



Guía para desarrolladores

# AWS Infrastructure Composer



# AWS Infrastructure Composer: Guía para desarrolladores

Copyright © 2026 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
Siguientes pasos .....	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 .....	15
Conexiones entre tarjetas .....	15
Conexiones entre tarjetas de componentes mejorados .....	16
Conexiones hacia y desde tarjetas de recursos laC estándar .....	17
Introducción .....	18
Haga un recorrido por la consola .....	18
Siguientes pasos .....	19
Cargue y modifique .....	19
Paso 1: Abre la demostración .....	20
Paso 2: Explore el lienzo visual .....	20
Paso 3: Amplíe su arquitectura .....	22
Paso 4: Guarda tu solicitud .....	24
Siguientes pasos .....	24
Build .....	24
Propiedades de recursos .....	25
Paso 1: Crea tu proyecto .....	26
Añadir tarjetas .....	28
Paso 3: Configura tu API REST .....	28
Paso 4: Configure sus funciones .....	29
Paso 5: Conecta tus tarjetas .....	30
Paso 6: Organice el lienzo .....	31

Agregar una tabla de DynamoDB .....	32
Paso 8: Revisa tu plantilla .....	33
Paso 9: Intégrelo en sus flujos de trabajo .....	34
Sigüientes pasos .....	34
Dónde usar Infrastructure Composer .....	35
Consola Infrastructure Composer .....	35
Descripción visual .....	36
Administre su proyecto .....	39
Conéctese a su IDE local .....	42
Permitir el acceso a la página web .....	45
Sincronice y guarde localmente .....	46
Importación desde la consola Lambda .....	49
Exporte el lienzo .....	50
CloudFormation modo consola .....	52
¿Por qué usar este modo? .....	52
Acceda a este modo .....	52
Visualice una implementación .....	53
Cree una plantilla nueva .....	53
Actualice una pila existente .....	55
AWS Toolkit for Visual Studio Code .....	57
Descripción visual .....	57
Acceda desde VS Code .....	59
Sincronizar con Nube de AWS .....	60
Infrastructure Composer con Amazon Q .....	62
¿Cómo componer .....	65
Coloque las tarjetas en el lienzo .....	65
Agrupe las tarjetas .....	66
Agrupar tarjetas de componentes mejorados .....	66
Agrupar una tarjeta de componentes estándar en otra .....	67
Connect tarjetas .....	68
Conexión de tarjetas de componentes mejorados .....	68
Conexión de tarjetas estándar .....	69
Ejemplos .....	71
Desconecte las tarjetas .....	73
Tarjetas con componentes mejorados .....	73
Tarjetas de componentes estándar .....	73

Organice las tarjetas .....	75
Configure y modifique las tarjetas .....	75
Tarjetas mejoradas .....	76
Tarjetas estándar .....	90
Eliminar tarjetas .....	92
Tarjetas de componentes mejoradas .....	92
Tarjetas de componentes estándar .....	92
Vea las actualizaciones de código .....	93
Ventajas del Inspector de Cambios .....	93
Procedimiento .....	94
Más información .....	95
Haga referencia a archivos externos .....	95
Prácticas recomendadas .....	97
Cree una referencia de archivo externo .....	97
Cargue un proyecto .....	98
Cree una aplicación mediante el AWS SAMCLI .....	99
Haga referencia a una OpenAPI especificación .....	102
Integración con Amazon VPC .....	105
Identifique los recursos y la información .....	105
Configure las funciones .....	111
Parámetros de las plantillas importadas .....	111
Añadir nuevos parámetros a las plantillas importadas .....	114
Configurar una función Lambda con una VPC en otra plantilla .....	115
Implemente en la nube AWS .....	118
Conceptos importantes AWS SAM .....	118
Sigüentes pasos .....	118
Configure el AWS SAMCLI .....	119
Instala la CLI AWS .....	119
Instala la CLI AWS SAM .....	119
Acceda al AWS SAMCLI .....	119
Sigüentes pasos .....	120
Cree e implemente .....	120
Eliminar una pila .....	128
Resolución de problemas .....	130
Mensajes de error .....	130
«No se puede abrir esta carpeta» .....	130

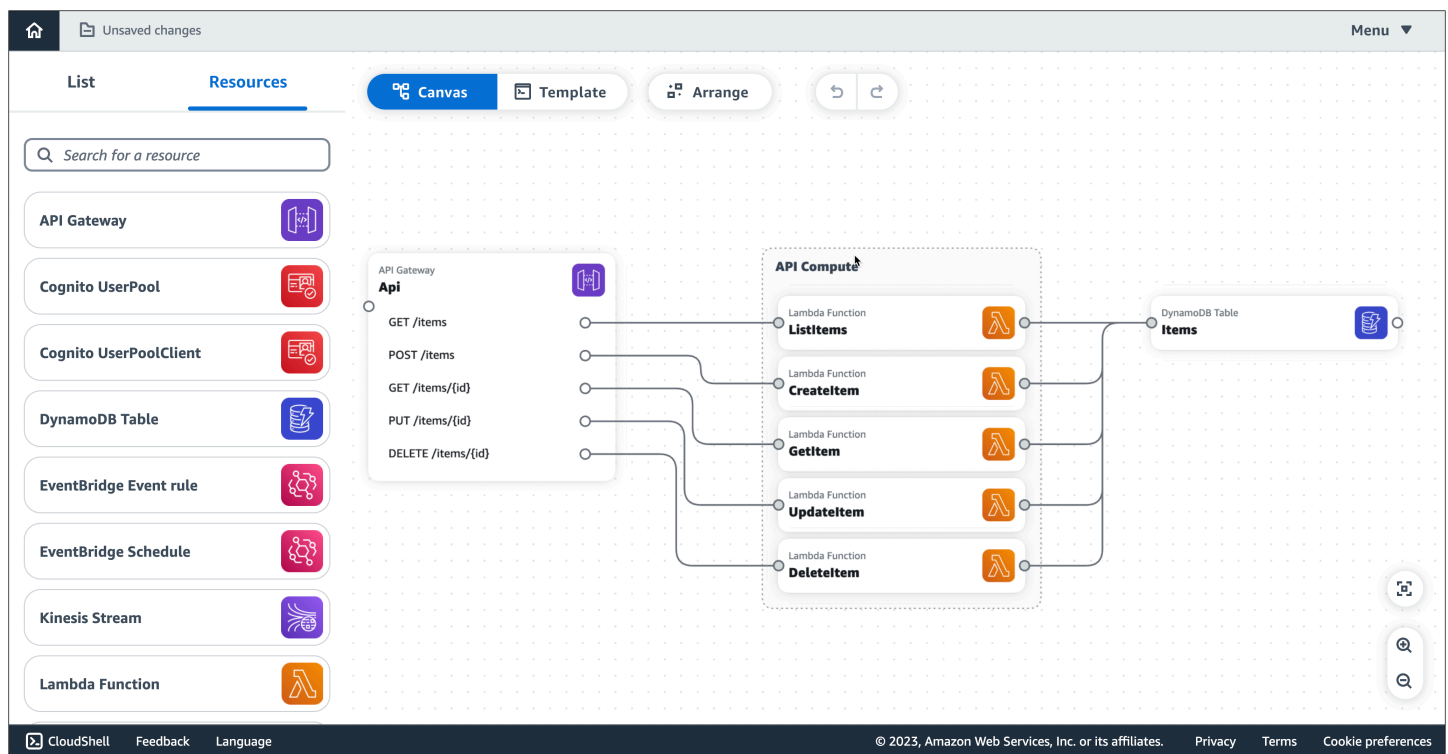
---

«Plantilla incompatible» .....	130
«La carpeta proporcionada contiene un template.yaml existente» .....	131
«Su navegador no tiene permisos para guardar el proyecto en esa carpeta...» .....	131
Seguridad .....	132
Protección de datos .....	132
Cifrado de datos .....	134
Cifrado en tránsito .....	134
Administración de claves .....	134
Privacidad del tráfico entre redes .....	134
AWS Identity and Access Management .....	134
Público .....	135
Autenticación con identidades .....	135
Administración del acceso con políticas .....	136
¿Cómo AWS Infrastructure Composer funciona con IAM .....	138
Validación de conformidad .....	144
Resiliencia .....	144
Historial de revisión .....	145
.....	cli

# ¿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 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

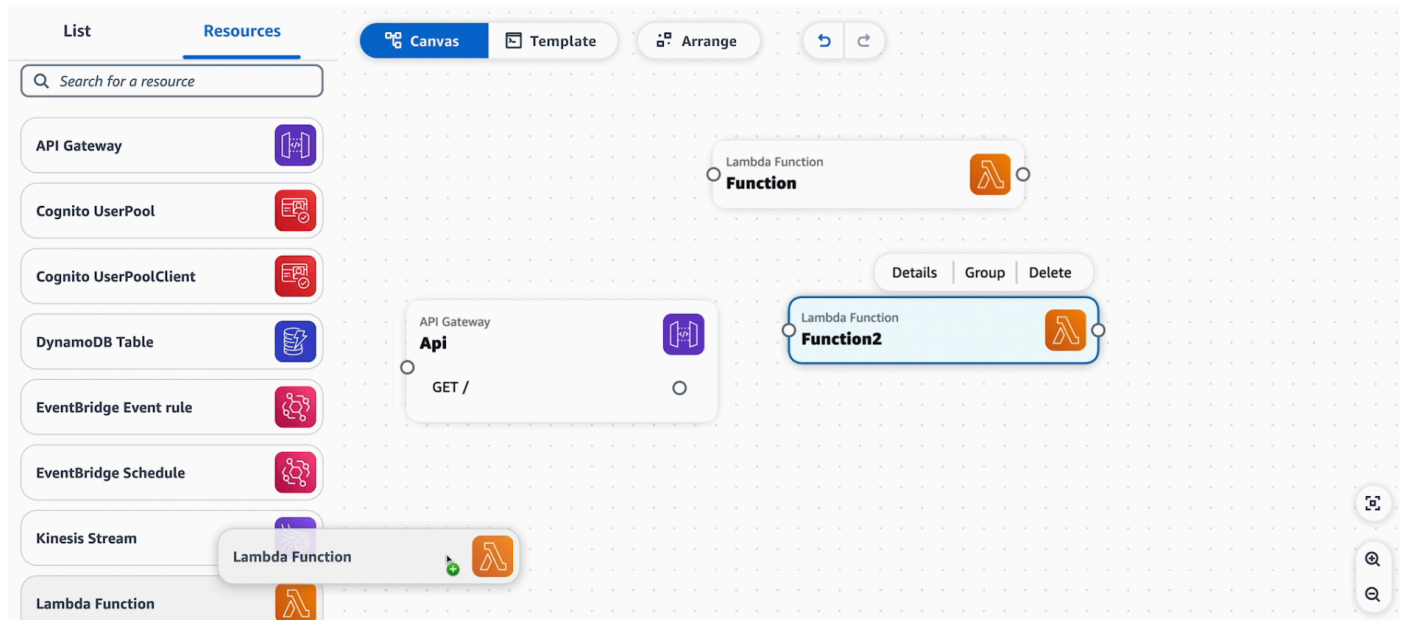
- [Cree la arquitectura de su aplicación](#)
- [Defina su infraestructura como plantillas de código \(IaC\)](#)
- [Intégrelo con sus flujos de trabajo existentes](#)
- [Formas de acceder a Infrastructure Composer](#)

- [Más información](#)
- [Sigüentes pasos](#)
- [Conceptos sin servidor para AWS Infraestructura Composer](#)

## Cree la arquitectura de su aplicación

### Construye con cartas

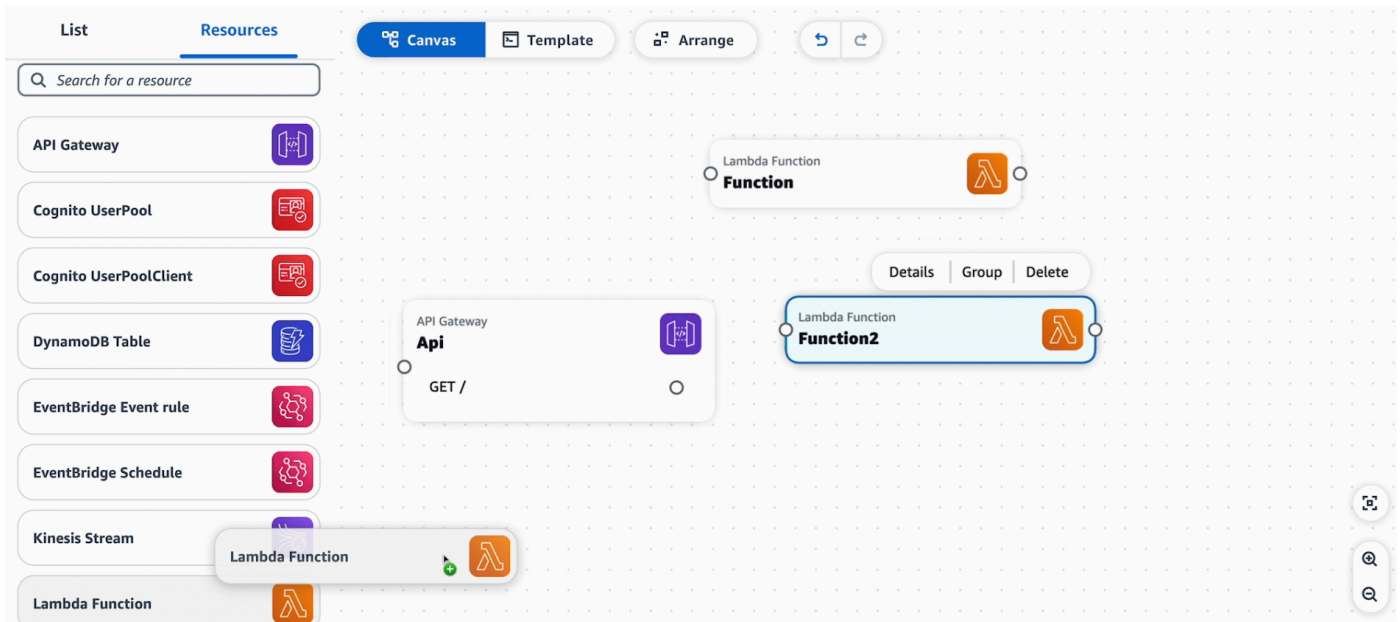
Coloque las tarjetas en el lienzo de Infrastructure Composer para visualizar y crear la arquitectura de su aplicación.



### Connect tarjetas

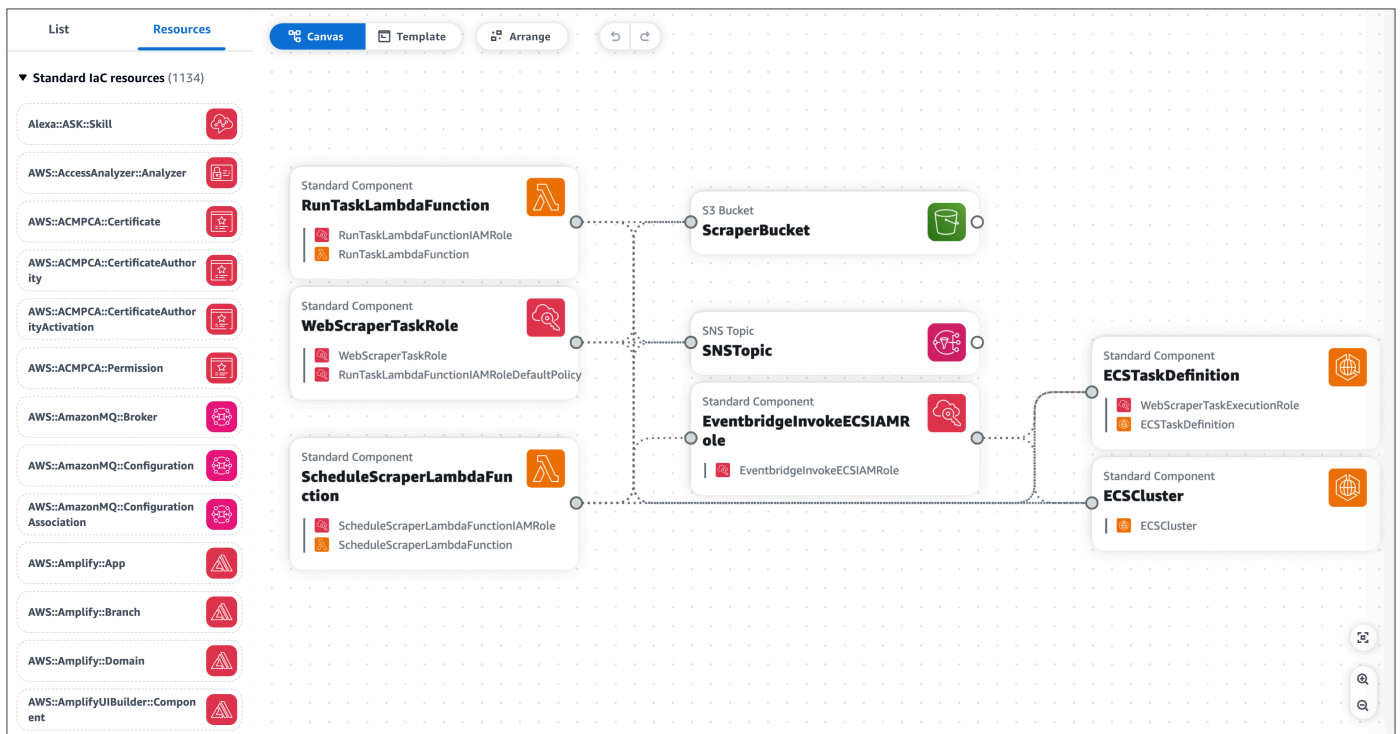
Configure la forma en que sus recursos interactúan entre sí conectándolos visualmente.

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



### Trabaja con cualquier AWS CloudFormation recurso

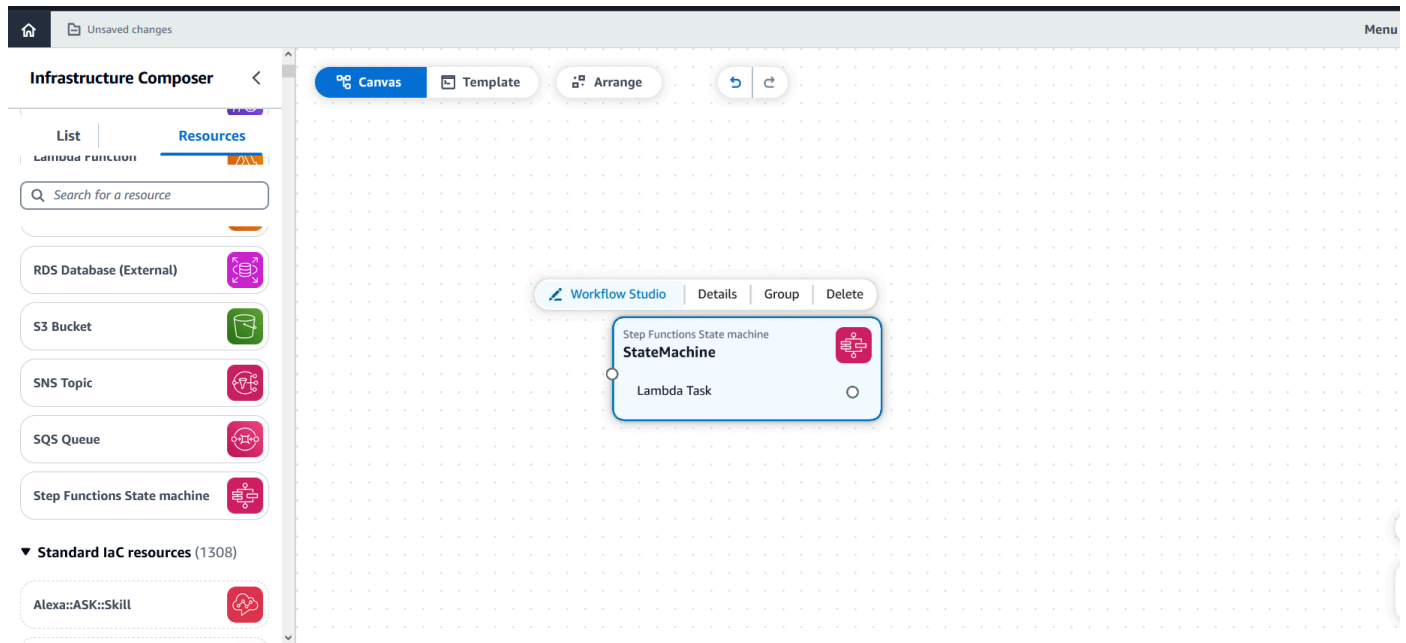
Arrastra cualquier CloudFormation recurso al lienzo para crear la arquitectura de tu aplicación. Infrastructure Composer proporciona una plantilla de IaC inicial que puede usar para especificar las propiedades del recurso. Para obtener más información, consulte [Configurar y modificar tarjetas en Infrastructure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

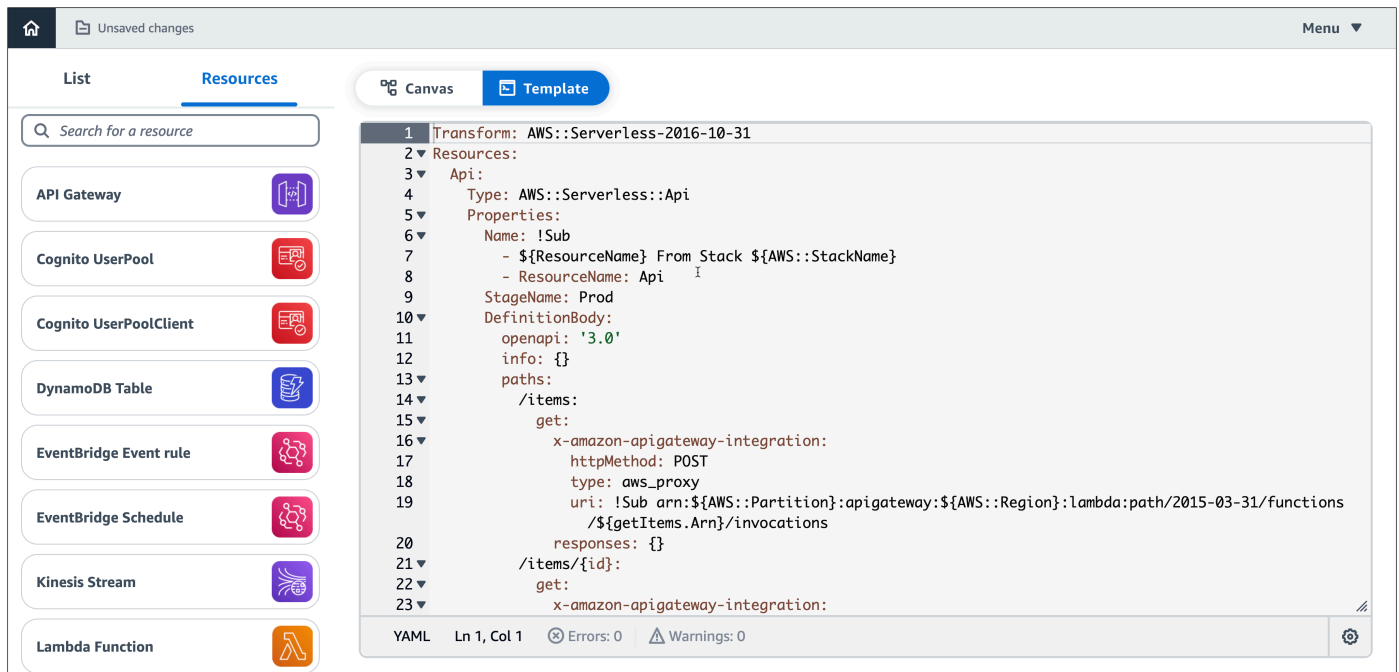
A continuación se muestra un ejemplo de la AWS Step Functions función, que proporciona una integración para lanzar Step Functions Workflow Studio directamente en el lienzo de Infrastructure Composer.



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

Infrastructure Composer crea su código de infraestructura

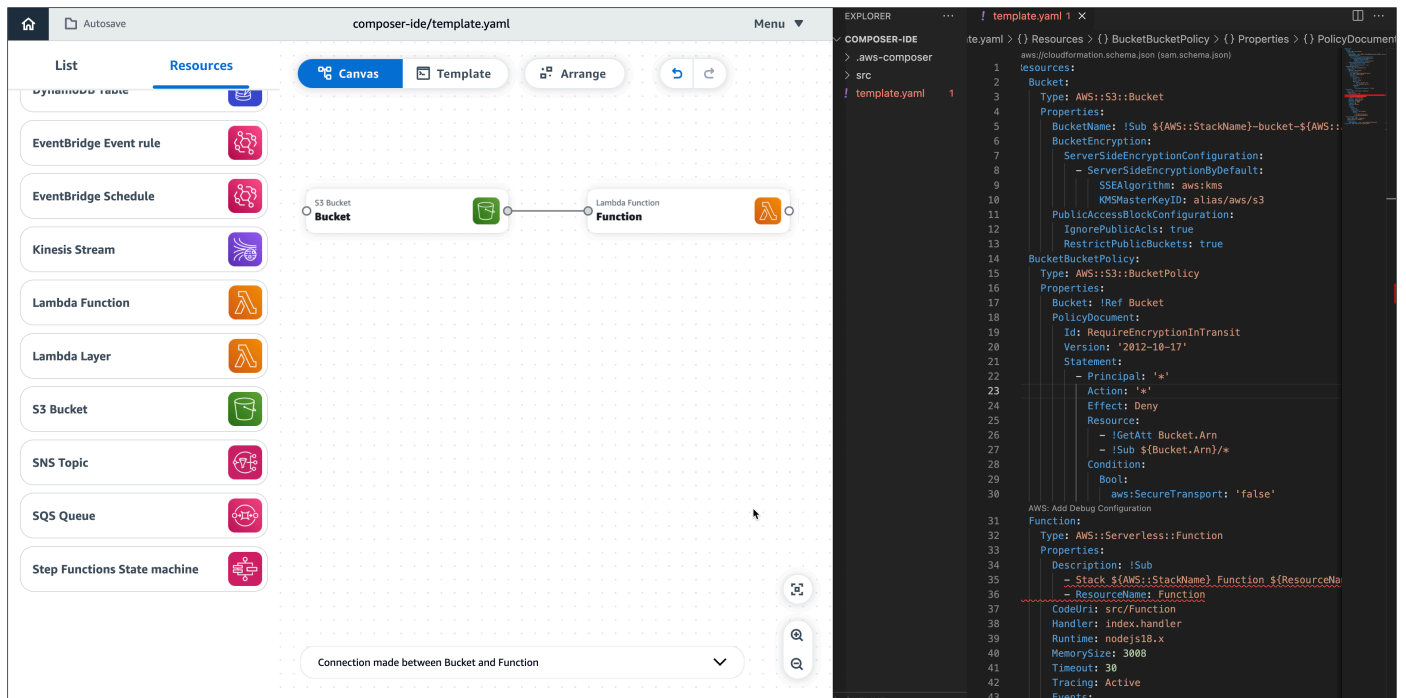
A medida que redacta, Infrastructure Composer crea automáticamente sus plantillas AWS CloudFormation and AWS Serverless Application Model (AWS SAM) siguiendo las prácticas AWS recomendadas. Puede ver y modificar las plantillas directamente desde Infrastructure Composer. Infrastructure Composer sincroniza 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

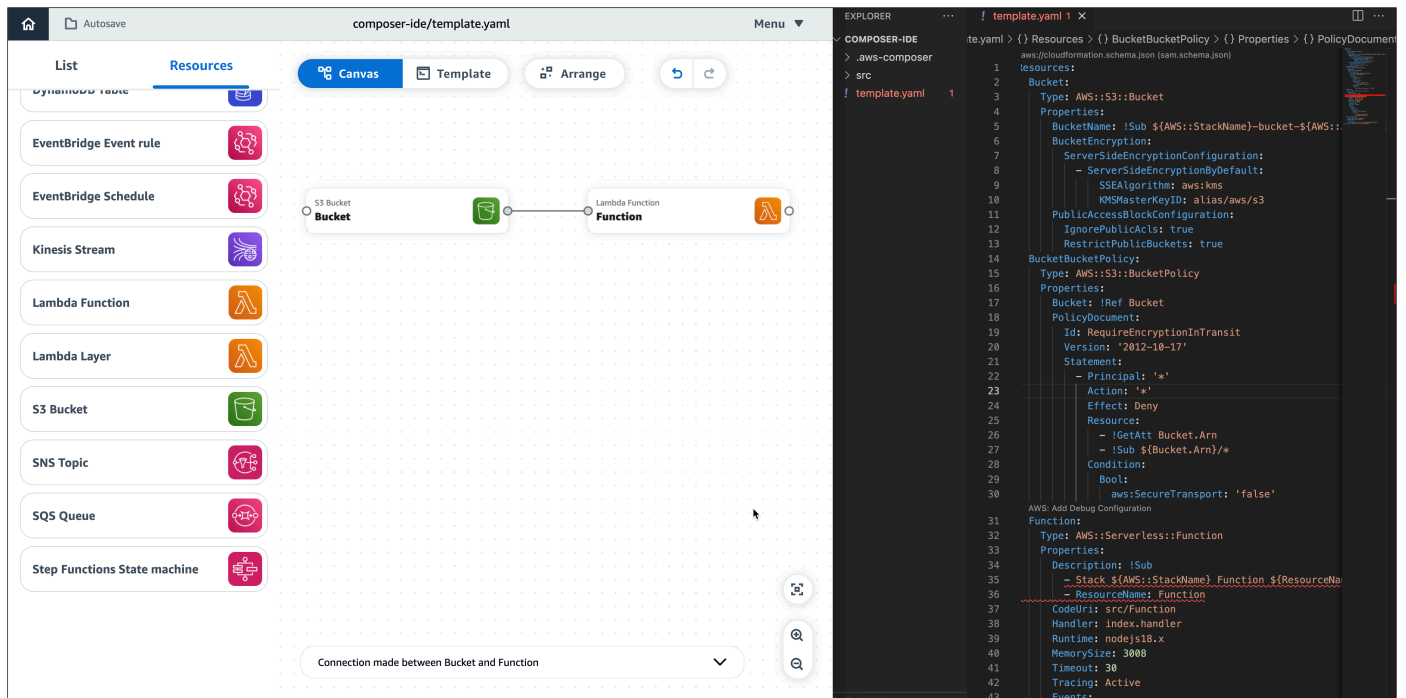
Importe AWS SAM plantillas CloudFormation y plantillas existentes para visualizarlas y comprender mejor su diseño y modificarlas. Exporte las plantillas que cree en Infrastructure Composer e intégreelas en sus flujos de trabajo existentes para su implementación.



## 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 CloudFormation

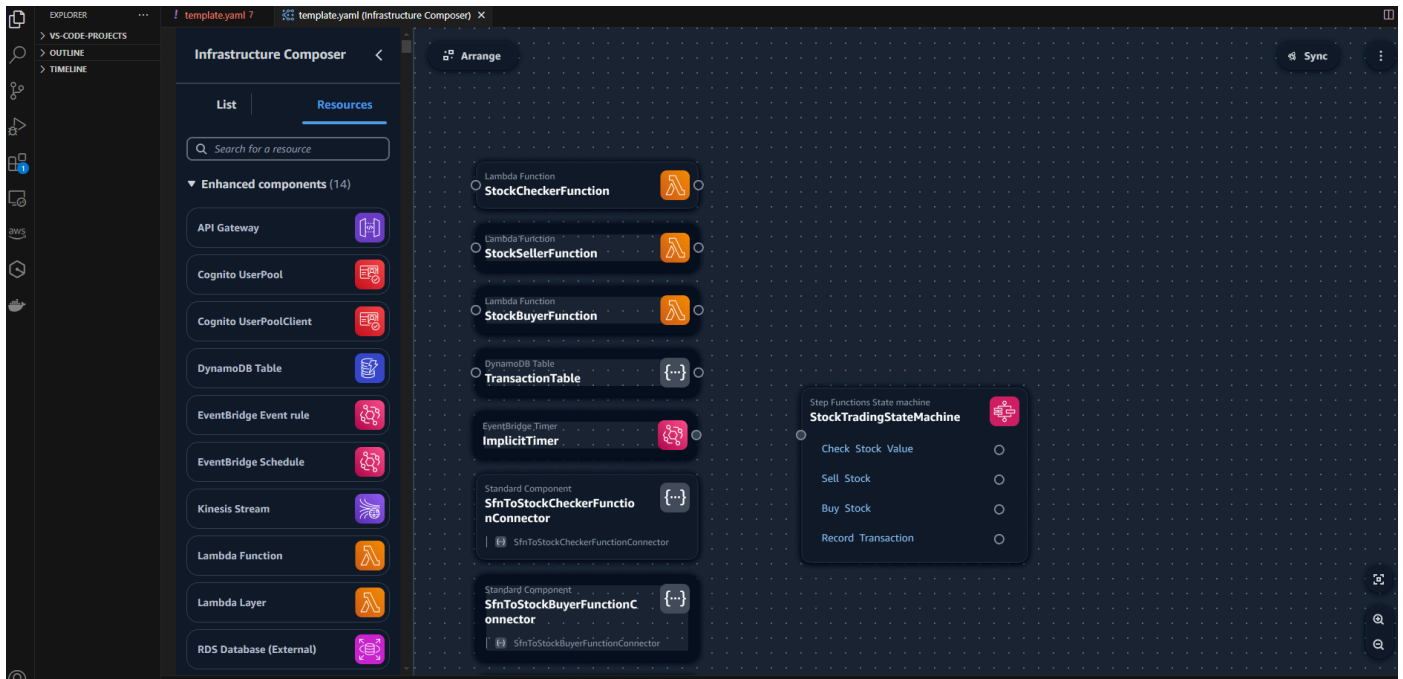
La consola Infrastructure Composer también es compatible con [el modo CloudFormation consola](#), una mejora con respecto a CloudFormation Designer, que se integra con el flujo de trabajo de la CloudFormation pila. Esta nueva herramienta es ahora la herramienta recomendada para visualizar CloudFormation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

## Siguientes pasos

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 Con el IAC, usted define en el código lo que quiere AWS aprovisionar. Para obtener más información, consulte [La infraestructura como código](#) en la introducción a un documento DevOps AWS AWS 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 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.](#)

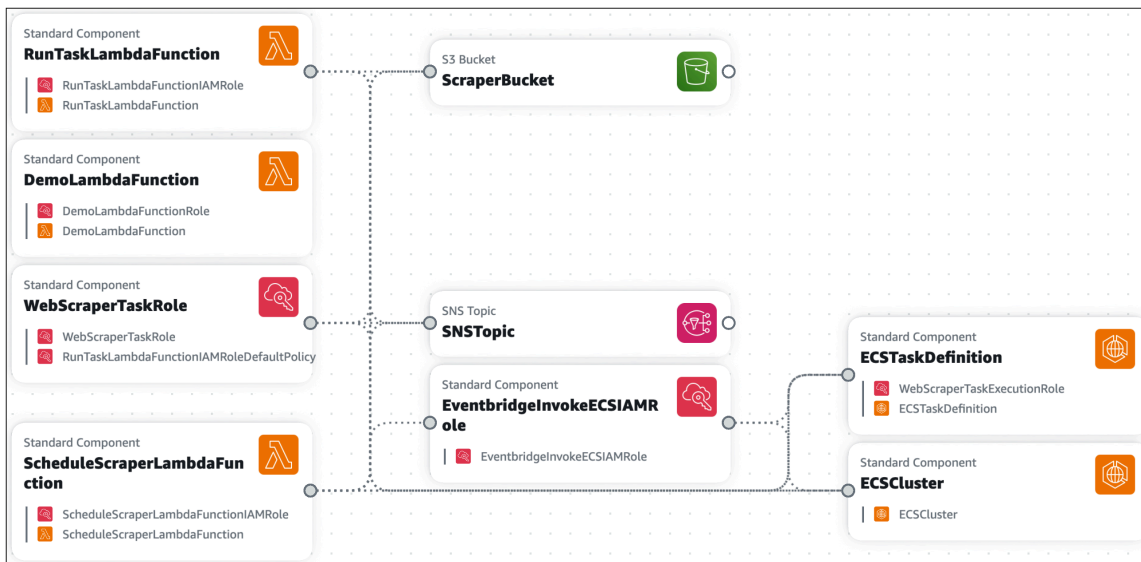
En Infrastructure Composer, las tarjetas representan CloudFormation recursos. Existen dos categorías generales de tarjetas:

- [Tarjeta con componentes mejorada](#): colección de 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 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 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::Serverless::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::Table</a>
Regla de Amazon EventBridge Event	<a href="#">AWS::Events::Rule</a>
EventBridge Calendario	<a href="#">AWS::Scheduler::Schedule</a>
Amazon Kinesis Stream	<a href="#">AWS::Kinesis::Stream</a>
AWS Lambda Función	<a href="#">AWS::Serverless::Function</a>
Capa Lambda	<a href="#">AWS::Serverless::LayerVersion</a>
Bucket de Amazon Simple Storage Service (Amazon S3)	<a href="#">AWS::S3::Bucket</a>
Tema del Amazon Simple Notification Service (Amazon SNS)	<a href="#">AWS::SNS::Topic</a>
Cola del Amazon Simple Queue Service (Amazon SQS)	<a href="#">AWS::SQS::Queue</a>
AWS Step Functions Máquina de estados	<a href="#">AWS::Serverless::StateMachine</a>

## Ejemplo

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



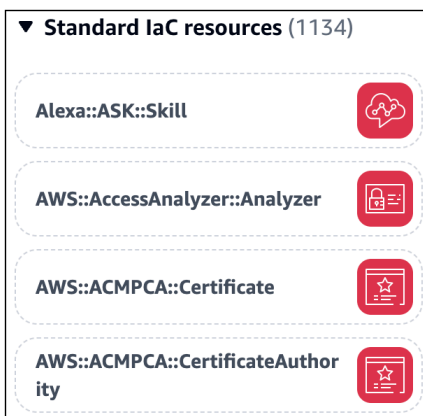
Al arrastrar una tarjeta de componentes de S3 Bucket al lienzo y ver la plantilla, verá que se han agregado CloudFormation los dos recursos siguientes a la plantilla:

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

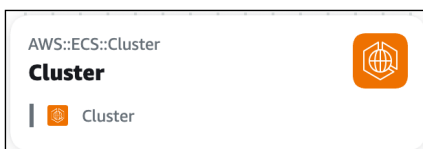
La tarjeta de componentes mejorada de S3 Bucket representa dos CloudFormation recursos que son necesarios para que un bucket de Amazon Simple Storage Service (Amazon S3) interactúe con otros servicios de la aplicación.

## 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 Recursos de Infrastructure Composer. Una tarjeta de recursos estándar (IaC) representa un único 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 CloudFormation recursos.



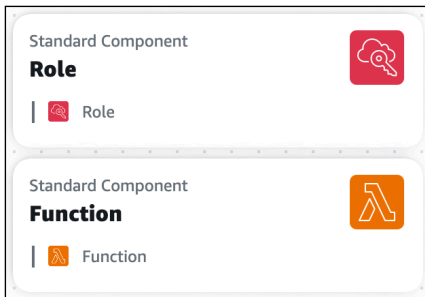
Cada tarjeta de recursos IaC estándar se puede identificar por su tipo de CloudFormation recurso. El siguiente es un ejemplo de una tarjeta de recursos IaC estándar que representa un tipo de `AWS::ECS::Cluster` CloudFormation recurso:



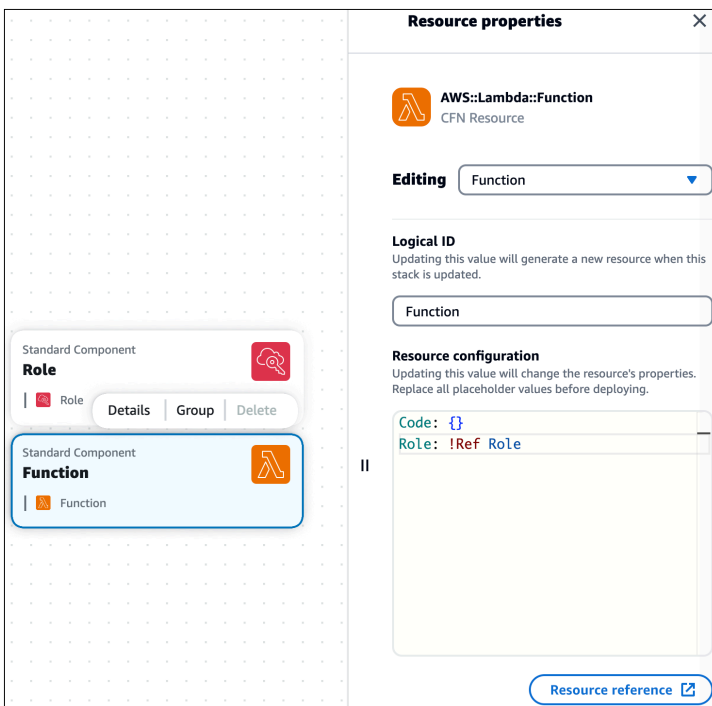
Cada tarjeta de componentes estándar visualiza los CloudFormation recursos que contiene. El siguiente es un ejemplo de una tarjeta de componentes estándar que incluye dos recursos IaC estándar:



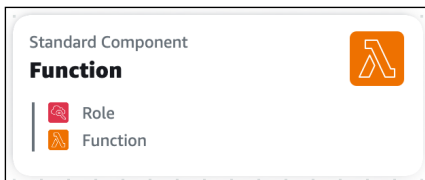
Al configurar las propiedades de las tarjetas de componentes estándar, Infrastructure Composer puede combinar tarjetas relacionadas. Por ejemplo, aquí hay dos tarjetas de componentes estándar:



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) mediante su identificador lógico:



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

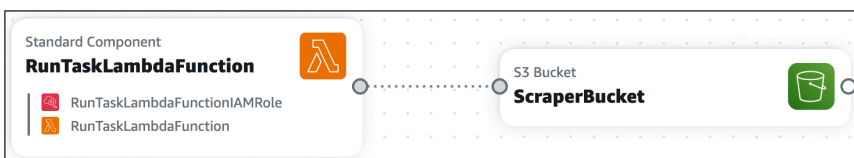
- [Conexiones entre tarjetas](#)
- [Conexiones entre tarjetas de componentes mejorados](#)
- [Conexiones hacia y desde tarjetas de recursos laC estándar](#)

## 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.

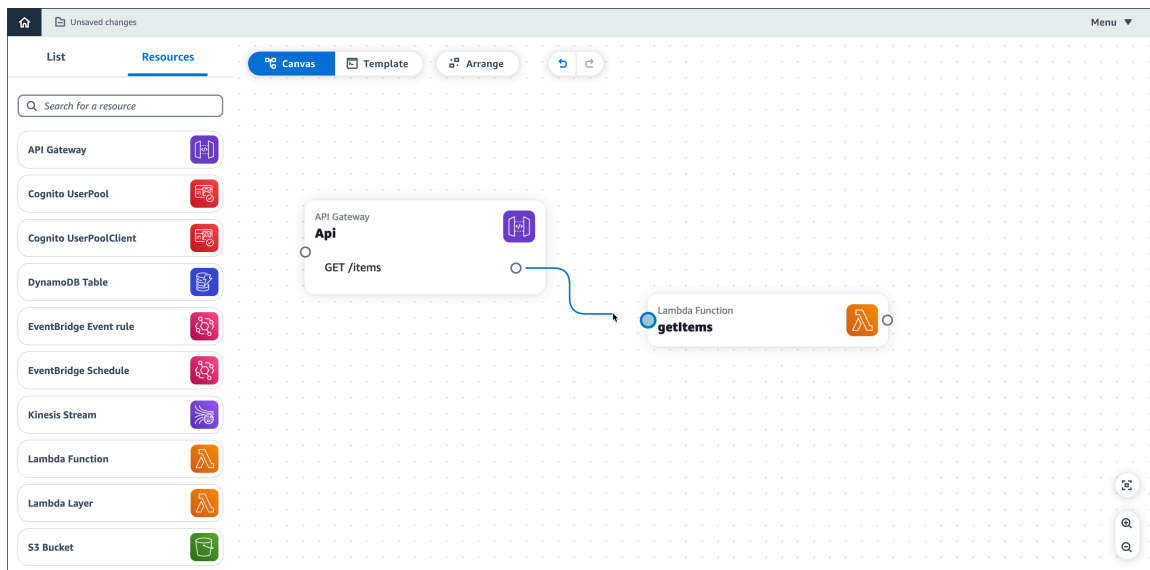


Para obtener más información, consulte las secciones siguientes.

## 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 Políticas (IAM)

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

## Events (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 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

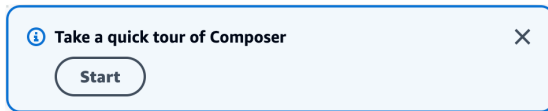
- [Haga un recorrido por la consola de Infrastructure Composer](#)
- [Cargue y modifique el proyecto de demostración de Infrastructure Composer](#)
- [Cree su primera aplicación con Infrastructure Composer](#)

## 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](#).

## Siguientes pasos

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 solicitud](#)
- [Siguientes pasos](#)

## Paso 1: Abre la demostración

Comience a usar Infrastructure Composer creando un proyecto de demostración.

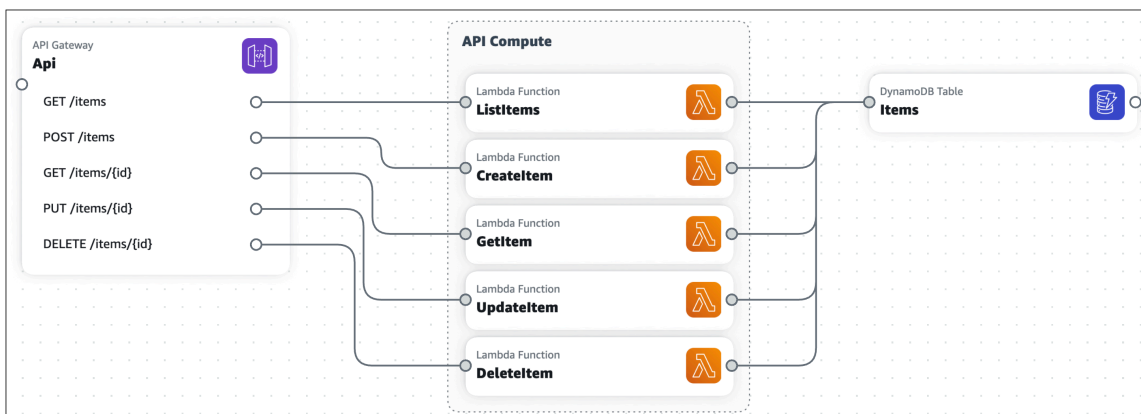
Para crear un proyecto de demostración

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:

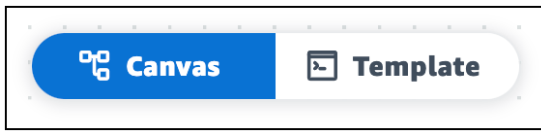


## Paso 2: Explore el lienzo visual de Infrastructure Composer

Conozca las características del lienzo visual para crear 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.



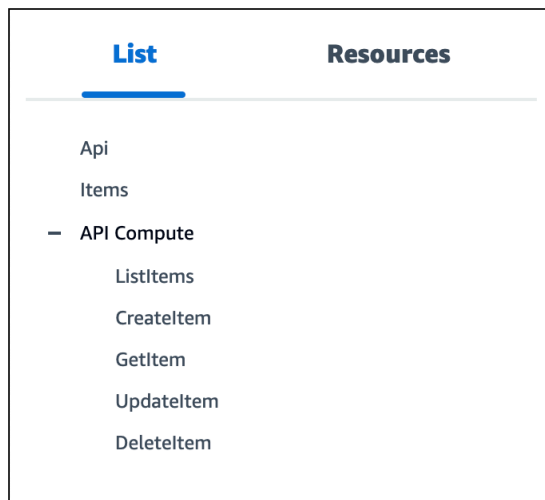
Para mostrar el código de infraestructura de la aplicación en el área de visualización principal, elija Plantilla. Por ejemplo, esta es la vista de plantilla AWS Serverless Application Model (AWS SAM) del proyecto de demostración de Infrastructure Composer.

 A screenshot of the 'Template' view in the application. It shows a code editor with the following content:
 

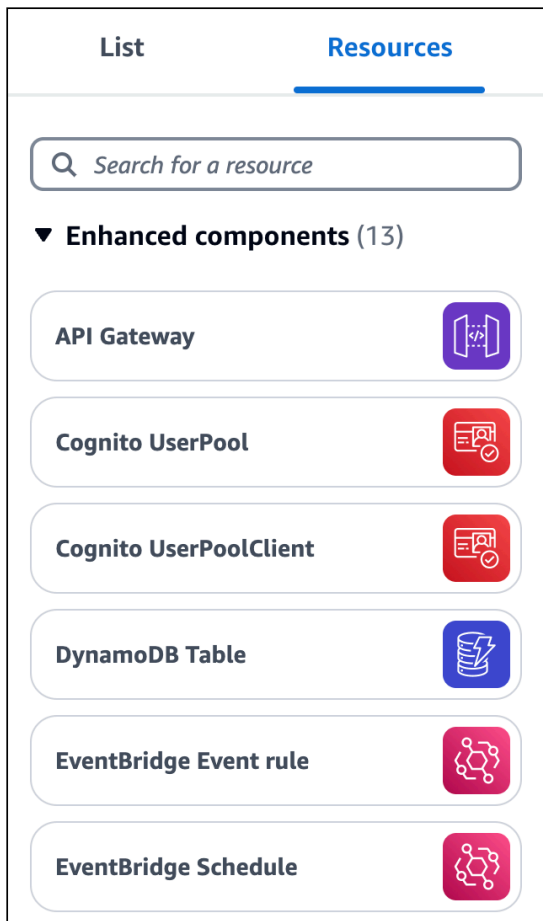
```

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 en 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.



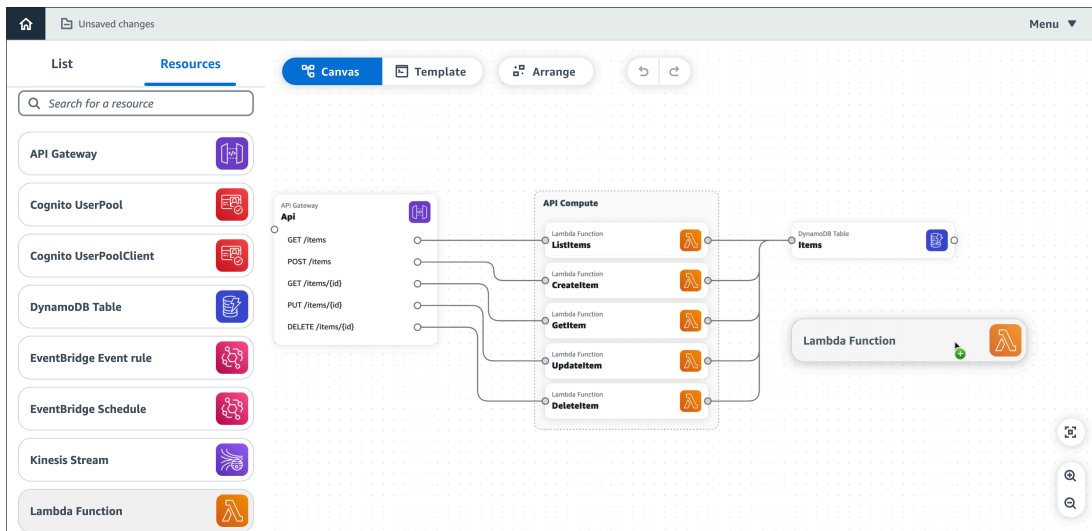
5. Para moverse por el lienzo visual, utilice gestos básicos. Para obtener más información, consulte [Coloque las tarjetas en el lienzo](#).

### Paso 3: Amplíe la arquitectura de su aplicación

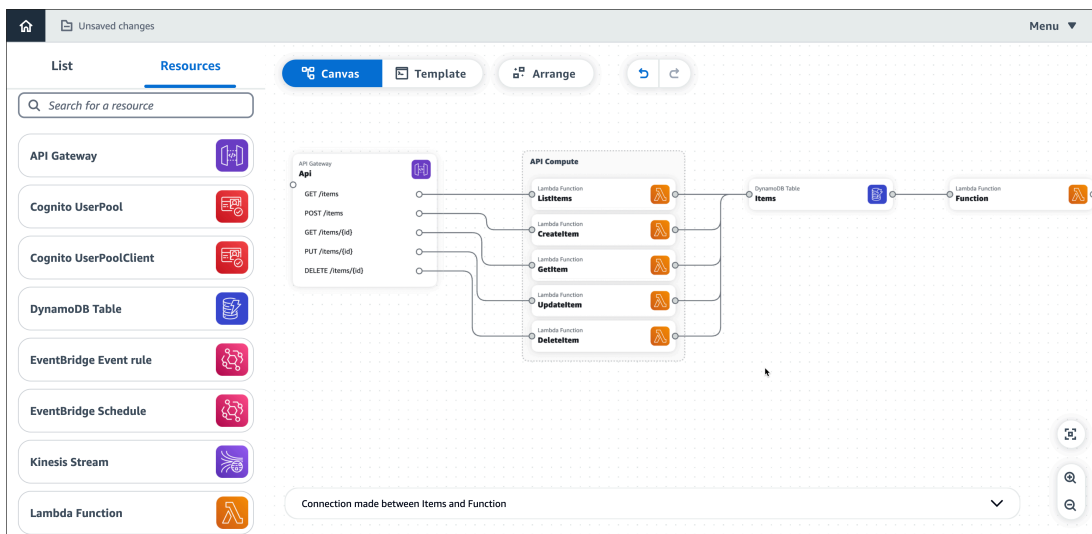
En este paso, ampliará la arquitectura de la aplicación añadiendo una función Lambda a la tabla de DynamoDB.

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](#) la referencia de AWS SAM recursos y propiedades.

## Paso 4: Guarde la solicitud

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](#).


## Siguientes pasos

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) sin servidor que gestione los usuarios de una base de datos.

Para este tutorial, utilizamos Infrastructure Composer en. Consola de administración de AWS 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: Configurar la API REST de API Gateway](#)
- [Paso 4: Configurar 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](#)
- [Siguiendo pasos](#)

## Referencia de propiedades del recurso

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

Método	Ruta	Nombre de la función
GET	/artículos	Obtener artículos
GET	/items/ {id}	getItem
PUT	/items/ {id}	Actualizar artículo
POST	/artículo	Añadir artículo
DELETE	/elementos/ {id}	deleteItem

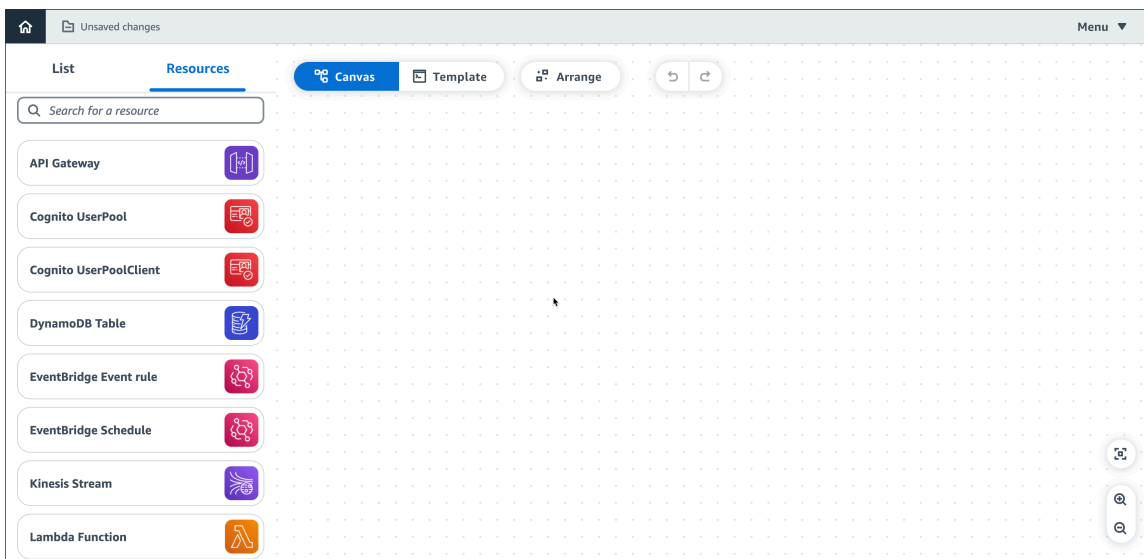
## Paso 1: Crea tu proyecto

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

Para crear un nuevo proyecto en blanco

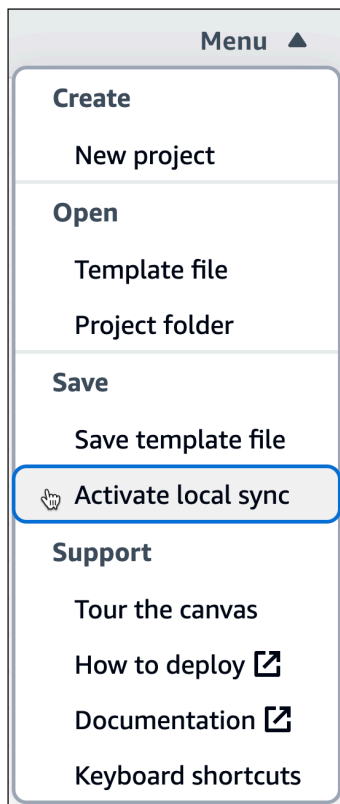
1. Inicie sesión en la [consola de Infrastructure Composer](#).
2. En la página de inicio, elija Crear proyecto.

Como se muestra en la siguiente imagen, 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 la API de acceso al sistema de archivos. 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.

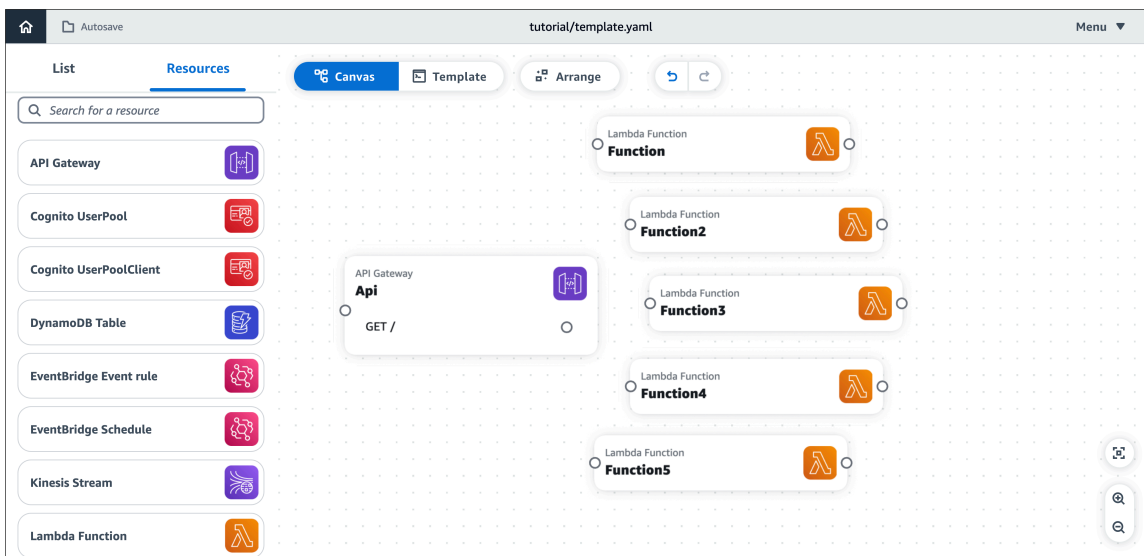
## Paso 2: Añade cartas al lienzo

Comience a diseñar la arquitectura de su aplicación con tarjetas de componentes mejoradas, empezando por una API REST de API Gateway y cinco funciones de 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 tarjeta API Gateway 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: Configurar la API REST de API Gateway

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

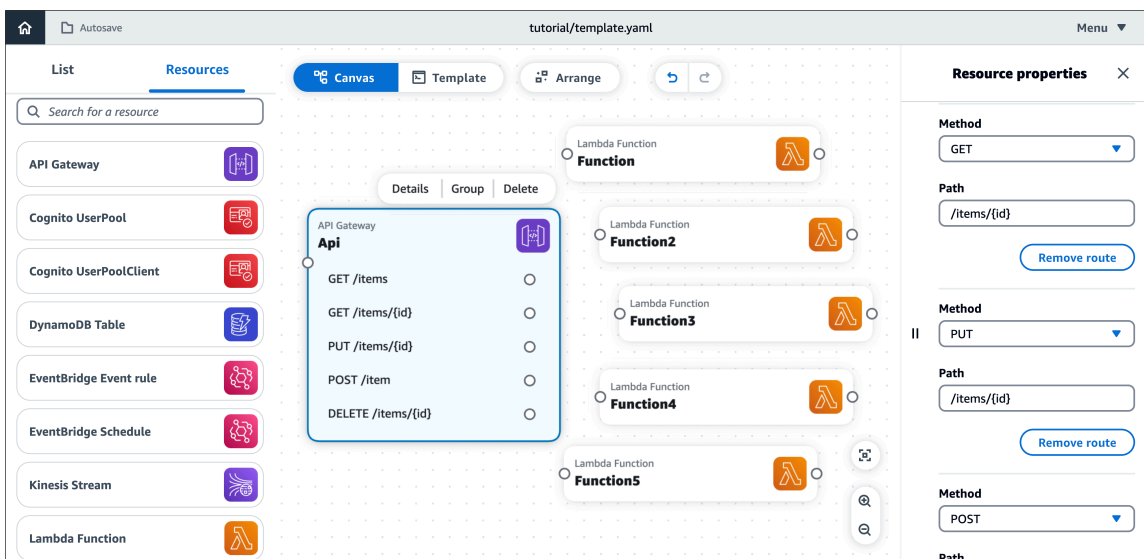
Para añadir rutas a la tarjeta API Gateway

1. Abre el panel de propiedades del recurso de la tarjeta API Gateway. 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 el método HTTP y los valores de ruta especificados en la [tabla de referencia de propiedades del recurso](#).

- a. En Método, elija el método HTTP 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 Save.



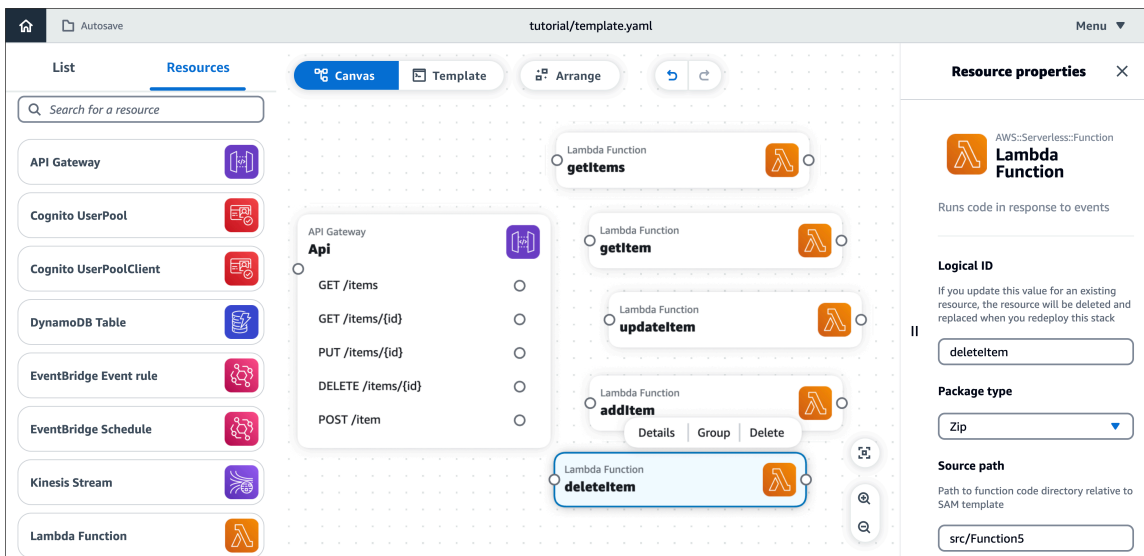
## Paso 4: Configurar las funciones de Lambda

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 Save.
4. Repita los pasos anteriores hasta que haya asignado un nombre a las cinco funciones.

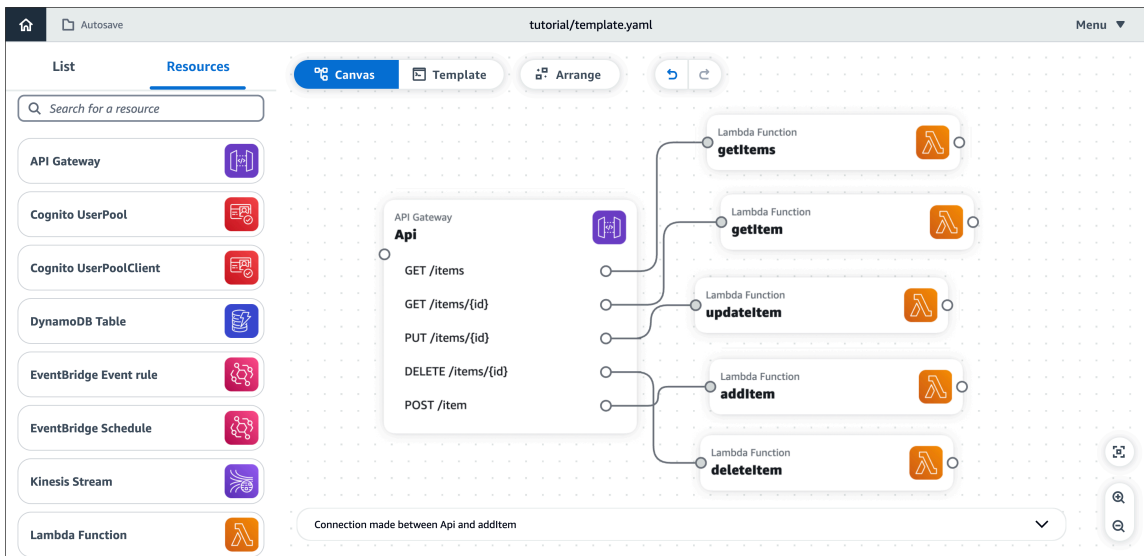


## Paso 5: Conecta tus tarjetas

Connect cada ruta de la tarjeta API Gateway a la tarjeta de función Lambda correspondiente, 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 API Gateway 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 que haya conectado las cinco rutas de la tarjeta API Gateway 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.

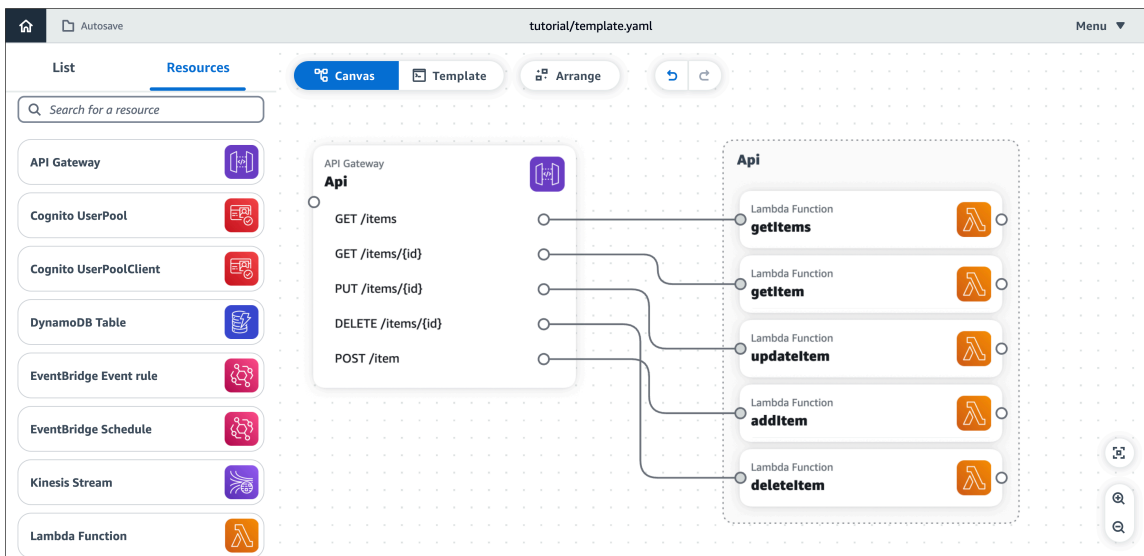
Para asignar un nombre a su grupo

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 Save.

Para organizar las tarjetas

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

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

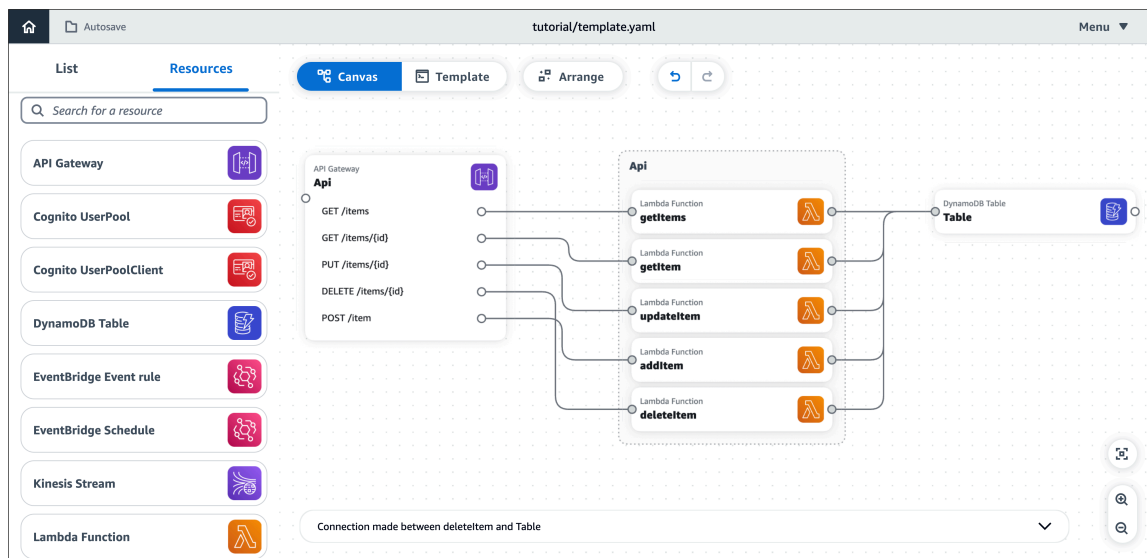


## Paso 7: Agregar y conectar una tabla de DynamoDB

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.

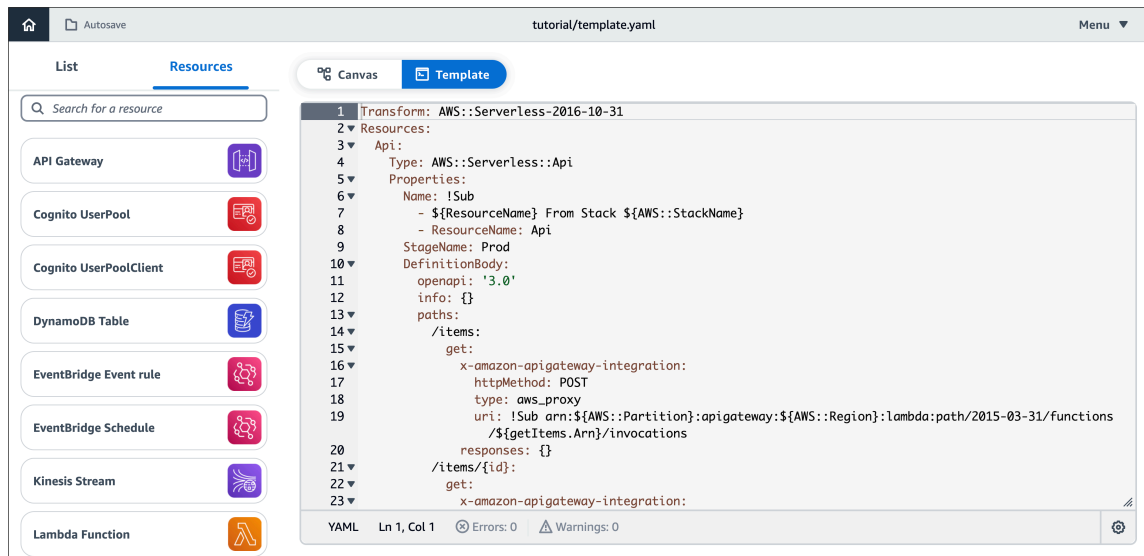


## Paso 8: Revise la plantilla AWS CloudFormation

¡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 la API REST de API Gateway 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 su grupo de recursos (API). Para obtener más información sobre esta sección, consulte [Metadatos](#) en la Guía del AWS CloudFormation usuario.



## 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 al IDE de su máquina local para acelerar el desarrollo. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su IDE local](#).
- Con la sincronización local, puede usar 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](#).

## Siguientes pasos

Ahora está preparado 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 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 IDE local. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su IDE local](#).

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 CloudFormation](#).

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

## Temas

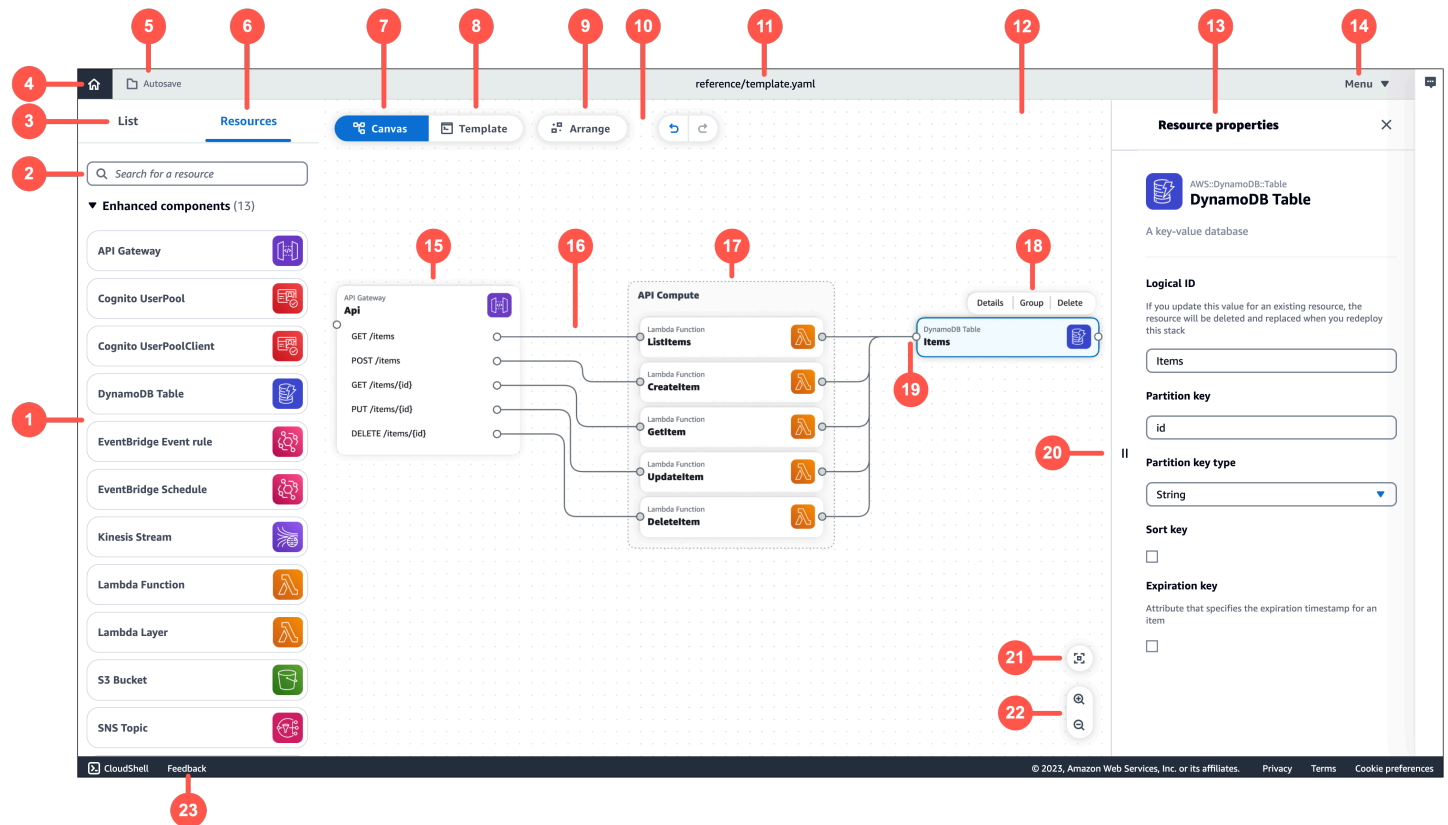
- [AWS Infrastructure Composer descripción visual de la consola](#)
- [Administre su proyecto desde la consola de Infrastructure Composer](#)



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 CloudFormation o que defina el código de su 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 sea compatible con la API de acceso al sistema de archivos, necesaria 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 la API de acceso al sistema de archivos, necesaria 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 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 Save.

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

Para conectar la consola de Infrastructure Composer con su entorno de desarrollo integrado (IDE) local, 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**

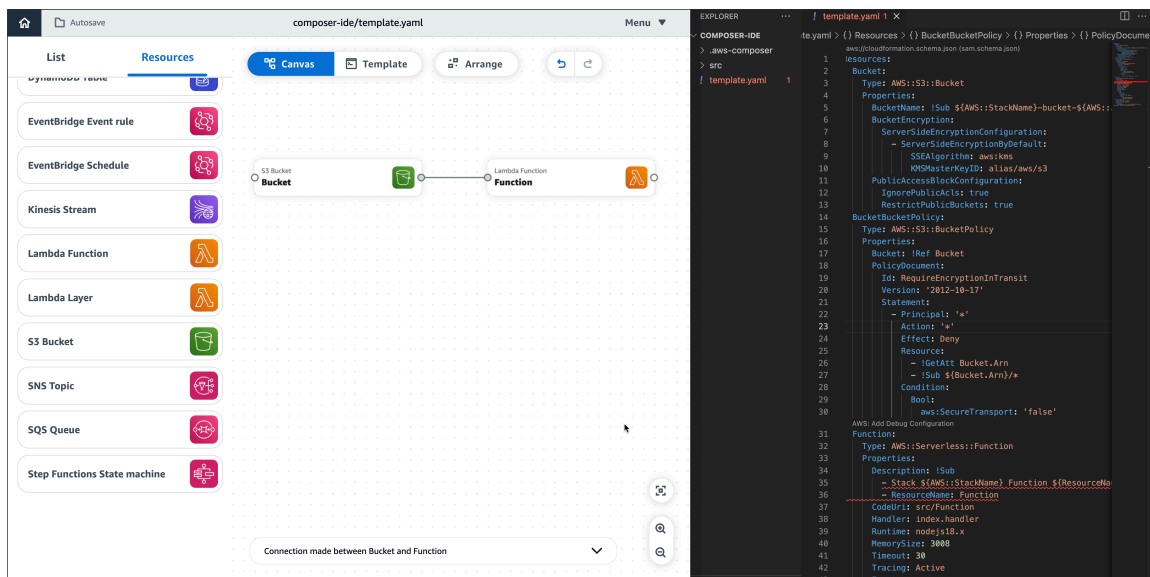
La opción Activar la sincronización local no está disponible en todos los navegadores. Está disponible en Google Chrome y Microsoft Edge.

## Ventajas de usar Infrastructure Composer con su IDE local

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

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

Puede usar herramientas locales, como la interfaz de línea de AWS Serverless Application Model comandos (AWS SAM CLI), para crear, probar e implementar su aplicación y más. 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 del IDE local.

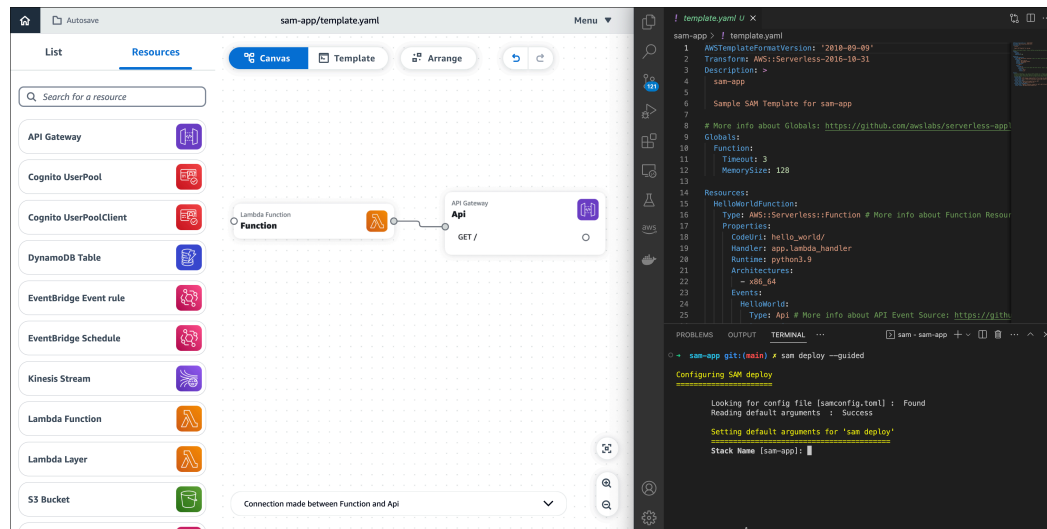


## Integre Infrastructure Composer con su IDE local

Para integrar Infrastructure Composer con su IDE local

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.





## 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 compatible con la API de acceso al sistema de archivos. Todas las versiones recientes de Google Chrome y Microsoft Edge admiten todas las funciones de la API de acceso al sistema de archivos y se pueden utilizar con el modo de sincronización local en Infrastructure Composer.

La API de acceso al sistema de archivos 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 tu 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 la API de acceso al sistema de archivos, consulte:

- La [API de acceso al sistema de archivos](#) está en los documentos web de mdn.
- [La API de acceso al sistema de archivos: 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 la API de acceso al 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

La API de acceso al sistema de archivos 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 la utiliza Windows Subsystem for Linux (WSL), la API de acceso al sistema de archivos excluye el acceso a todo el Linux directorio debido a su ubicación en el sistema. Windows Puede usar Infrastructure Composer con la sincronización local desactivada o configurar una solución para sincronizar los archivos del proyecto desde su WSL directorio con un directorio de trabajo. 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 su IDE local para acelerar el desarrollo. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su IDE local](#).

## ¿Qué guarda el modo de sincronización local

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

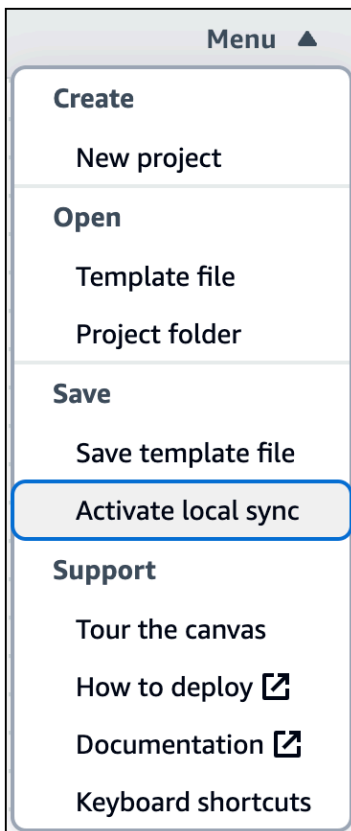
- Archivo de plantilla de aplicación: la plantilla AWS CloudFormation or AWS Serverless Application Model (AWS SAM) que contiene la infraestructura como código (IaC).
- Carpetas de proyectos: estructura de directorios general que organiza AWS Lambda las funciones.
- Directorio de respaldo: un directorio 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 la API de acceso al sistema de archivos. 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, siga 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 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 la API de acceso al sistema de archivos. 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.

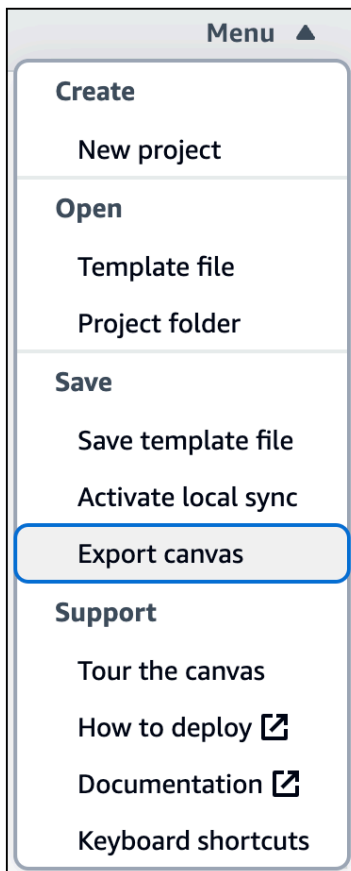
Para obtener una visión general visual de todas las funciones de Infrastructure Composer, consulte [AWS Infrastructure Composer descripción visual de la consola](#).

### 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 es png.
- 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.



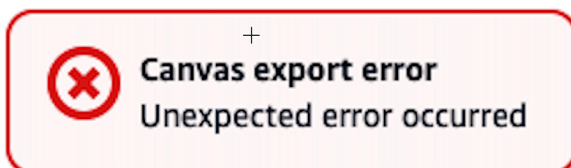
## Exportación de lienzo

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 CloudFormation las plantillas. También puede usar esta herramienta para crear y editar 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 CloudFormation consola. Esto le permite utilizar Infrastructure Composer directamente en ella 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 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 CloudFormation Designer. Recomendamos usar Infrastructure Composer para visualizar CloudFormation las plantillas. También puede usar esta herramienta para crear y editar 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 CloudFormation stack/ Infrastructure Composer implementada.

1. Ve a la [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 [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. Selecciona 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 CloudFormation recomendadas](#) en la Guía del 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.

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 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 [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 CloudFormation, seleccione Actualizar plantilla.
8. Selecciona 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. Selecciona 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 CloudFormation recomendadas](#) en la Guía del 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.

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 sincronizar, usa el `sam sync` comando de AWS SAMCLI. En esta sección también se proporcionan instrucciones sobre su Amazon Q uso en Infrastructure Composer desde AWS Toolkit for Visual Studio Code.

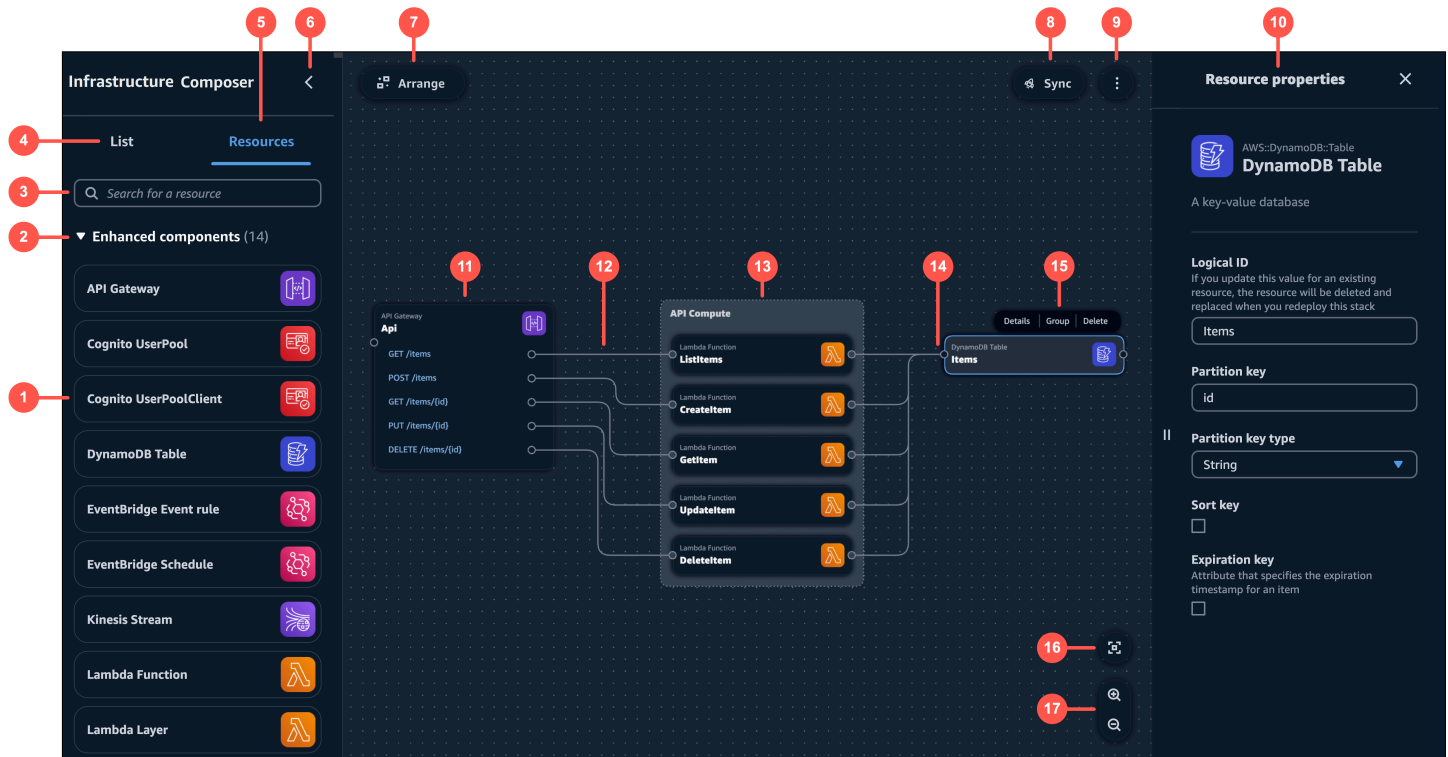
Para obtener más información sobre el uso de Infrastructure Composer desde el AWS Toolkit for Visual Studio Code, consulte [¿Cómo componer](#). El contenido de esta sección se aplica a esta experiencia, así como a la experiencia de la consola de Infrastructure Composer.

## 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 CLI `aws sam sync` comando AWS Serverless Application Model (AWS SAM) 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](#).

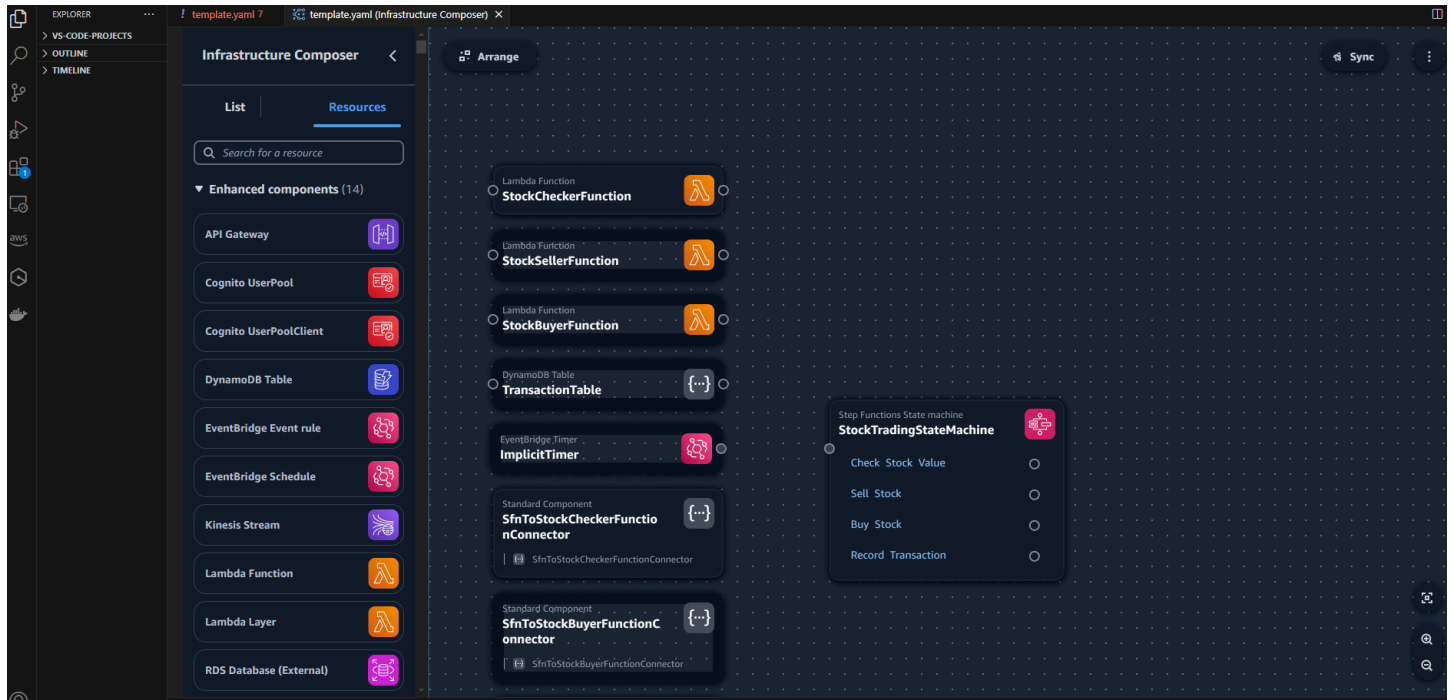
Para acceder a Infrastructure Composer desde el Toolkit for VS Code

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

1. Al seleccionar el botón Infrastructure Composer desde cualquier AWS SAM plantilla CloudFormation o plantilla.
2. A través del menú contextual haciendo clic con el botón derecho en su AWS SAM plantilla 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 su 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.

- En el caso de los recursos que respaldan el AWS servicio APIs, los AWS SAM CLI utilizarán APIs para implementar los cambios. Lo AWS SAM CLI hace para actualizar rápidamente sus recursos en la nube.
- Si es necesario, 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 AWS SAM CLI tenerla instalada en su máquina local. Para obtener instrucciones, consulte [Instalación del AWS SAMCLI](#) en la Guía para AWS Serverless Application Model desarrolladores.

Al utilizar la función de sincronización de Infrastructure Composer, AWS SAM CLI hace referencia al archivo de configuración para obtener la información necesaria para sincronizar la aplicación con el 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 Aceptar 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.
  - CloudFormation nombre de la pila: el nombre de la 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. Aquí AWS SAM CLI empaquetará y almacenará los archivos de la aplicación y el código de función. Puede seleccionar un depósito existente o crear uno nuevo.

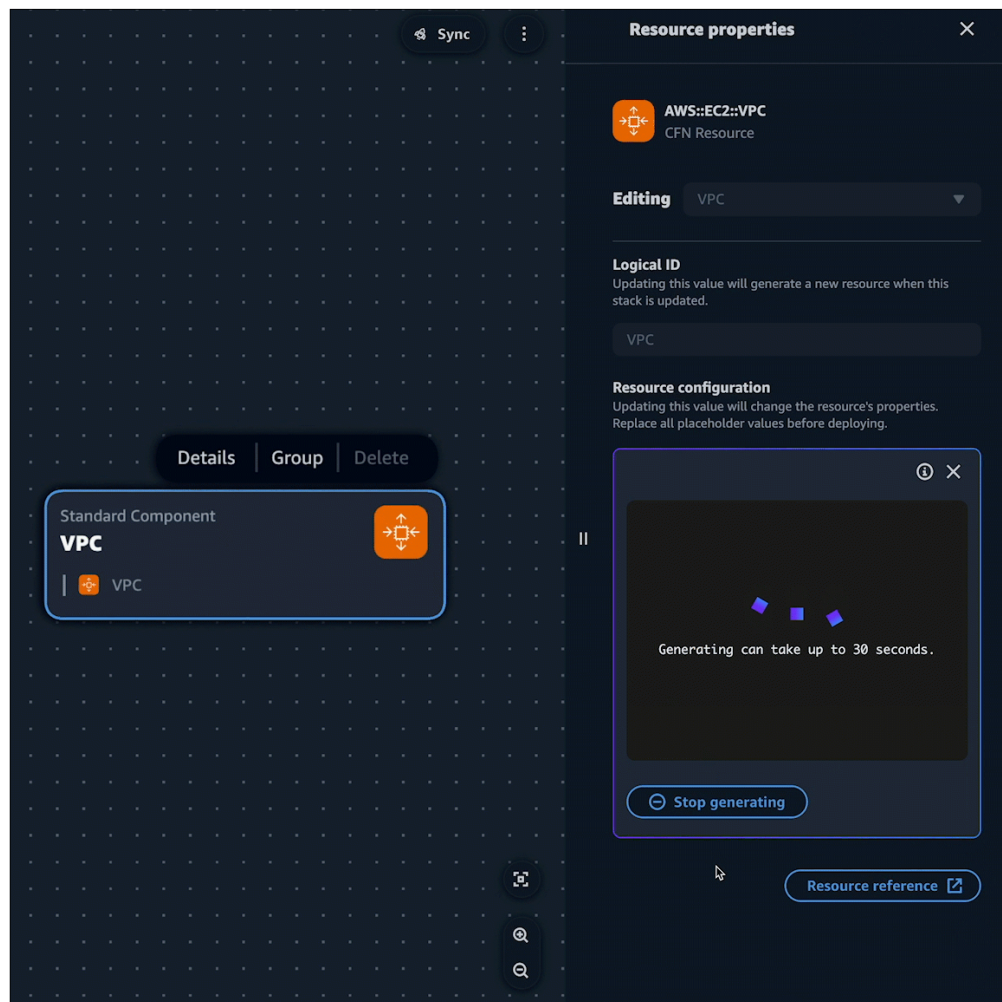
Infrastructure Composer iniciará el AWS SAM CLI `sam sync` comando y abrirá una ventana de terminal en el IDE para mostrar su progreso.

## 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. Puede usarlo Amazon Q en 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 la Guía Amazon Q Developer del usuario.

En el caso de las tarjetas de recursos y componentes estándar, puede Amazon Q utilizarlas para generar sugerencias de código de infraestructura para sus recursos.



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

## Configuración

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

## Utilización Amazon Q Developer en Infrastructure Composer

Se puede utilizar Amazon Q Developer desde el panel de propiedades del recurso de cualquier tarjeta de recursos o componentes estándar.

## Para usar 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. Seleccione 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, seleccione 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 acerca de Amazon Q, consulte [¿Qué es Amazon Q?](#) en la Guía del usuario de Amazon Q Developer.

# Cómo componer en AWS Infrastructure Composer

En esta sección se describen los aspectos básicos del uso de Infrastructure Composer desde [Consola Infrastructure Composer CloudFormation 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.

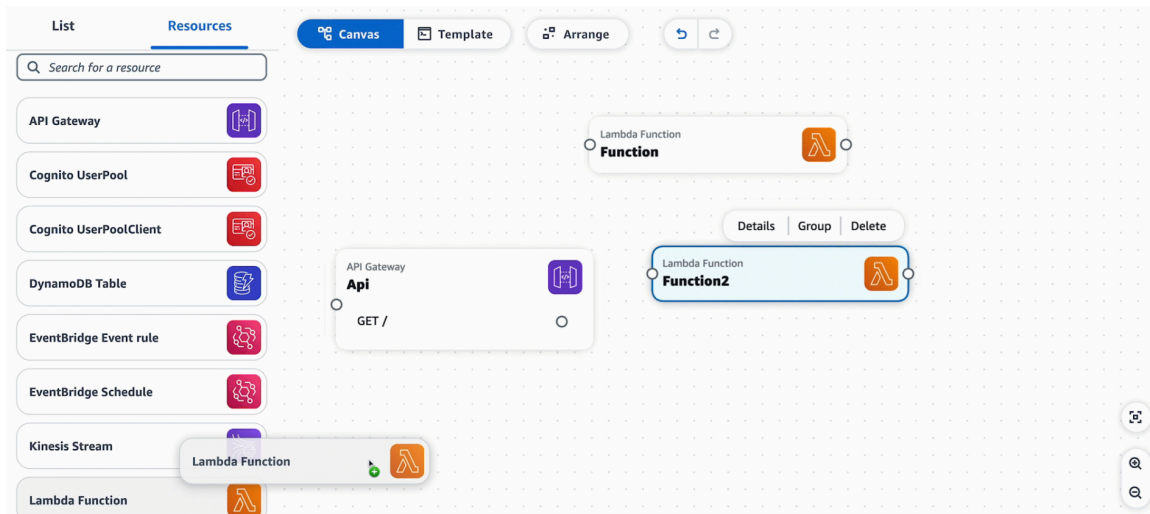
## Temas

- [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 \(Amazon VPC\)](#)

## 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](#)

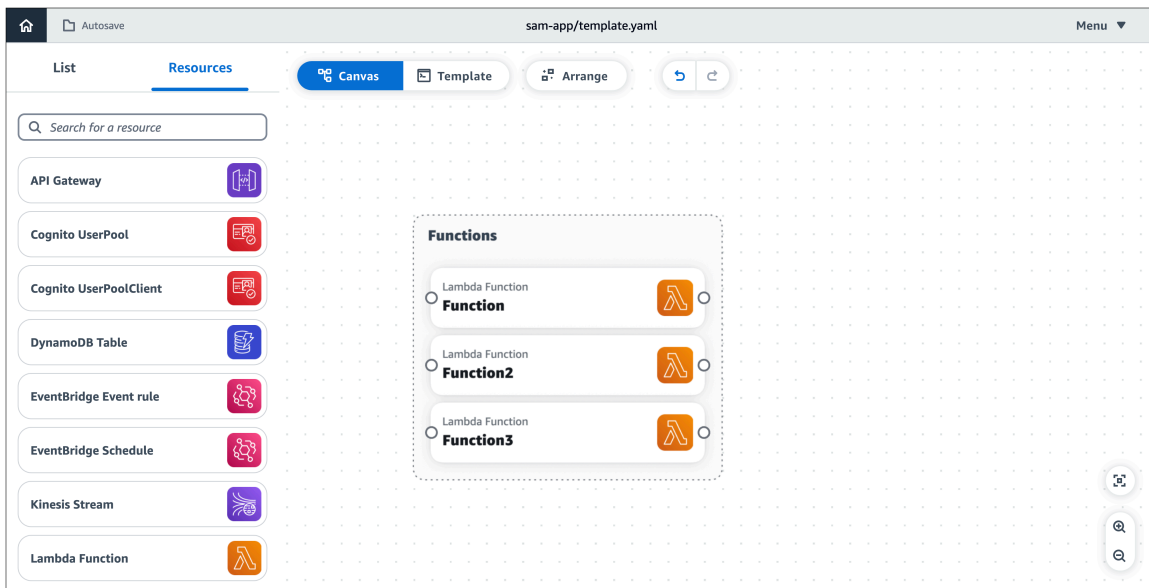
## Agrupar las tarjetas en el lienzo visual de Infrastructure Composer

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

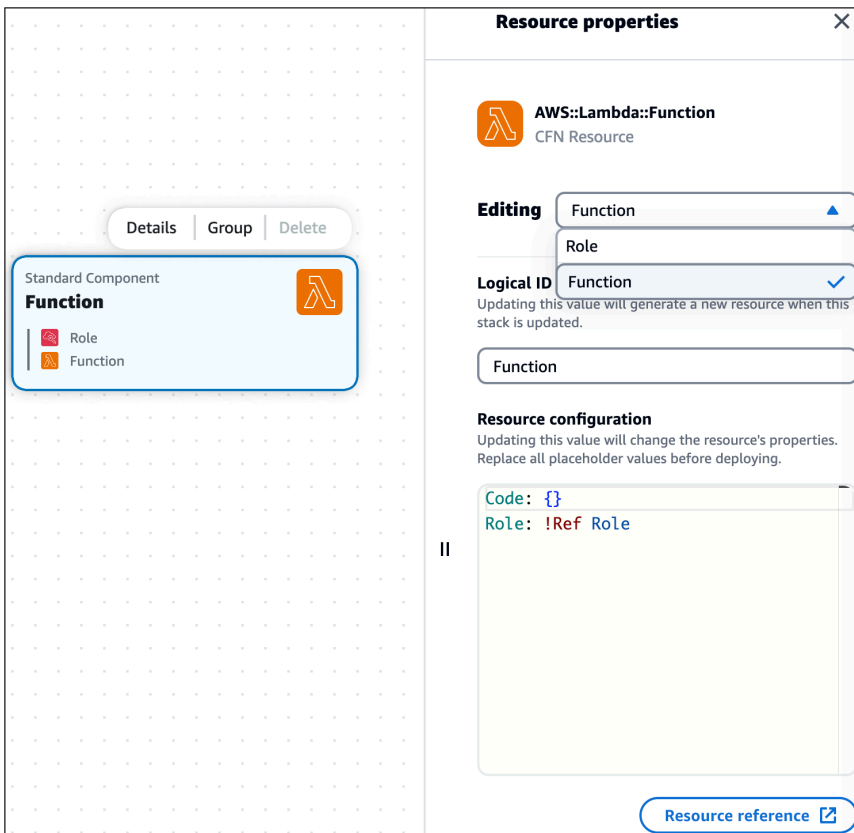
Hay dos formas de agrupar las tarjetas con componentes mejorados:

- 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:



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

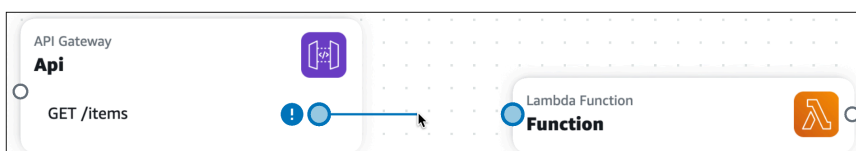
Utilice este tema para entender cómo conectar tarjetas en Infrastructure Composer. En esta sección se incluyen detalles sobre la conexión de tarjetas de componentes mejorados y tarjetas de componentes estándar. También proporciona algunos ejemplos que ilustran las diferentes formas en que se pueden conectar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

### Conexión de tarjetas de componentes mejorados

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

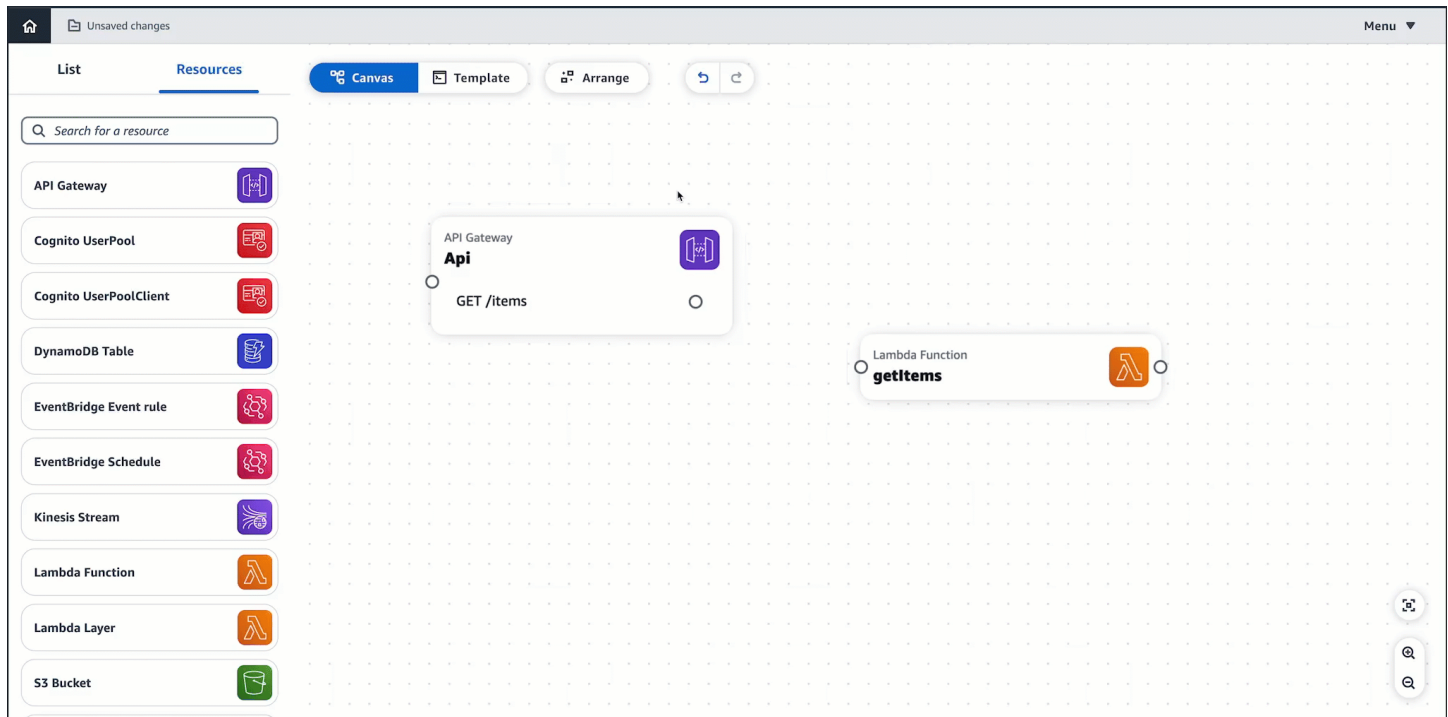
- Un puerto en el lado derecho de una tarjeta indica la posibilidad de que la tarjeta invoque a otra tarjeta.
- Un puerto en el lado izquierdo de una carta indica la posibilidad de que otra carta invoque la carta.

Conecte las tarjetas 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.

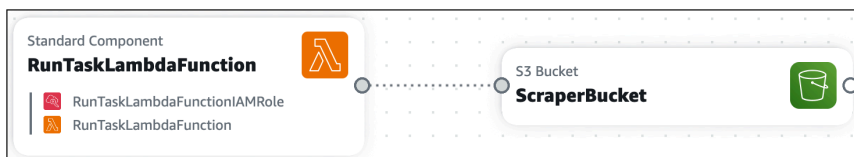
- Si se realiza correctamente, haga clic en el mensaje para ver el inspector de cambios. Aquí puede ver lo que Infrastructure Composer modificó para aprovisionar su conexión.
- Si no lo consigue, aparecerá un mensaje. Puede seleccionar la vista de plantilla y actualizar manualmente el código de infraestructura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

```
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

```

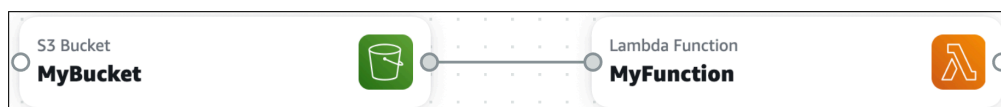
En el ejemplo anterior, el fragmento de código que aparece `ApiGatewayMethod:` debajo `Integration:` especifica la relación basada en eventos que conecta las dos tarjetas.

## 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
    ...

```

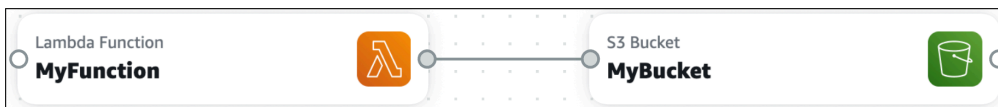
```

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 operaciones CRUD en los elementos del bucket de Amazon S3.



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

- Políticas de IAM 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

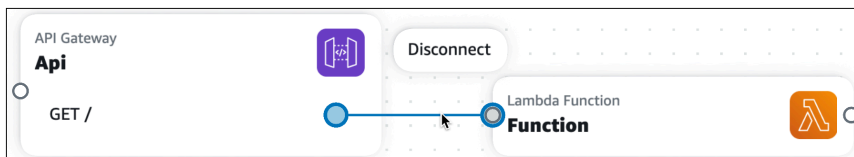
```

## Desconectar tarjetas en Infrastructure Composer

En Infrastructure Composer, AWS los recursos se conectan y desconectan mediante tarjetas de componentes mejoradas 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 de Lambda que está conectada a una API REST de Amazon API Gateway:

```

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
```

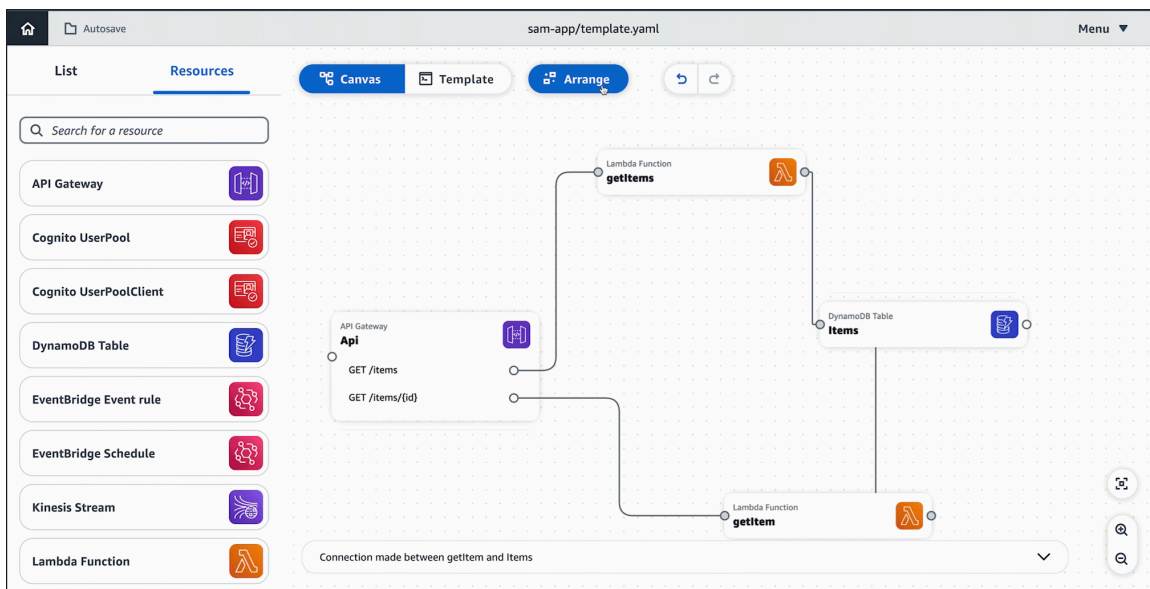
```

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 a las que se MyLambdaFunction indican 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 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 laC 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.

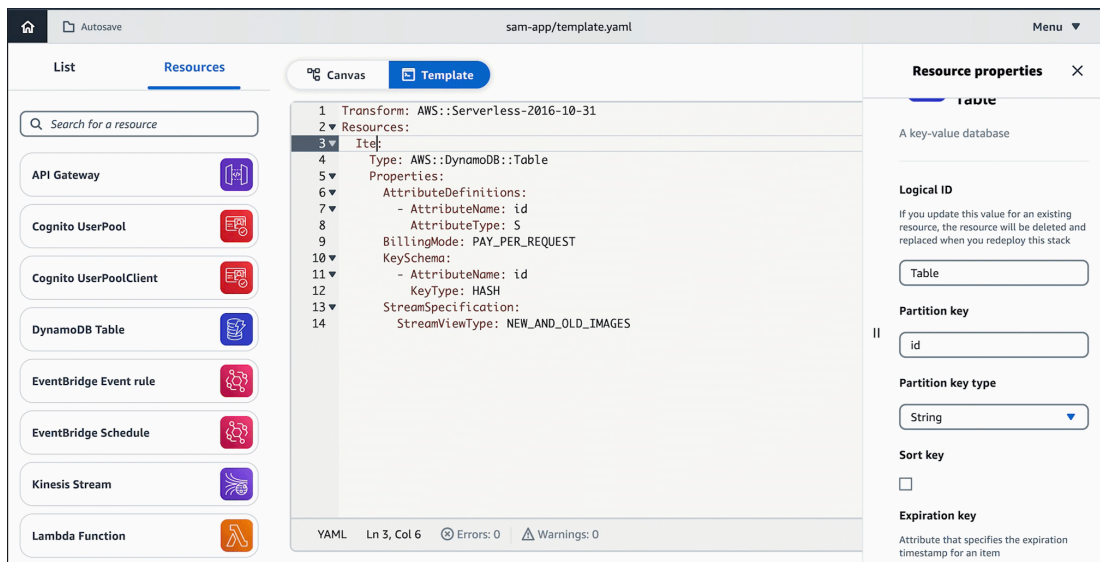
Para obtener más información sobre las tarjetas con componentes mejorados, consulte [Tarjetas de componentes mejoradas en Infrastructure Composer](#) y [Tarjetas de componentes mejoradas en Infrastructure Composer](#)

## 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 Consola de administración de AWS, 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 (Amazon RDS)

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

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

- DBClúster o instancia de Amazon RDS que está definido en otra plantilla
- Proxy de Amazon RDS DB

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



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

Puede conectar la aplicación al DB clúster o instancia de Amazon RDS externo mediante una función Lambda.

## Requisitos

Para utilizar esta función, debe cumplir los siguientes requisitos:

1. Debe utilizar su DB clúster, instancia o proxy externo de Amazon RDS AWS Secrets Manager para administrar la contraseña de usuario. Para obtener más información, consulte [Administración de contraseñas con Amazon RDS y AWS Secrets Manager](#) en la Guía del usuario de Amazon RDS.
2. La aplicación en Infrastructure Composer debe ser un proyecto nuevo o debe haberse creado originalmente en Infrastructure Composer.

## Procedimiento

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

Desde la paleta Recursos, arrastre al lienzo una tarjeta de componentes mejorada (externa) de RDS Database.

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:

**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 exclusivo para su DB clúster, instancia o proxy externo de Amazon RDS. Este ID no tiene que coincidir con el valor de ID lógico de su DB recurso externo de Amazon RDS.
- Secreto de base de datos: identificador del AWS Secrets Manager secreto asociado a su DB clúster, instancia o proxy de Amazon RDS. Este campo acepta los siguientes valores:
  - Valor estático: identificador único del secreto de la base de datos, como el ARN secreto. 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 parámetros de SSM: puede almacenar su secreto en el almacén de parámetros de SSM y especificar su valor mediante una referencia dinámica. Por ejemplo, `{{resolve:ssm:MySecret}}`. Para obtener más información, consulte los [parámetros del SSM](#) en la Guía del AWS CloudFormation usuario.

- Nombre de host de la base de datos: el nombre de host que se puede usar para conectarse a su DB clúster, instancia o proxy de Amazon RDS. Este valor se especifica en la plantilla externa que define el recurso de Amazon RDS. 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 DB clúster, instancia o proxy de Amazon RDS implementado. Puede especificar el valor de salida mediante la función [Fn::ImportValue](#) intrínseca. Por ejemplo, `!ImportValue myStack-myDatabase-abcd1234`.
  - Valor del almacén de parámetros SSM: puede almacenar el nombre de host de la base de datos en el almacén de parámetros SSM 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 su DB clúster, instancia o proxy de Amazon RDS. Este valor se especifica en la plantilla externa que define el recurso de Amazon RDS. 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 DB clúster, instancia o proxy de Amazon RDS implementado. Por ejemplo, `!ImportValue myStack-MyRDSInstancePort`.
  - Valor del almacén de parámetros de SSM: puede almacenar el nombre de host de la base de datos en el almacén de parámetros de SSM 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 base de datos RDS (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 DB clúster, la instancia o el proxy externo de Amazon RDS

Al arrastrar una tarjeta de base de datos de RDS (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 CloudFormation de la 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 DB clúster, instancia o proxy de Amazon RDS.

La sección de [parámetros](#) de una 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 proporcione, Infrastructure Composer puede almacenar valores aquí para su DB clúster, instancia o proxy de Amazon RDS y especificarlos en toda la plantilla.

Los valores de cadena de la Parameters sección Metadata y utilizan el valor de ID lógico que especifique en la tarjeta de base de datos de RDS (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 base de datos (externa) de RDS, Infrastructure Composer aprovisiona variables de entorno y políticas AWS Identity and Access Management (IAM). 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 utilizar 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 su función a Secrets Manager para obtener su contraseña de acceso al DB clúster, instancia o proxy de Amazon RDS.

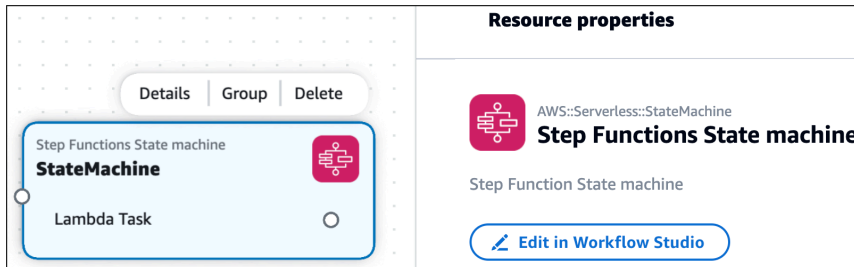
## Uso AWS Infrastructure Composer con AWS Step Functions

AWS Infrastructure Composer presenta una integración con [AWS Step FunctionsWorkflow 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 de Infrastructure Composer, puede utilizar las ventajas de dos diseñadores visuales potentes 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](#)
- [Introducción a 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 a los recursos, Infrastructure Composer crea automáticamente las políticas AWS Identity and Access Management (de IAM) necesarias para autorizar la interacción entre los 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  
  ...
```

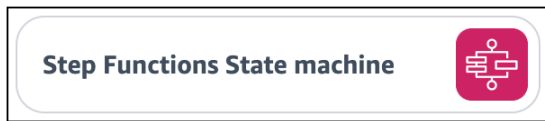
Si es necesario, puede añadir más políticas de IAM a su plantilla.

## Introducción a 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. Abra el panel de propiedades del recurso de la tarjeta y seleccione 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, elija 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:
  - **Workflow Studio** Vuelva a abrir y modifique el 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 el IDE local. 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 de aplicaciones definidas mediante plantillas AWS Serverless Application Model (AWS SAM). Utilice 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 utilizar 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, la ruta local del archivo se especifica 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](#).

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

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

## Uso de Step Functions Workflow Studio en Infrastructure Composer

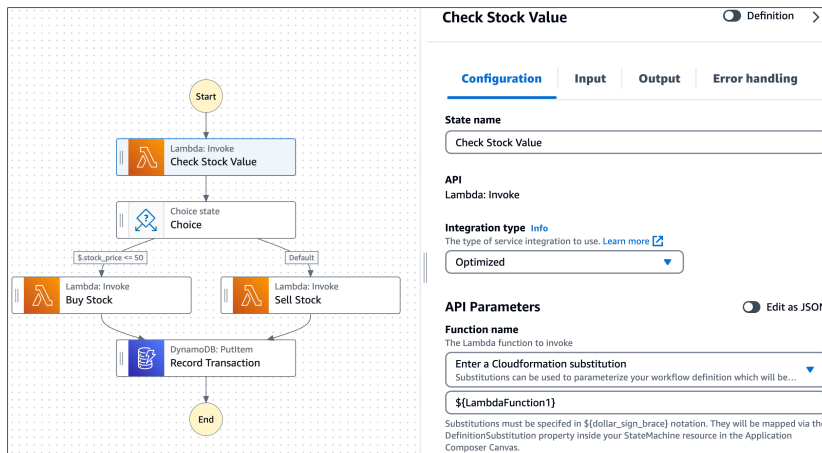
### Cree flujos de trabajo

Infrastructure 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 Infrastructure Composer.

### Para especificar una sustitución de definición en Workflow Studio

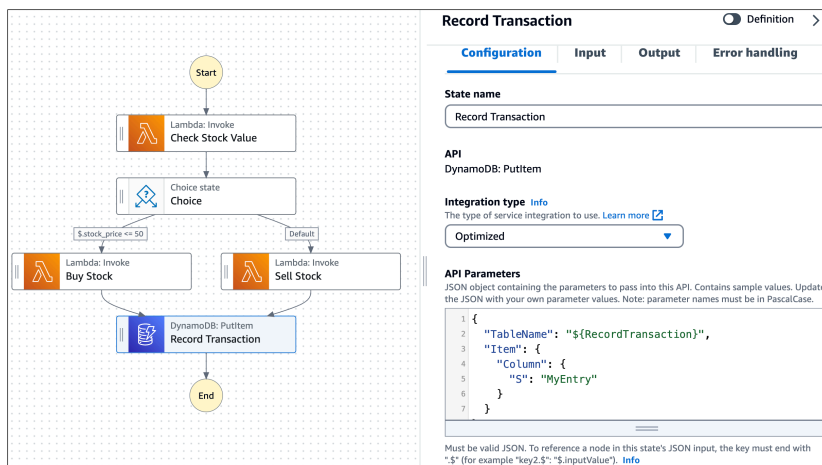
1. Abra la pestaña Configuración de la tarea y busque el campo Parámetros de la API.



- Si el campo Parámetros de la API tiene una opción desplegable, selecciona Introducir una CloudFormation sustitución. A continuación, proporciona un nombre único.

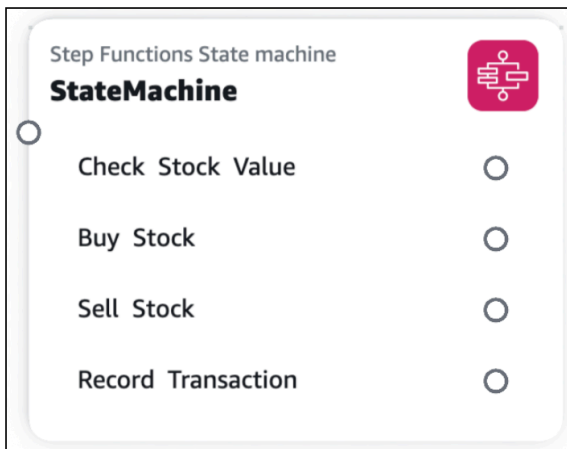
Para las tareas que se conectan al mismo recurso, especifique la misma sustitución de definición para cada tarea. Para usar una sustitución de definición existente, elija Seleccione una CloudFormation sustitución y seleccione la sustitución que desee utilizar.

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



- Seleccione Guardar y volver a Infrastructure Composer.

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



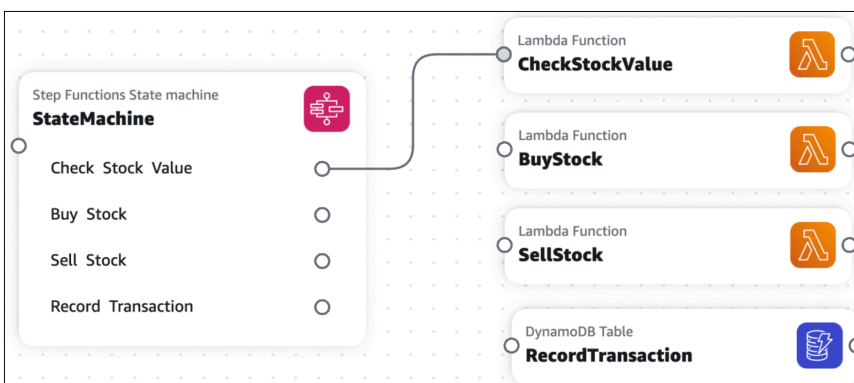
Connect los recursos a las tareas del flujo de trabajo

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 Consola de administración de AWS, 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 Save.

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:

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

#### Note

Se puede utilizar Amazon Q para generar sugerencias de códigos de infraestructura para 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.

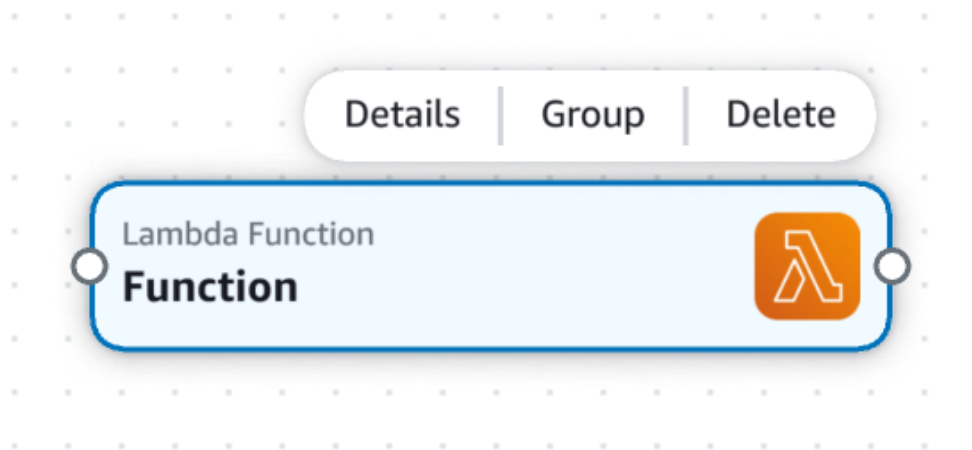
2. En el campo Edición, seleccione el recurso laC estándar que desee editar en la lista desplegable.
3. Modifique el código de infraestructura y guárdelo.

## 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 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

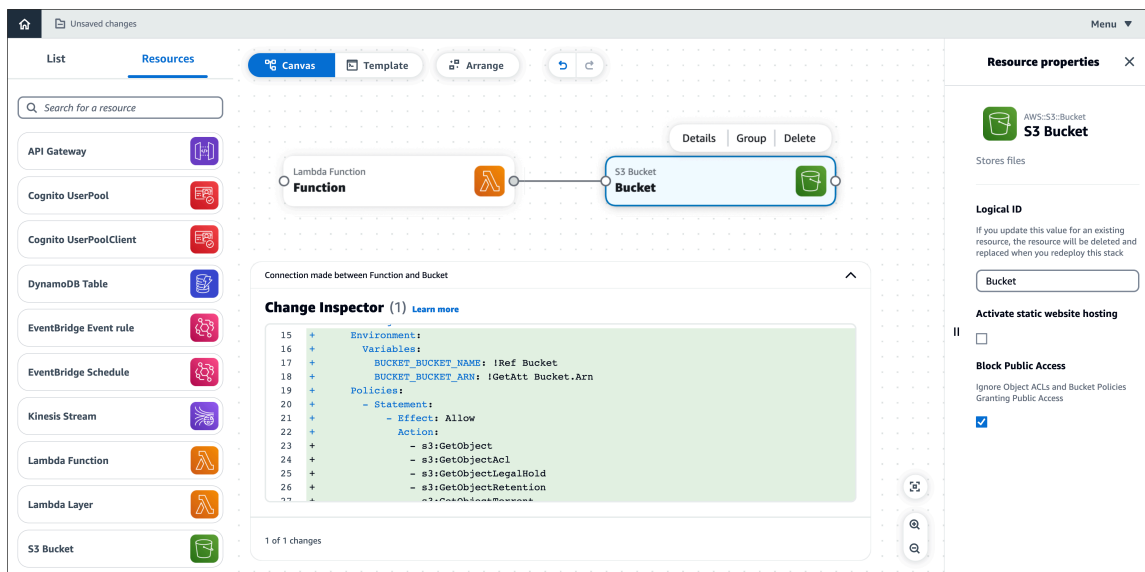
A medida que diseña en la consola de Infrastructure Composer, el código de infraestructura se crea automáticamente. Utilice el Inspector de cambios para ver las actualizaciones del código de la plantilla y obtener información sobre lo que Infrastructure Composer está creando para usted.

En este tema se describe el uso de Infrastructure Composer desde la extensión Consola de administración de AWS 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:



## Ventajas del Inspector de Cambios

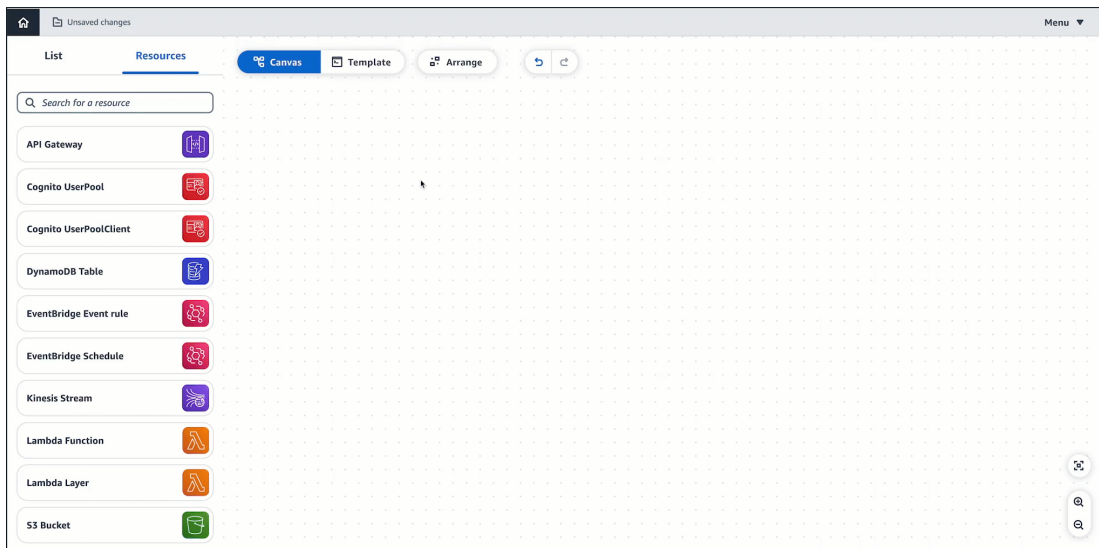
El Inspector de cambios es una excelente forma de ver el código de plantilla que Infrastructure Composer crea para usted. También es una excelente forma de aprender a escribir código de infraestructura. Al diseñar aplicaciones en Infrastructure Composer, consulte las actualizaciones

de código en el Inspector de cambios para obtener información sobre el código necesario para aprovisionar el diseño.

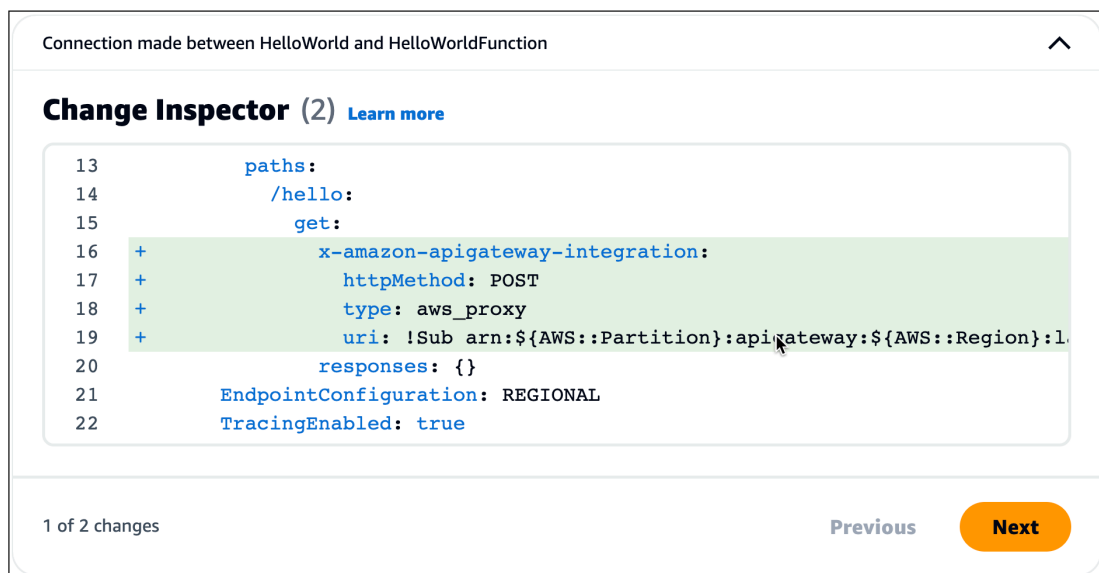
## Procedimiento

Para usar el Inspector de cambios

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

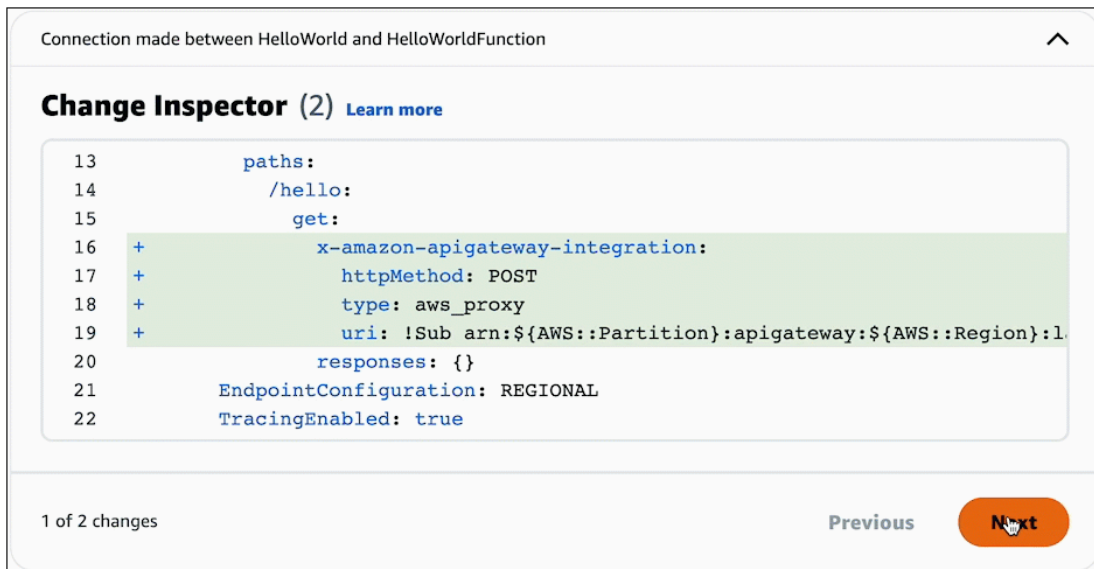


2. Vea el código que se ha redactado automáticamente para usted.



- a. El código resaltado en verde indica el código recién agregado.
- b. El código resaltado en rojo indica el código recién eliminado.
- c. 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 un IDE local y ver la plantilla completa en su máquina local. Para obtener más información, consulte [Conecte la consola de Infrastructure Composer con su IDE local](#).

## 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, puede tener varios recursos de la API REST de Amazon API Gateway descritos en una OpenAPI especificación. En lugar de replicar el

código de OpenAPI especificación en 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:

- REST API Recursos de API Gateway definidos por archivos de OpenAPI especificaciones externos.
- AWS Step Functions recursos de máquinas de estado definidos por archivos de definición de máquinas 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](#)
- [Consulte un archivo externo de OpenAPI especificaciones con Infrastructure Composer](#)

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

### Utilice Infrastructure Composer con un IDE local

Cuando usa Infrastructure Composer con un IDE local en modo de sincronización local, puede usar su IDE local 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 IDE local](#).

### 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 la AWS SAM CLI para facilitar este proceso automáticamente. 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 API Gateway o Step Functions, seleccione Detalles para que aparezca el panel de propiedades del recurso.
2. Busque 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 la API de acceso al sistema de archivos, 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 [Resolución 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 [Resolución de problemas](#).

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

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

### Ejemplo

1. En primer lugar, utilice el `sam init` comando AWS SAM CLI para inicializar una nueva aplicación denominada `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
...

```

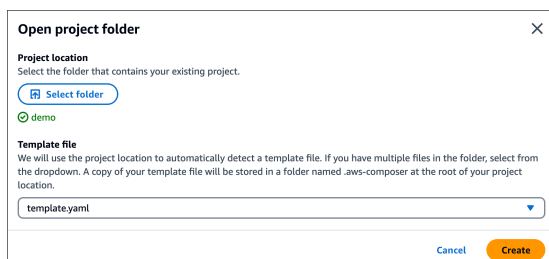
El archivo externo se encuentra en el `statemachine` subdirectorio de nuestra aplicación.

```

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

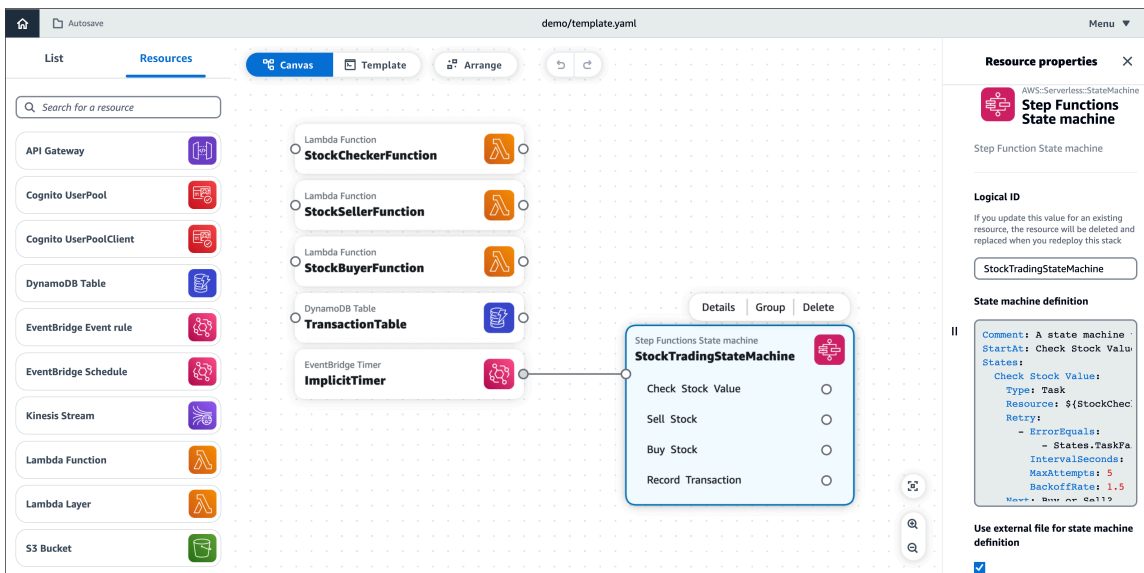
```

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.



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.



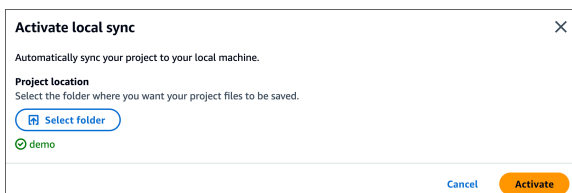
Cualquier cambio realizado en el archivo de definición de la máquina de estados se reflejará automáticamente en Infrastructure Composer.

## Consulte un archivo externo de OpenAPI especificaciones con Infrastructure Composer

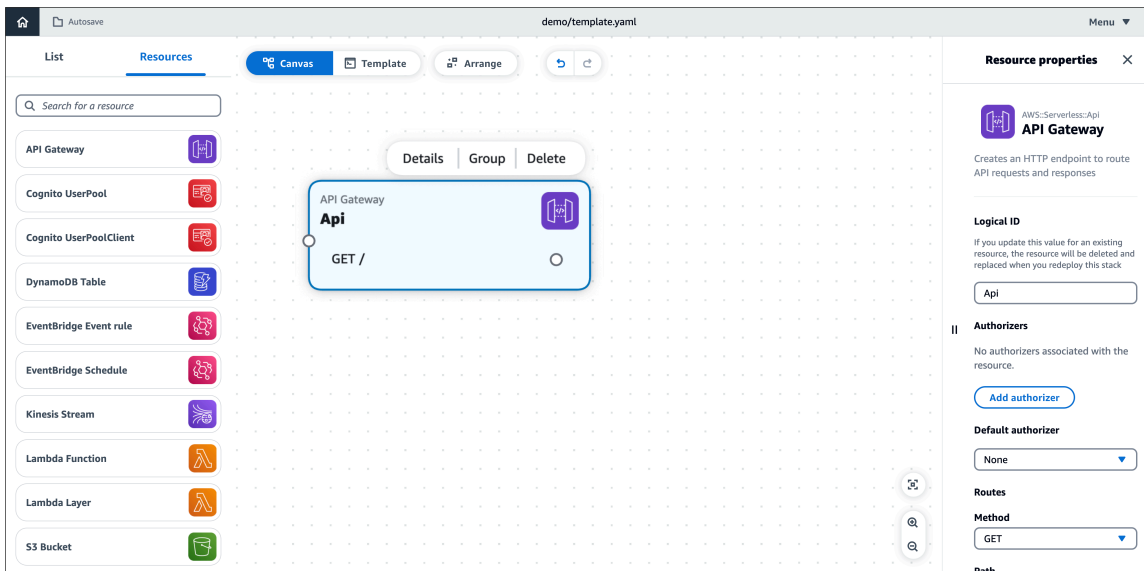
En este ejemplo, se utiliza Infrastructure Composer desde la consola para hacer referencia a un archivo de OpenAPI especificaciones externo que define una API Gateway REST API.

En primer lugar, cree un proyecto nuevo 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. Seleccione Detalles para que aparezca el panel de propiedades del recurso.



En el panel de propiedades del recurso, configura lo siguiente y guárdalo.

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

**Use external file for api definition**

**Relative path to external file**

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 `api-spec.yaml` se encuentra en la carpeta de su proyecto. Sustituya su contenido por lo siguiente:

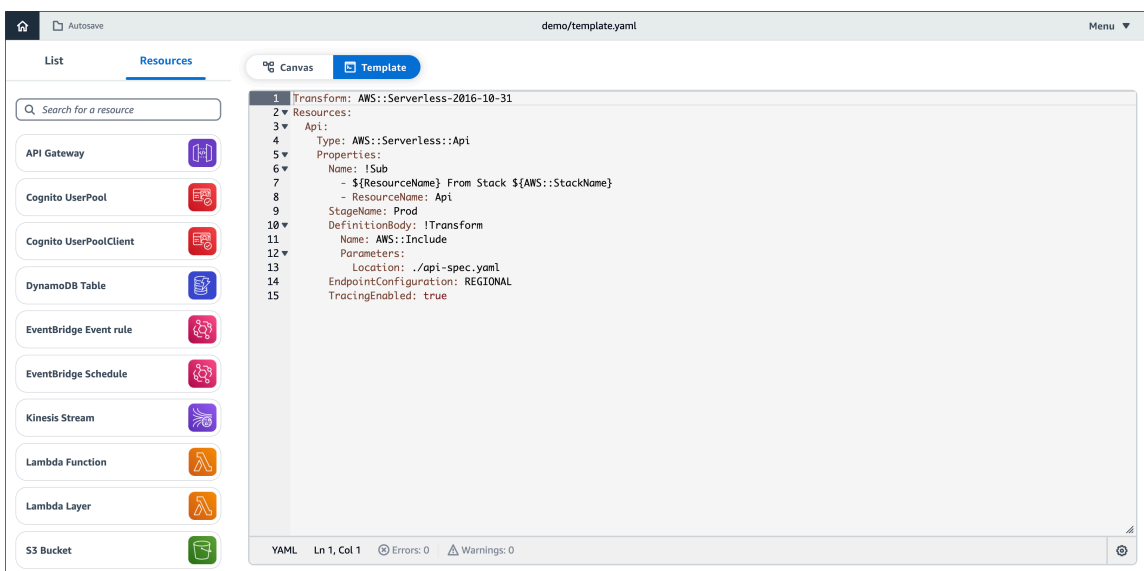
```
openapi: '3.0'
info: {}
paths:
```

```

/:
  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.

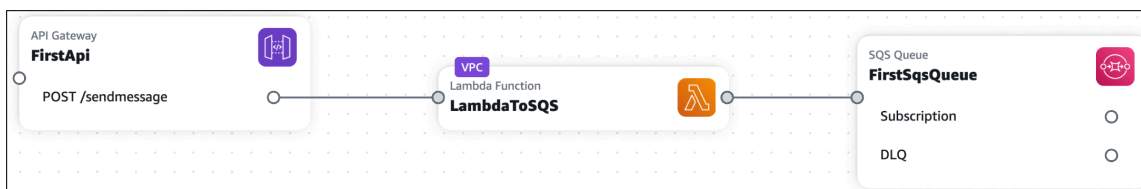


# Integre Infrastructure Composer con Amazon Virtual Private Cloud (Amazon VPC)

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

- Identifique los recursos de su lienzo que se encuentran en una VPC mediante una etiqueta de VPC 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 una VPC.



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

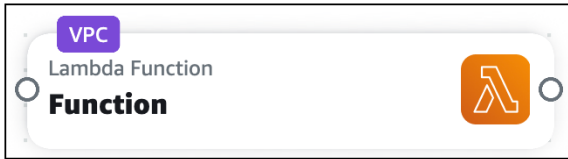
## Temas

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

## Identificar los recursos de Infrastructure Composer y la información relacionada en una VPC

Para integrar Infrastructure Composer con Amazon VPC, primero debe identificar los recursos de una 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 de SSM y los tipos de valores estáticos.

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



Las etiquetas de VPC se aplican a las tarjetas del lienzo cuando haces lo siguiente:

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

## 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 usar para almacenar información de 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%;" type="text" value="PrivateSubnet1"/> <span style="float: right; color: blue; font-size: 1.2em;">✕</span>	Parameter <span style="float: right; color: blue; font-size: 1.2em;">▼</span>

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

Para el tipo SSM, puede proporcionar los siguientes valores:

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

## Referencia dinámica

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

Infrastructure Composer crea el código de infraestructura para configurar la función Lambda con el valor del almacén de parámetros de SSM. 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: 0 5px;">✕</span>	<input style="width: 90%; border: none;" type="text" value="SSM"/> <span style="float: right; border: 1px solid #ccc; border-radius: 50%; padding: 0 5px;">▼</span>

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

```

...
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 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 elGroupId. 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 laSubnetId. 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 style="width: 90%;" type="text" value="MySecurityGroup"/>	<input style="width: 90%;" type="text" value="Parameter"/>	<a href="#" style="border: 1px solid #007bff; border-radius: 10px; padding: 2px 10px; color: #007bff;">Remove</a>
<input style="width: 90%;" type="text" value="sg-0b61d5c742dc2c773"/>	<input style="width: 90%;" type="text" value="Static value"/>	<a href="#" style="border: 1px solid #007bff; border-radius: 10px; padding: 2px 10px; color: #007bff;">Remove</a>
<input style="width: 90%;" type="text" value="{{resolve::ssm::demo/sg-0b61d5c742dc23}}"/>	<input style="width: 90%;" type="text" value="SSM"/>	<a href="#" style="border: 1px solid #007bff; border-radius: 10px; padding: 2px 10px; color: #007bff;">Remove</a>

[Add new item](#)

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 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 una VPC desde una plantilla externa

1. En el panel de propiedades de recursos de la función Lambda, amplíe la sección desplegable de configuración de VPC (avanzada).
2. Seleccione Asignar a una VPC externa.
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 de plantillas importadas para una VPC externa con Infrastructure Composer

Al importar una plantilla existente con parámetros definidos para los grupos de seguridad y las subredes de una VPC externa, 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 VPC externa para una nueva función de 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 Lambda, verá el siguiente mensaje:

**Security group IDs**

List of VPC security group identifiers

Value	Type
<input style="width: 90%;" type="text" value="VPCSecurityGroups"/> <span style="float: right; border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px; color: #007bff; font-weight: bold;">×</span>	<span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px;">Parameter</span> <span style="float: right; color: #007bff; font-weight: bold;">▼</span>
<span style="border: 1px solid #ccc; border-radius: 15px; padding: 5px 15px; background-color: #f0f0f0; display: inline-block;">Add new item</span>	

Only one List<AWS::EC2::SecurityGroup::Id> parameter type can be provided.

Esta limitación se produce porque SecurityGroupIds tanto las SubnetIds propiedades de un AWS::Lambda::Function VpcConfig objeto como las propiedades 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 type="text" value="MySecurityGroup"/>	Parameter
Use: "MySecurityGroup"	
VPCSecurityGroups	

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

```

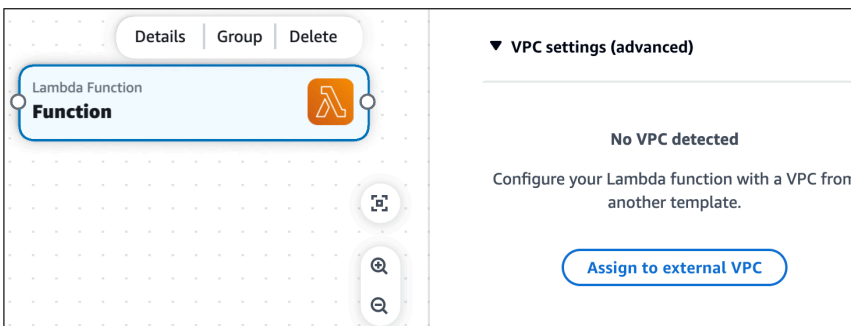
## Configure una función Lambda y una VPC definidas en otra plantilla con Infrastructure Composer

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.



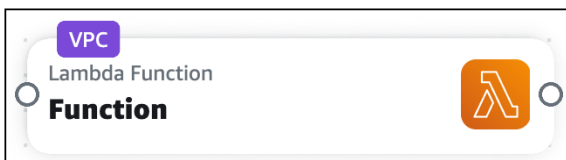
A continuación, abrimos el panel de propiedades de los recursos de la tarjeta y ampliamos la sección desplegable de configuración de VPC (avanzada).



A continuación, seleccionamos Asignar a una VPC externa para empezar a configurar una VPC a 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 define la VPC. Elegimos el tipo de valor estático e ingresamos el valor de nuestros IDs. Seleccionamos Guardar cuando terminamos.

Ahora que nuestra función Lambda está configurada con nuestra VPC, la etiqueta de VPC aparece en nuestra tarjeta.



Infrastructure Composer ha creado el código de infraestructura para configurar nuestra función Lambda con el grupo de seguridad y la subred de la VPC externa.

```

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 desarrollador para crear y ejecutar aplicaciones sin servidor. Con AWS SAM su sintaxis abreviada, los desarrolladores declaran los 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: el AWS SAMCLI AWS SAM proyecto y la plantilla. AWS SAM
- [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 seguir para poder implementar su aplicación en la AWS nube. AWS SAM

Al diseñar la aplicación en Infrastructure Composer, puede usar el `aws sam sync` comando para que detecte AWS SAMCLI automáticamente los cambios locales y los implemente allí CloudFormation. Para obtener más información, consulte [Uso de sam sync](#) en la Guía para AWS Serverless Application Model desarrolladores.

## Siguientes pasos

Consulte para [Prepárese para la implementación con Infrastructure Composer AWS SAMCLI y preparar la implementación de la aplicación.](#)

# Prepárese para la implementación con Infrastructure Composer AWS SAMCLI y

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

## Instala la CLI AWS

Recomendamos instalar y configurar el AWSCLI antes de instalar el AWS SAMCLI. 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 AWSCLI, debe configurar AWS las credenciales. Para obtener más información, consulte [Configuración rápida](#) en la Guía del AWS Command Line Interface usuario.

## Instala la CLI AWS SAM

Para instalarlo AWS SAMCLI, consulte [Instalación del AWS SAMCLI en la Guía para AWS Serverless Application Model desarrolladores](#).

## Acceda al AWS SAMCLI

Si usa Infrastructure Composer desde Consola de administración de AWS, tiene las siguientes opciones para usar el AWS SAMCLI.

### 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 ejecutarlo AWS SAMCLI 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, ejecútelo AWS SAMCLI desde la carpeta principal que contiene la plantilla. También puede utilizar la `--template-file` opción con cualquier AWS SAMCLI comando y proporcionar la ruta a la plantilla.

## Utilice Infrastructure Composer desde 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 el AWS SAMCLI de VS Code.

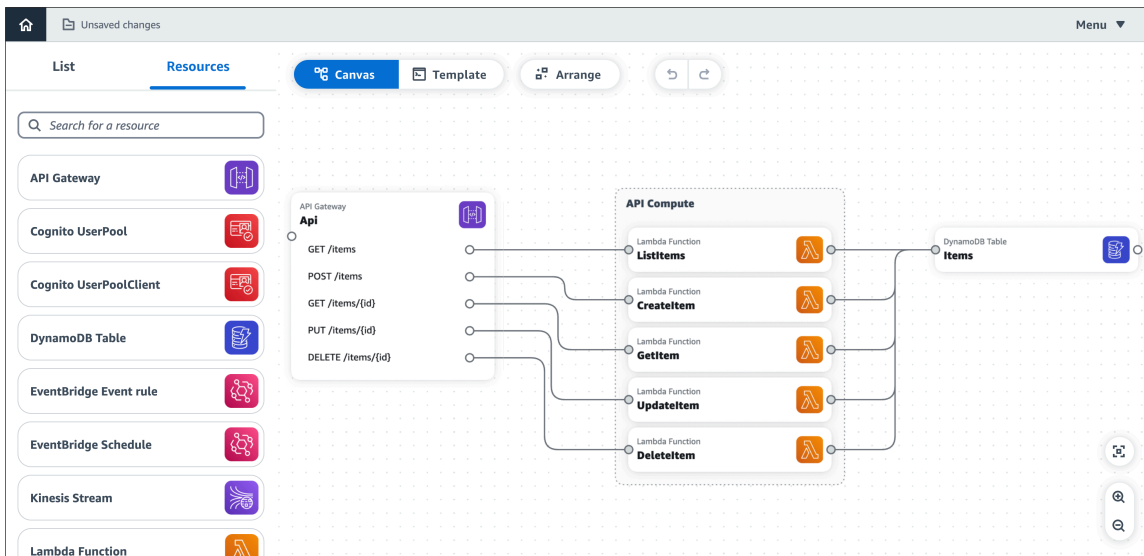
## Siguientes pasos

Para implementar la 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 la implementación con Infrastructure Composer AWS SAMCLI y](#), 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
```

AWS SAMCLI 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]:
ListItem may not have authorization defined, Is this okay? [y/N]: y
CreateItem may not have authorization defined, Is this okay? [y/N]: y
GetItem may not have authorization defined, Is this okay? [y/N]: y
UpdateItem may not have authorization defined, Is this okay? [y/N]: y
DeleteItem may not have authorization defined, Is this okay? [y/N]: y
Save arguments to configuration file [Y/n]:
SAM configuration file [samconfig.toml]:
SAM configuration environment [default]:

```

AWS SAMCLIMuestra un resumen de lo que se implementará:

```

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

```

AWS SAMCLIDespliega la aplicación, primero creando un conjunto de 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           LogicalResourceId
ResourceType         Replacement
-----

```

+ Add	ApiDeploymentcc153d135b
AWS::ApiGateway::Deployment	N/A
+ Add	ApiProdStage
AWS::ApiGateway::Stage	N/A
+ Add	Api
AWS::ApiGateway::RestApi	N/A
+ Add	CreateItemApiPOSTitemsPermissionP
AWS::Lambda::Permission	N/A
	rod
+ Add	CreateItemRole
AWS::IAM::Role	N/A
+ Add	CreateItem
AWS::Lambda::Function	N/A
+ Add	DeleteItemApiDELETEitemsidPermiss
AWS::Lambda::Permission	N/A
	ionProd
+ Add	DeleteItemRole
AWS::IAM::Role	N/A
+ Add	DeleteItem
AWS::Lambda::Function	N/A
+ Add	GetItemApiGETitemsidPermissionPro
AWS::Lambda::Permission	N/A
	d
+ Add	GetItemRole
AWS::IAM::Role	N/A
+ Add	GetItem
AWS::Lambda::Function	N/A
+ Add	Items
AWS::DynamoDB::Table	N/A
+ Add	ListItemsApiGETitemsPermissionPro
AWS::Lambda::Permission	N/A
	d
+ Add	ListItemsRole
AWS::IAM::Role	N/A
+ Add	ListItems
AWS::Lambda::Function	N/A
+ Add	UpdateItemApiPUTitemsidPermission
AWS::Lambda::Permission	N/A
	Prod
+ Add	UpdateItemRole
AWS::IAM::Role	N/A
+ Add	UpdateItem
AWS::Lambda::Function	N/A

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

Luego, AWS SAMCLI 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
DeleteItemRole         Resource creation Initiated
CREATE_IN_PROGRESS     AWS::IAM::Role
ListItemsRole          Resource creation Initiated
CREATE_IN_PROGRESS     AWS::IAM::Role            GetItemRole
                        Resource creation Initiated
CREATE_IN_PROGRESS     AWS::IAM::Role
UpdateItemRole         Resource creation Initiated
CREATE_IN_PROGRESS     AWS::IAM::Role
CreateItemRole         Resource creation Initiated
CREATE_COMPLETE        AWS::IAM::Role
DeleteItemRole         -
CREATE_COMPLETE        AWS::IAM::Role
ListItemsRole          -
```

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

CREATE_IN_PROGRESS	AWS::Lambda::Permission	d
ListItemsApiGETItemsPermissionPro	-	
CREATE_IN_PROGRESS	AWS::Lambda::Permission	d
DeleteItemApiDELETEItemsidPermiss	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	ionProd
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	-	rod

```

CREATE_COMPLETE          AWS::Lambda::Permission
  UpdateItemApiPUTitemsidPermission  -
                                                                    Prod
CREATE_COMPLETE          AWS::Lambda::Permission
  ListItemsApiGETitemsPermissionPro  -
                                                                    d
CREATE_COMPLETE          AWS::Lambda::Permission
  DeleteItemApiDELETEitemsidPermiss  -
                                                                    ionProd
CREATE_COMPLETE          AWS::Lambda::Permission
  GetItemApiGETitemsidPermissionPro  -
                                                                    d
CREATE_COMPLETE          AWS::CloudFormation::Stack
composer-basic-api      -
-----

```

Por último, aparece un mensaje que le informa de que la implementación se realizó 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 CloudFormation pila mediante el `aws-sam delete` comando.

Introduzca el comando `aws-sam delete` en AWS SAMCLI 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.

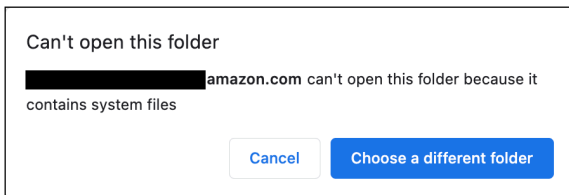
## Temas

- [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.

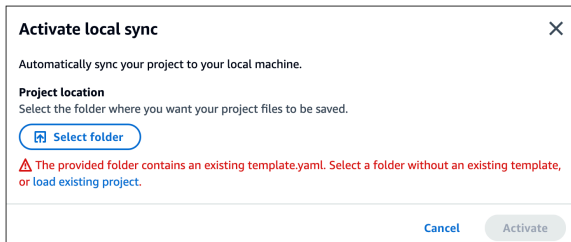
Para obtener información sobre los archivos externos compatibles en Infrastructure Composer, consulte [Haga referencia a archivos externos](#).

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 es 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 comprender 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 las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS .

Con fines de protección de 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:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios 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.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 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 Servicios de AWS mediante la consola, la API o. AWS CLI AWS SDKs Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

**Note**

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, 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 una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de

IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos de Infrastructure Composer. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

## Temas

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

## Público

Infrastructure Composer requiere, como mínimo, acceso de solo lectura al. Consola de administración de AWS 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 autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte [Cómo iniciar sesión en la Cuenta de AWS](#) en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte [AWS Signature Version 4 para solicitudes de API](#) en la Guía del usuario de IAM.

## Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Se recomienda encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de 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 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 empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

## Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte [Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte [Casos de uso para usuarios de IAM](#) en la Guía del usuario de IAM.

## Roles de IAM

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

## Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa

estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte [Información general de políticas de JSON](#) en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

## Políticas basadas en identidades

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

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte [Selección entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

## Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe [especificar una entidad principal](#) en una política basada en recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política 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 políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF 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 que pueden establecer los permisos máximos otorgados por los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte [Políticas de control de servicios](#) en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte [Políticas de control de recursos \(RCPs\)](#) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. 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 usuario de IAM.

## ¿Cómo AWS Infrastructure Composer funciona con IAM

AWS Infrastructure Composer requiere, como mínimo, acceso de solo lectura al. Consola de administración de AWS 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 las funciones de IAM con las que se pueden utilizar. AWS Infrastructure Composer

Característica de IAM	Compatibilidad con 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 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 las funciones de IAM, consulte [AWS los servicios que funcionan con IAM](#) en la Guía del usuario de IAM.

## Políticas basadas en la identidad para Infrastructure Composer

Admite políticas basadas en la identidad: no

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, 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 más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de la política de JSON de IAM](#) en la Guía del usuario de IAM.

## Políticas basadas en recursos dentro de Infrastructure Composer

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores 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 basada en recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

## Acciones políticas para Infrastructure Composer

Apoya las acciones políticas: No

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

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. 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 las políticas de AWS JSON 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` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). En el caso de las acciones que no admiten permisos por recurso, utilice 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 tipos de recursos ARNs, consulte [los recursos definidos por AWS Infrastructure Composer](#) en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte [Acciones definidas por AWS Infrastructure Composer](#).

## Claves de condición de política para Infrastructure Composer

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

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

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede 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. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del usuario de IAM.

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 políticas JSON.

## ABAC con Infrastructure Composer

Compatibilidad con ABAC (etiquetas en las políticas): no

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos denominados etiquetas. Puede adjuntar etiquetas a las entidades y AWS los recursos de IAM y, a continuación, diseñar políticas de ABAC para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso.

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 sobre ABAC, consulte [Definición de permisos con la autorización de ABAC](#) en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte [Uso del control de acceso basado en atributos \(ABAC\)](#) en la Guía del usuario de IAM.

## Uso de credenciales temporales con Infrastructure Composer

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a AWS los recursos a corto plazo y se crean automáticamente cuando se utiliza la federación o se cambia de rol. 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](#) y [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

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

## Permisos principales multiservicio para Infrastructure Composer

Compatibilidad con sesiones de acceso directo (FAS): no

Las sesiones de acceso directo (FAS) utilizan los permisos de la persona principal que realiza la llamada Servicio de AWS, además de los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Sesiones de acceso directo](#).

## 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 a un Servicio de AWS](#) en la Guía del usuario de IAM.

### 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 más información sobre cómo crear o administrar roles vinculados a servicios, consulta [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. Para obtener más información sobre su responsabilidad de conformidad al utilizarlos Servicios de AWS, consulte [AWS la documentación de seguridad](#).

## 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 obtener notificaciones sobre las actualizaciones de esta documentación, puedes suscribirte a una fuente RSS.

- 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 para usar 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 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 ha añadido documentación para implementar la aplicación con Infrastructure Composer desde el 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 documentación sobre Infrastructure Composer desde la AWS Toolkit for Visual Studio Code</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> .	30 de noviembre de 2023
<a href="#">Se agregó la integración de Step Functions con Workflow Studio</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> .	27 de noviembre de 2023
<a href="#">Se agregó la integración de la consola Lambda y de Infrastructure Composer</a>	Inicie el lienzo de Infrastructure Composer desde la consola Lambda. Para obtener más información, consulte <a href="#">Utilización AWS Infrastructure Composer con la AWS Lambda consola</a> .	14 de noviembre de 2023

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

Infrastructure Composer presenta una etiqueta de VPC para visualizar los recursos configurados con una VPC. 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 Infrastructure Composer a un clúster o instancia de base de datos de Amazon RDS 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 recursos CloudFormation](#)

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

26 de septiembre de 2023

<a href="#">Se agregó documentación para tarjetas en Infrastructure Composer</a>	Infrastructure Composer admite varios tipos de tarjetas que puede usar para diseñar y crear su aplicación. Para obtener más información, consulte <a href="#">Diseñar con tarjetas en Infrastructure Composer</a> .	20 de septiembre de 2023
<a href="#">Se agregó documentación para la función de deshacer y rehacer</a>	Utilice los botones de deshacer y rehacer del lienzo de Infrastructure Composer. Para obtener más información, consulte <a href="#">Deshacer y rehacer</a> .	1 de agosto de 2023
<a href="#">Se agregó documentación para el modo de sincronización local</a>	Usa el modo de sincronización local para sincronizar y guardar automáticamente el proyecto en tu máquina local. Para obtener más información, consulta <a href="#">Modo de sincronización local</a> .	1 de agosto de 2023
<a href="#">Se agregó documentación para la función de exportación de lienzo</a>	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 <a href="#">Exportar lienzo</a> .	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

<a href="#">Infrastructure Composer ahora está disponible de forma general</a>	AWS Infrastructure Composer ahora está disponible para el público en general. Para obtener más información, consulte <a href="#">AWS Infrastructure Composer ahora disponible para todos: Cree aplicaciones sin servidor de forma visual y rápida.</a>	7 de marzo de 2023
<a href="#">Ampliamos las ventajas del uso del modo conectado</a>	Utilice Infrastructure Composer en modo conectado con su IDE local para acelerar el desarrollo. Para obtener más información, consulte <a href="#">Uso de Infrastructure Composer con su IDE local.</a>	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 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

[Versión pública](#)

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.