructure Composer ahora está disponible para el público en general. Para obtener más información, consulte [AWS Infrastructure Composer ahora disponible para todos: Cree aplicaciones sin servidor de forma visual y rápida](#).

7 de marzo de 2023

[Ampliamos las ventajas del uso del modo conectado](#)

Utilice Infrastructure Composer en modo conectado con su local IDE para acelerar el desarrollo. Para obtener más información, consulte [Uso de Infrastructure Composer con su entorno local IDE](#).

7 de marzo de 2023

---

<a href="#">Tema actualizado sobre el uso de otros AWS servicios para implementar la aplicación</a>	Use Infrastructure Composer para diseñar aplicaciones sin servidor listas para su implementación. Úselo para implementar su AWS SAM aplicación sin servidor. Para obtener más información, consulte <a href="#">Uso de Infrastructure Composer con AWS CloudFormation y AWS SAM</a> .	3 de marzo de 2023
<a href="#">Se agregó la sección de conceptos sin servidor</a>	Obtenga información sobre los conceptos básicos de la tecnología sin servidor antes de usar Infrastructure Composer. Para obtener más información, consulte <a href="#">Conceptos de la tecnología sin servidor</a> .	2 de marzo de 2023
<a href="#">Versión pública</a>	Versión pública inicial de Infrastructure Composer.	1 de diciembre de 2022

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.