



Guida per gli sviluppatori

# AWS Infrastructure Composer



# AWS Infrastructure Composer: Guida per gli sviluppatori

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

---

# Table of Contents

Che cos'è Infrastructure Composer? .....	1
Componi la tua architettura .....	2
Definisci i tuoi modelli .....	4
Integrati con i tuoi flussi di lavoro .....	5
Modi per accedere a Infrastructure Composer .....	6
Ulteriori informazioni .....	8
Fasi successive .....	8
Concetti serverless .....	8
Concetti serverless .....	9
Schede .....	10
Schede componenti avanzate .....	11
Esempio .....	12
Schede componenti standard .....	13
Connessioni tramite scheda .....	17
Connessioni tra schede .....	17
Connessioni tra schede componenti avanzate .....	18
Connessioni da e verso schede di risorse IAc standard .....	20
Nozioni di base .....	21
Fai un tour della console .....	21
Fasi successive .....	22
Carica e modifica .....	22
Fase 1: Apri la demo .....	23
Fase 2: Esplora la tela visiva .....	23
Fase 3: Espandi l'architettura .....	27
Fase 4: Salvare l'applicazione .....	28
Fasi successive .....	29
Creazione .....	29
Proprietà delle risorse .....	30
Fase 1: Crea il tuo progetto .....	30
Aggiungi carte .....	33
Fase 3: Configura la tua API REST .....	34
Fase 4: Configura le tue funzioni .....	35
Fase 5: Connect le tue carte .....	36
Fase 6: Organizza la tela .....	37

Aggiungere una tabella DynamoDB .....	38
Fase 8: Rivedi il modello .....	39
Fase 9: Integrazione nei flussi di lavoro .....	40
Fasi successive .....	40
Dove usare Infrastructure Composer .....	41
Console Infrastructure Composer .....	41
panoramica visiva .....	42
Gestisci il tuo progetto .....	45
Connect al tuo IDE locale .....	48
Consenti l'accesso alle pagine Web .....	51
Sincronizzazione e salvataggio locali .....	52
Importazione dalla console Lambda .....	56
Esporta tela .....	56
CloudFormation modalità console .....	58
Perché usare questa modalità? .....	58
Accedere a questa modalità .....	59
Visualizza una distribuzione .....	59
Crea un nuovo modello .....	60
Aggiorna uno stack esistente .....	61
AWS Toolkit for Visual Studio Code .....	63
Panoramica visiva .....	63
Accesso da VS Code .....	65
Sincronizzazione con Cloud AWS .....	66
Infrastructure Composer con Amazon Q .....	68
Come comporre .....	71
Posiziona le carte sulla tela .....	71
Raggruppa le carte .....	72
Raggruppamento di schede componenti avanzate .....	72
Raggruppamento di una scheda componente standard in un'altra .....	73
Schede Connect .....	75
Collegamento di schede componenti avanzate .....	75
Collegamento di schede standard .....	76
Esempi .....	78
Disconnetti le schede .....	80
Schede componenti avanzate .....	80
Schede componenti standard .....	80

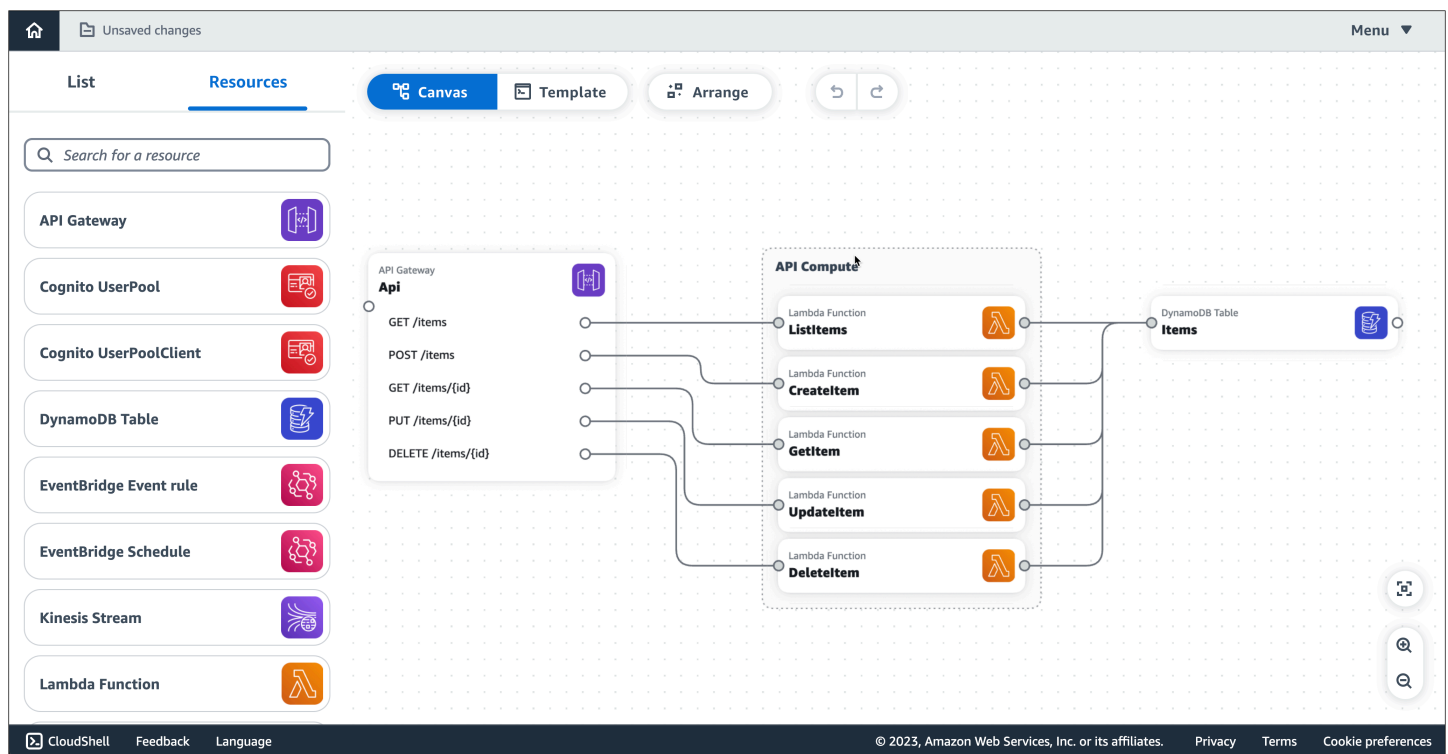
Disporre le carte .....	82
Configura e modifica le carte .....	83
Carte avanzate .....	84
Schede standard .....	100
Eliminare le carte .....	101
Schede componenti avanzate .....	101
Schede componenti standard .....	102
Visualizza gli aggiornamenti del codice .....	102
Vantaggi del Change Inspector .....	103
Procedura .....	103
Ulteriori informazioni .....	105
File esterni di riferimento .....	106
Best practice .....	107
Crea un riferimento a un file esterno .....	107
Carica un progetto .....	108
Create un'applicazione utilizzando il AWS SAMCLI .....	109
Fai riferimento a una specifica OpenAPI .....	113
Integrazione con Amazon VPC .....	116
Identifica risorse e informazioni .....	117
Configura le funzioni .....	123
Parametri nei modelli importati .....	123
Aggiungere nuovi parametri ai modelli importati .....	126
Configurare una funzione Lambda con un VPC in un altro modello .....	127
Implementa nel cloud AWS .....	130
Concetti importanti AWS SAM .....	130
Fasi successive .....	130
Configura il AWS SAMCLI .....	131
Installa il AWSCLI .....	131
Installa il AWS SAMCLI .....	131
Accedi a AWS SAMCLI .....	131
Fasi successive .....	132
Crea e distribuisce .....	132
Eliminazione di uno stack .....	140
Risoluzione dei problemi .....	142
Messaggi di errore .....	142
«Impossibile aprire questa cartella» .....	142

«Modello incompatibile» .....	142
«La cartella fornita contiene un template.yaml esistente» .....	143
«Il tuo browser non dispone delle autorizzazioni per salvare il progetto in quella cartella...» .	144
Sicurezza .....	145
Protezione dei dati .....	145
Crittografia dei dati .....	147
Crittografia dei dati in transito .....	147
Gestione delle chiavi .....	147
Riservatezza del traffico inter-rete .....	147
AWS Identity and Access Management .....	147
Destinatari .....	148
Autenticazione con identità .....	148
Gestione dell'accesso tramite policy .....	149
Come AWS Infrastructure Composer funziona con IAM .....	151
Convalida della conformità .....	157
Resilienza .....	157
Cronologia dei documenti .....	158
.....	clxiv

# Che cos'è AWS Infrastructure Composer?

AWS Infrastructure Composer consente di comporre visivamente applicazioni moderne su AWS. Più specificamente, puoi utilizzare Infrastructure Composer per visualizzare, creare e distribuire applicazioni moderne da tutti i AWS servizi supportati da AWS CloudFormation senza bisogno di essere un esperto in CloudFormation.

Durante la composizione AWS CloudFormation dell'infrastruttura, tramite un' drag-and-drop interfaccia intuitiva, Infrastructure Composer crea modelli di infrastruttura come codice (IaC), il tutto seguendo le migliori pratiche. AWS L'immagine seguente mostra quanto sia facile trascinare, rilasciare, configurare e connettere le risorse sulla tela visiva di Infrastructure Composer.



Infrastructure Composer può essere utilizzato dalla console Infrastructure Composer e in modalità console. AWS Toolkit for Visual Studio Code CloudFormation

## Argomenti

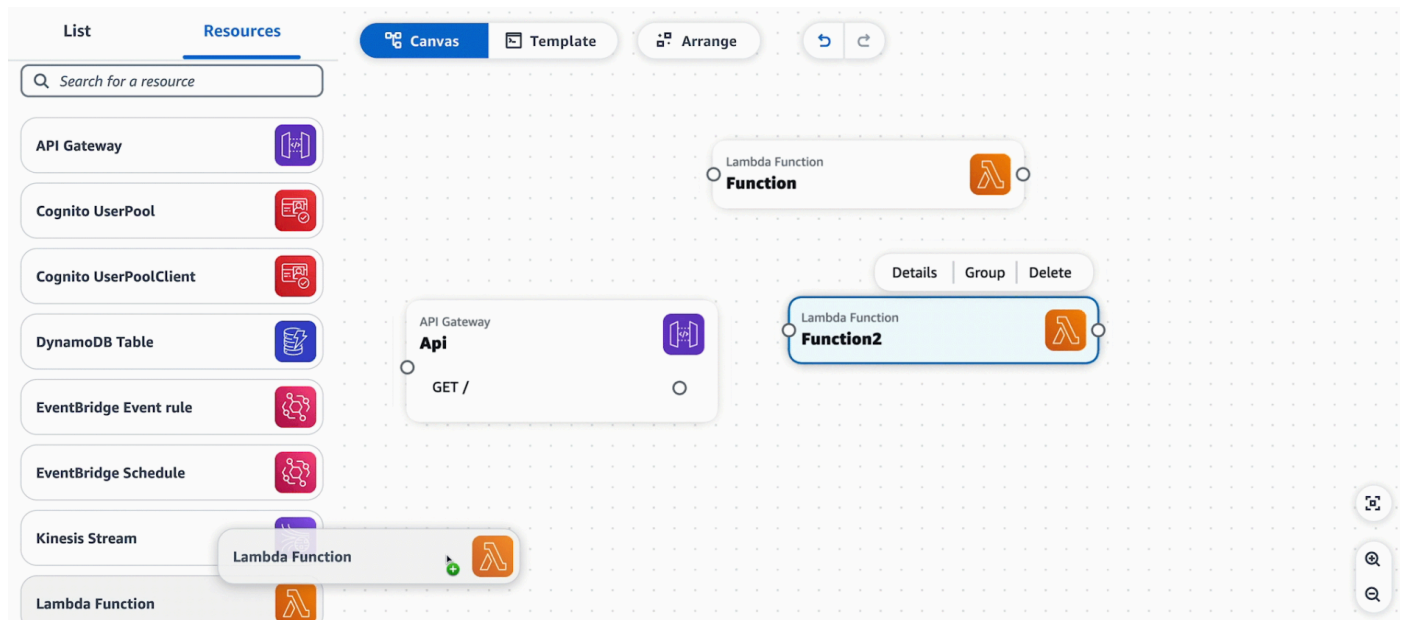
- [Componi l'architettura della tua applicazione](#)
- [Definisci i modelli di infrastruttura come codice \(IaC\)](#)
- [Esegui l'integrazione con i flussi di lavoro esistenti](#)
- [Modi per accedere a Infrastructure Composer](#)

- [Ulteriori informazioni](#)
- [Fasi successive](#)
- [Concetti serverless per AWS Infrastructure Composer](#)

## Componi l'architettura della tua applicazione

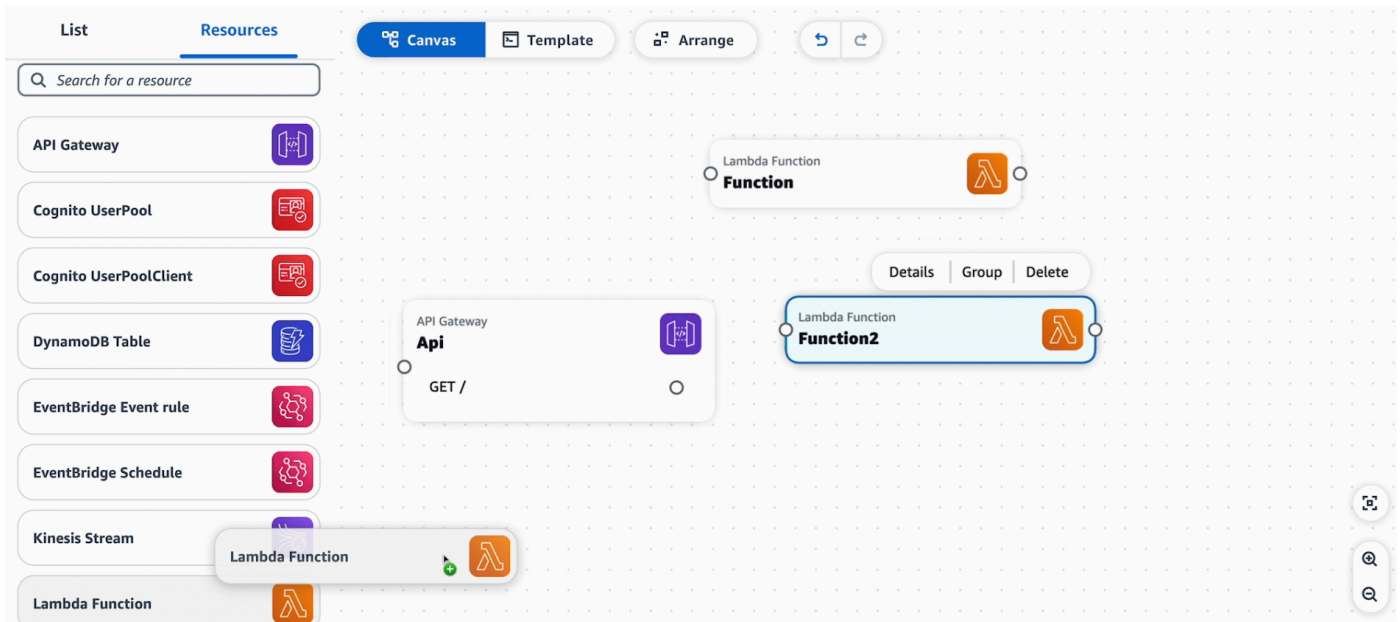
Costruisci con le carte

Posiziona le schede sulla tela di Infrastructure Composer per visualizzare e creare l'architettura della tua applicazione.



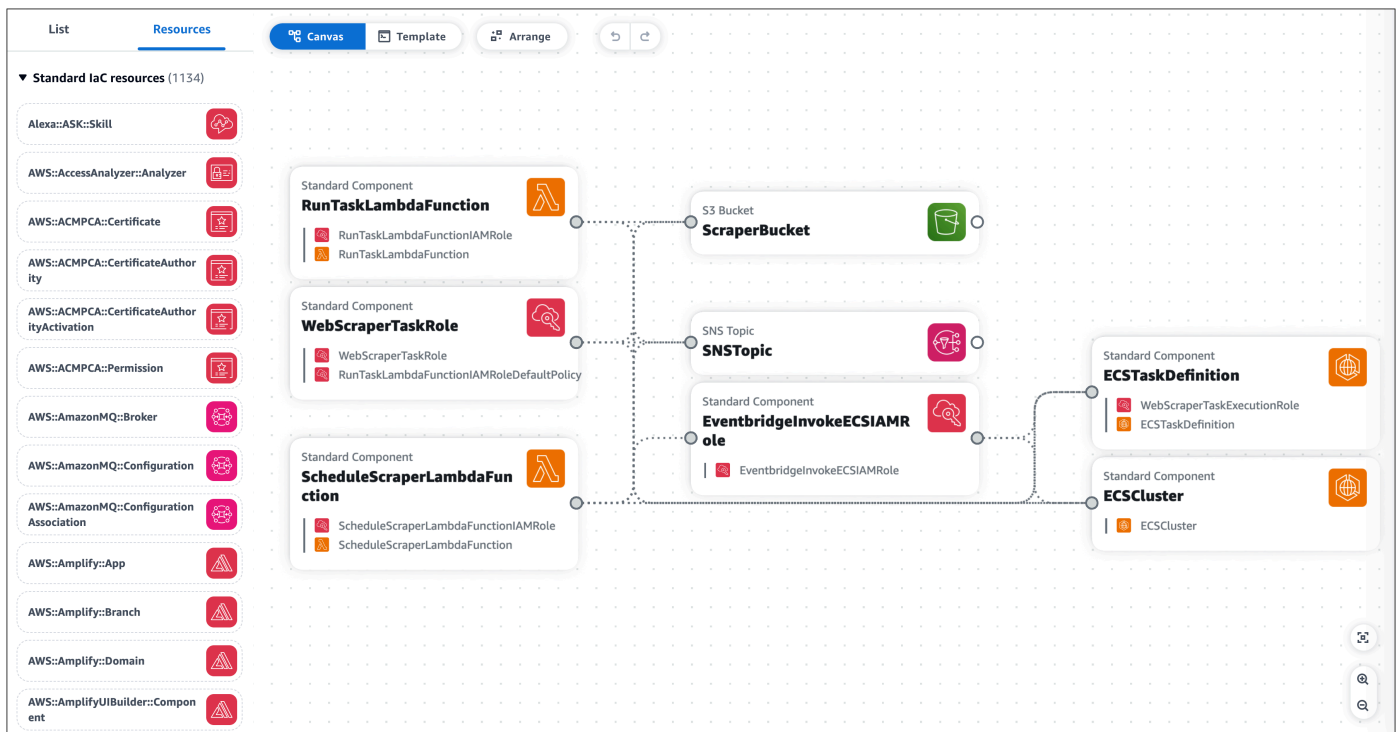
Connect le schede

Configura il modo in cui le tue risorse interagiscono tra loro collegandole visivamente. Specificate ulteriormente le loro proprietà tramite un pannello delle proprietà curato.



### Lavora con qualsiasi AWS CloudFormation risorsa

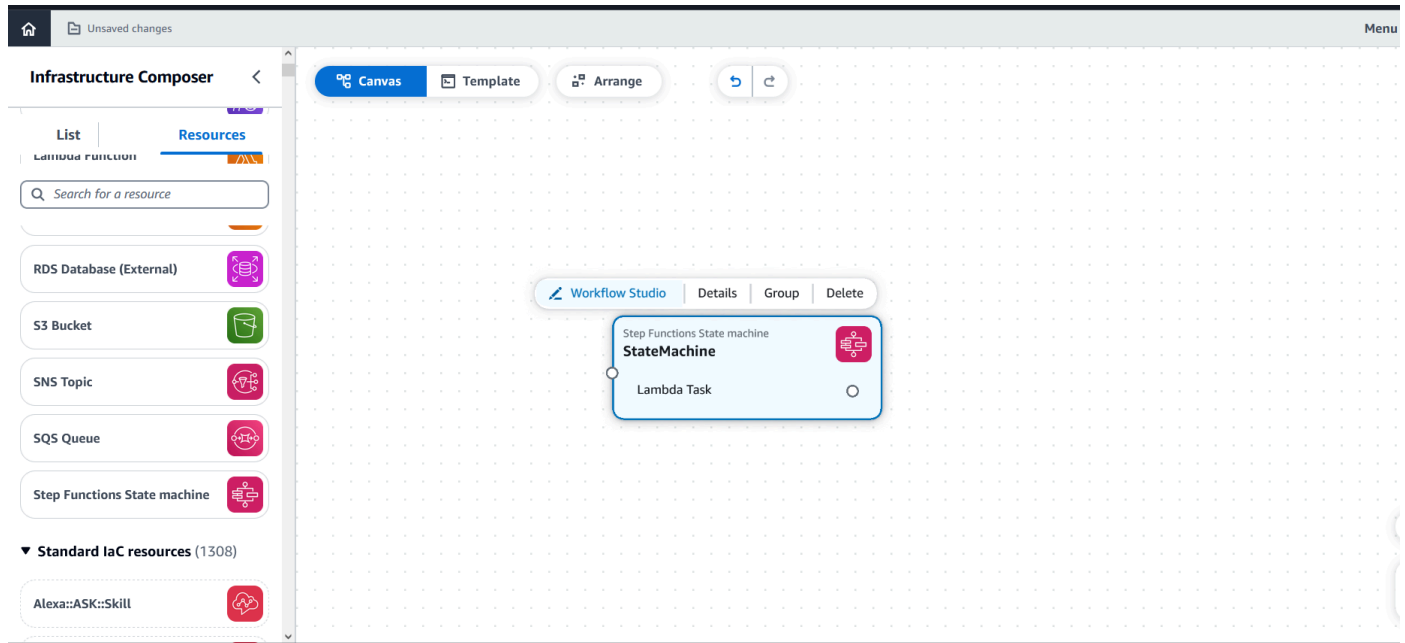
Trascina qualsiasi CloudFormation risorsa sull'area di lavoro per comporre l'architettura dell'applicazione. Infrastructure Composer fornisce un modello IAc iniziale che è possibile utilizzare per specificare le proprietà della risorsa. Per ulteriori informazioni, consulta [Configurazione e modifica delle schede in Infrastructure Composer](#).



## Accedi a funzionalità aggiuntive con funzionalità Servizi AWS

Funzionalità di Infrastructure Composer Servizi AWS che vengono comunemente utilizzate o configurate insieme durante la creazione di applicazioni. Per ulteriori informazioni, consulta [Integrazione con Amazon VPC](#).

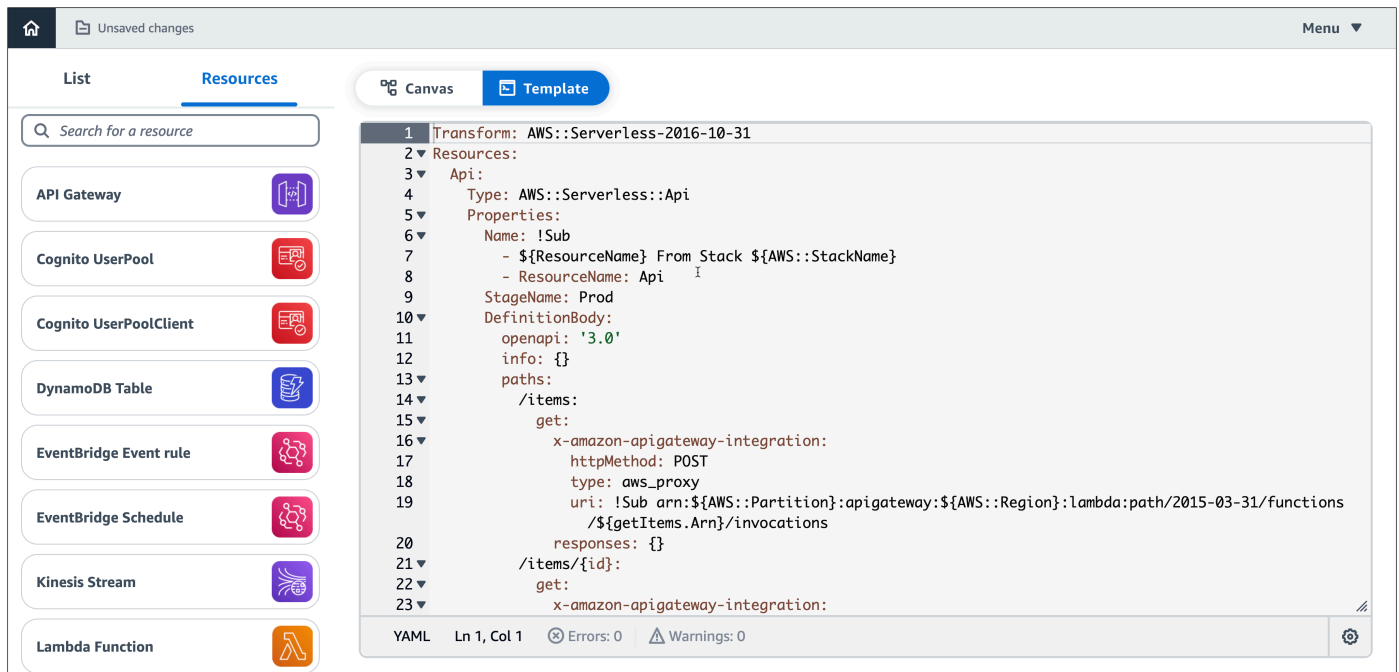
Di seguito è riportato un esempio della AWS Step Functions funzionalità, che fornisce un'integrazione per avviare Step Functions Workflow Studio direttamente all'interno del canvas di Infrastructure Composer.



## Definisci i modelli di infrastruttura come codice (IaC)

Infrastructure Composer crea il codice dell'infrastruttura

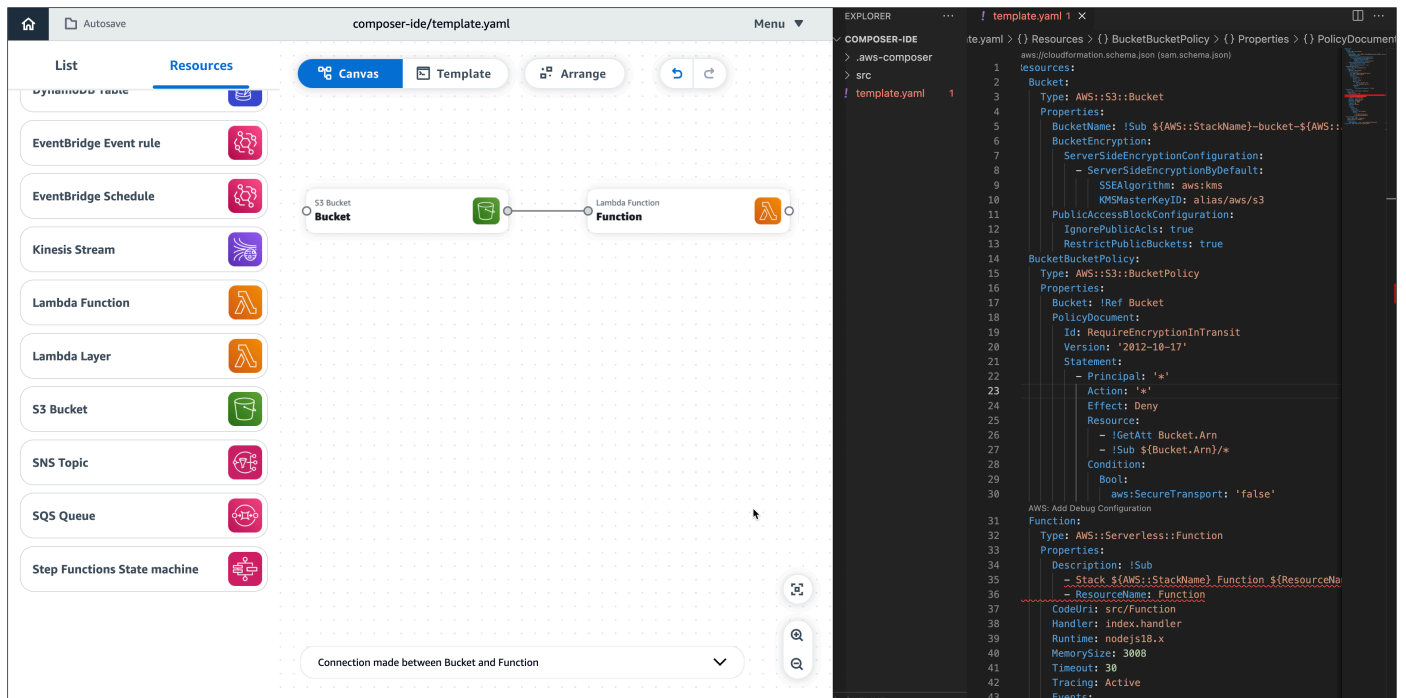
Durante la composizione, Infrastructure Composer crea automaticamente i tuoi modelli AWS CloudFormation and AWS Serverless Application Model (AWS SAM), seguendo le AWS migliori pratiche. È possibile visualizzare e modificare i modelli direttamente da Infrastructure Composer. Infrastructure Composer sincronizza automaticamente le modifiche tra il visual canvas e il codice del modello.



## Esegui l'integrazione con i flussi di lavoro esistenti

### Importa modelli e progetti esistenti

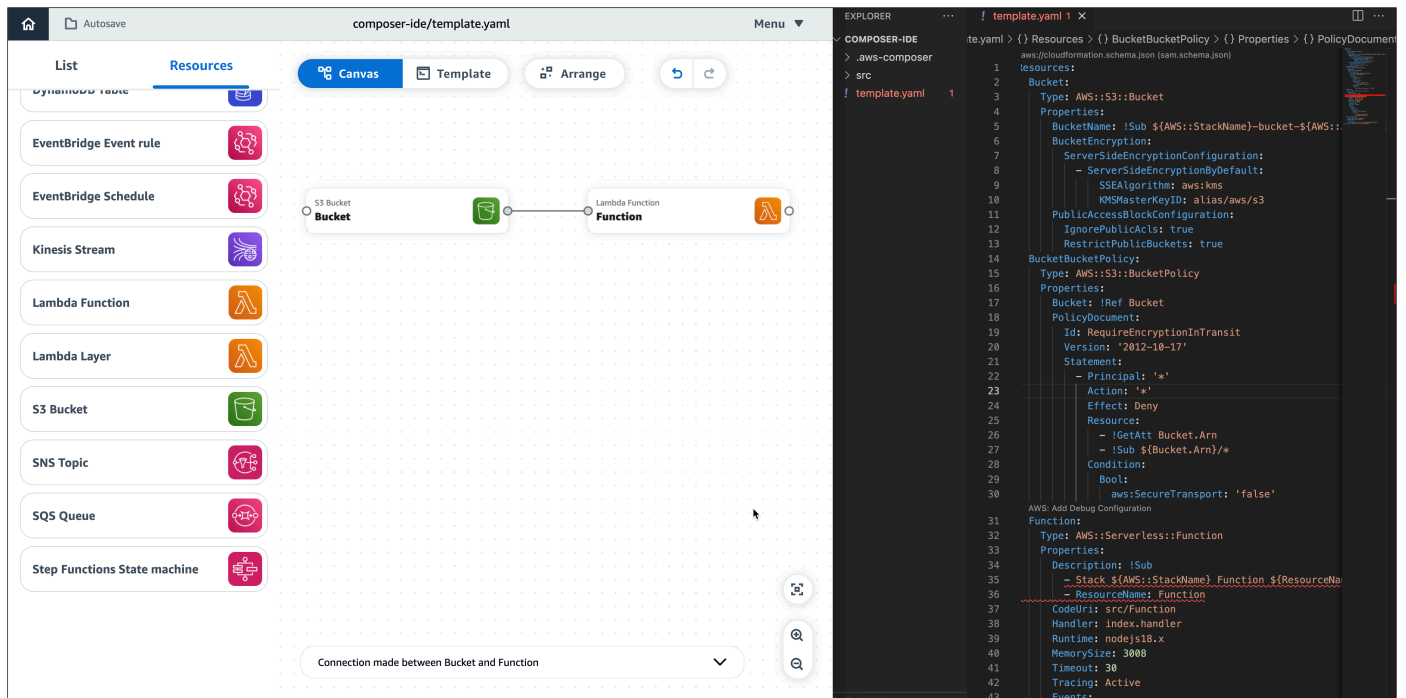
Importa modelli esistenti CloudFormation e AWS SAM modelli per visualizzarli per comprenderli meglio e modificarne il design. Esporta i modelli che crei in Infrastructure Composer e integrali nei flussi di lavoro esistenti per la distribuzione.



## Modi per accedere a Infrastructure Composer

### Dalla console Infrastructure Composer

Accedi a Infrastructure Composer tramite la console Infrastructure Composer per iniziare rapidamente. Inoltre, puoi utilizzare la modalità di sincronizzazione locale per sincronizzare e salvare automaticamente Infrastructure Composer con il tuo computer locale.



## Dalla console CloudFormation

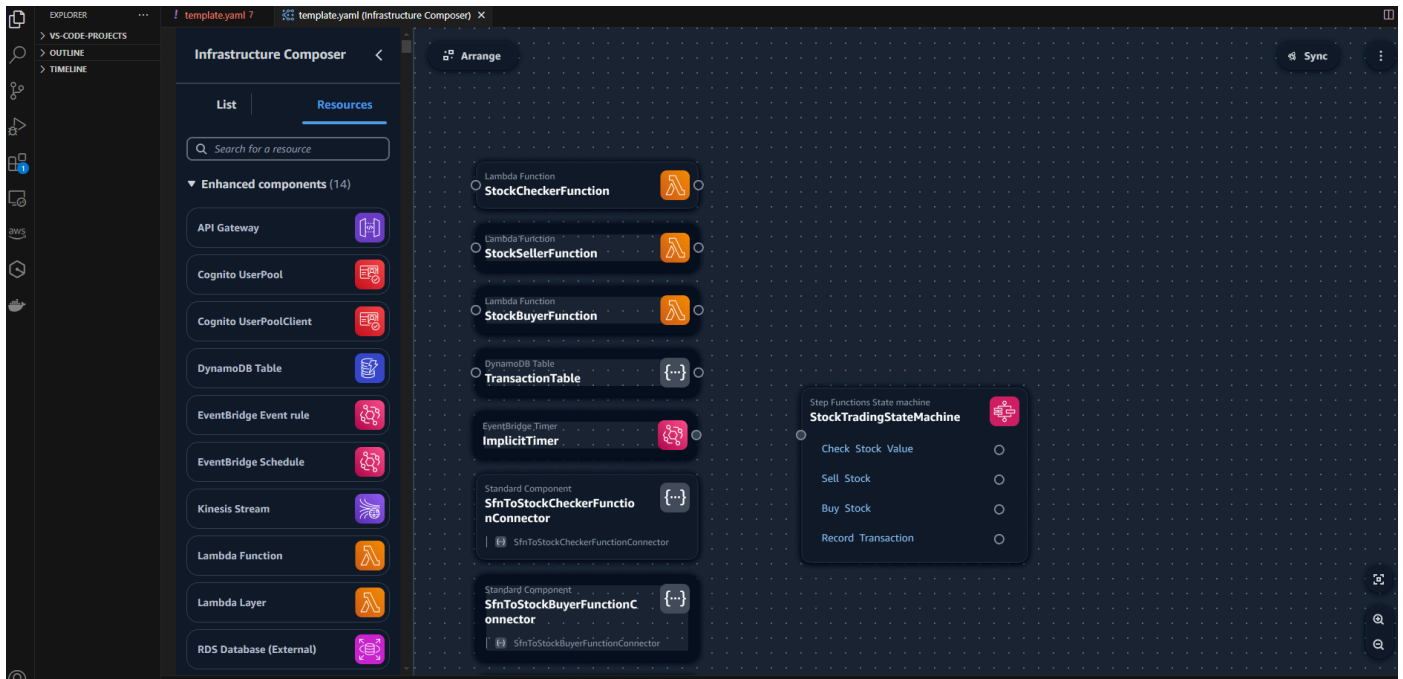
La console Infrastructure Composer supporta anche la [modalità CloudFormation console](#), un miglioramento rispetto a CloudFormation Designer che è integrato con il flusso di lavoro dello CloudFormation stack. Questo nuovo strumento è ora lo strumento consigliato per visualizzare i modelli. CloudFormation

## Dalla console Lambda

Con Infrastructure Composer, puoi anche importare funzioni Lambda dalla console Lambda. Per ulteriori informazioni, consulta [Importa le funzioni in Infrastructure Composer dalla console Lambda](#).

## Dalla AWS Toolkit for Visual Studio Code

Accedi a Infrastructure Composer tramite l'estensione Toolkit for VS Code per portare Infrastructure Composer nel tuo ambiente di sviluppo locale.



## Ulteriori informazioni

Per continuare a conoscere Infrastructure Composer, consulta le seguenti risorse:

- [Schede Infrastructure Composer](#)
- [Componi e crea visivamente applicazioni serverless | Orari di ufficio senza server: panoramica e demo di Infrastructure Composer.](#)

## Fasi successive

Per configurare Infrastructure Composer, vedere. [Guida introduttiva alla console Infrastructure Composer](#)

## Concetti serverless per AWS Infrastructure Composer

Scopri i concetti base del serverless prima di utilizzarli. AWS Infrastructure Composer

# Concetti serverless

## Architettura basata su eventi

Un'applicazione serverless è costituita da singoli AWS servizi, ad esempio AWS Lambda per l'elaborazione e Amazon DynamoDB per la gestione dei database, che svolgono ciascuno un ruolo specializzato. Questi servizi sono quindi liberamente integrati tra loro tramite un'architettura basata sugli eventi. Per ulteriori informazioni sull'architettura basata sugli eventi, vedi [Cos'è un'architettura basata sugli eventi?](#) .

## Infrastruttura come codice (IaC)

Infrastructure as Code (IaC) è un modo di trattare l'infrastruttura nello stesso modo in cui gli sviluppatori trattano il codice, applicando lo stesso rigore dello sviluppo del codice applicativo al provisioning dell'infrastruttura. Definisci la tua infrastruttura in un file modello, la distribuisce e AWS crea le AWS risorse per te. Con IAC, definisci in codice ciò che desideri AWS fornire. Per ulteriori informazioni, consulta [Infrastructure as Code](#) nell'introduzione a DevOps on AWS AWS Whitepaper.

## Tecnologie serverless

Con le tecnologie AWS serverless, puoi creare ed eseguire applicazioni senza dover gestire i tuoi server. Tutta la gestione dei server viene effettuata da AWS noi e offre numerosi vantaggi, come la scalabilità automatica e l'elevata disponibilità integrata, che consentono di portare rapidamente la propria idea alla produzione. Utilizzando le tecnologie serverless, puoi concentrarti sul core del tuo prodotto senza doverti preoccupare della gestione e del funzionamento dei server. Per ulteriori informazioni sul serverless, consulta [Serverless](#) su. AWS

Per un'introduzione di base ai principali servizi AWS serverless, vedi Serverless [101: Understanding the serverless services at Serverless Land](#).

# Schede Infrastructure Composer

Infrastructure Composer semplifica il processo di scrittura dell'infrastruttura come codice (IaC) per le risorse. CloudFormation [Per utilizzare in modo efficace Infrastructure Composer, è necessario innanzitutto comprendere due concetti di base: schede Infrastructure Composer e connessioni tra schede.](#)

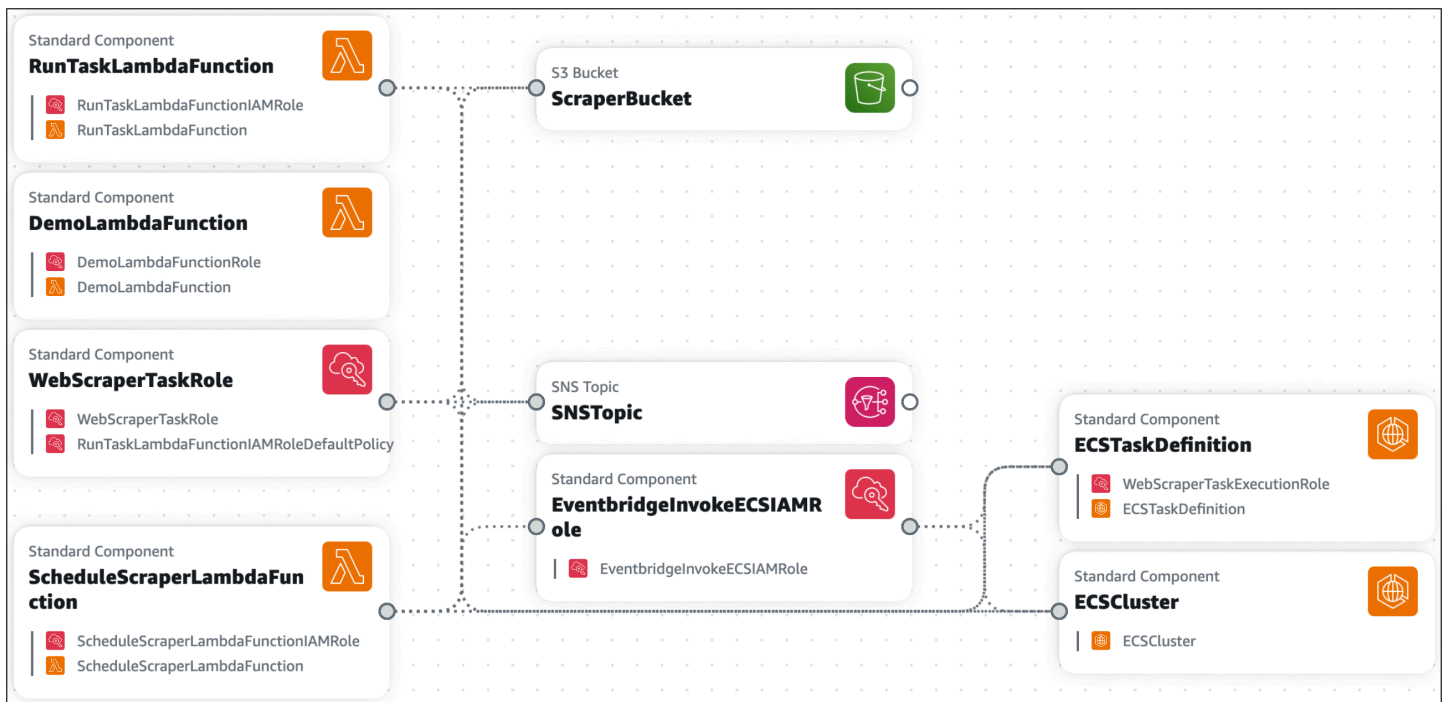
In Infrastructure Composer, le schede rappresentano CloudFormation risorse. Esistono due categorie generali di schede:

- [Scheda componente avanzata](#): una raccolta di CloudFormation risorse che sono state combinate in un'unica scheda curata che migliora la facilità d'uso e la funzionalità e sono progettate per un'ampia varietà di casi d'uso. Le schede componente avanzate sono le prime schede elencate nella palette Risorse di Infrastructure Composer.
- [Scheda di risorse IAC standard: una singola AWS CloudFormation risorsa](#). Ogni scheda di risorse IaC standard, una volta trascinata sulla tela, è etichettata come componente standard e può essere combinata in più risorse.

## Note

A seconda della scheda, una scheda di risorse IAc standard può essere etichettata come scheda componente standard dopo essere stata trascinata sull'area di disegno. Ciò significa semplicemente che la scheda è una raccolta di una o più schede di risorse IAc standard.

Sebbene alcuni tipi di schede siano disponibili nella palette Risorse, le schede possono anche apparire nell'area di disegno quando si importa un modello esistente CloudFormation o AWS Serverless Application Model (AWS SAM) in Infrastructure Composer. L'immagine seguente è un esempio di un'applicazione importata che contiene vari tipi di schede:



## Argomenti

- [Schede componenti avanzate in Infrastructure Composer](#)
- [Schede componenti standard in Infrastructure Composer](#)
- [Connessioni a schede in Infrastructure Composer](#)

## Schede componenti avanzate in Infrastructure Composer

Le schede componenti avanzate vengono create e gestite da Infrastructure Composer. Ogni scheda contiene CloudFormation risorse che vengono comunemente utilizzate insieme per la creazione di applicazioni. AWS Il loro codice di infrastruttura viene creato da Infrastructure Composer seguendo le AWS migliori pratiche. Le schede componenti avanzate sono un ottimo modo per iniziare a progettare un'applicazione.

Le schede dei componenti avanzati sono disponibili nella palette Risorse, nella sezione Componenti avanzati.

Le schede componenti avanzate possono essere completamente configurate e utilizzate all'interno di Infrastructure Composer per progettare e creare applicazioni serverless. Si consiglia di utilizzare schede componenti avanzate durante la progettazione di applicazioni senza codice esistente.

Questa tabella mostra i nostri componenti avanzati con collegamenti alla specifica del modello AWS CloudFormation or AWS Serverless Application Model (AWS SAM) della risorsa in evidenza della scheda:

Card	Documentazione di riferimento
Gateway Amazon API	<a href="#">AWS::Serverless::API</a>
Amazon Cognito UserPool	<a href="#">AWS::Cognito::UserPool</a>
Amazon Cognito UserPoolClient	<a href="#">AWS::Cognito::UserPoolClient</a>
Tabella Amazon DynamoDB	<a href="#">AWS::DynamoDB::Table</a>
Regola Amazon EventBridge Event	<a href="#">AWS::Events::Rule</a>
EventBridge Pianificazione	<a href="#">AWS::Scheduler::Schedule</a>
Amazon Kinesis Stream	<a href="#">AWS::Kinesis::Stream</a>
AWS Lambda Funzione	<a href="#">AWS::Serverless::Function</a>
Strato Lambda	<a href="#">AWS::Serverless::LayerVersion</a>
Bucket Amazon Simple Storage Service (Amazon S3)	<a href="#">AWS::S3::Bucket</a>
Argomento Amazon Simple Notification Service (Amazon SNS)	<a href="#">AWS::SNS::Topic</a>
Coda Amazon Simple Queue Service (Amazon SQS)	<a href="#">AWS::SQS::Queue</a>
AWS Step Functions Macchina a stati	<a href="#">AWS::Serverless::StateMachine</a>

## Esempio

Di seguito è riportato un esempio di componente avanzato per S3 Bucket:



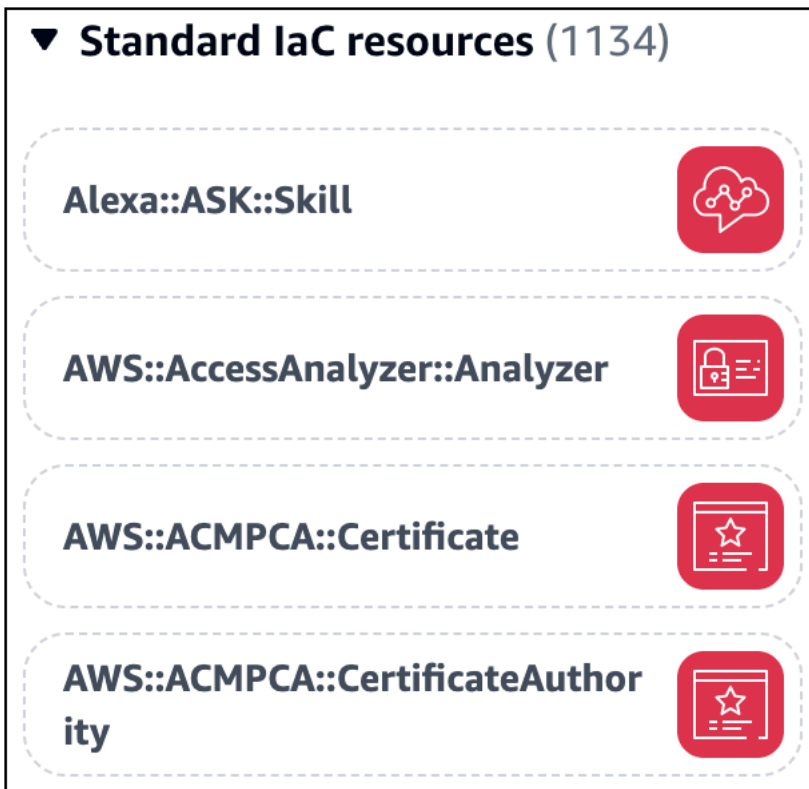
Quando trascini una scheda componente S3 Bucket sull'area di disegno e visualizzi il modello, al modello verranno aggiunte le due CloudFormation risorse seguenti:

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

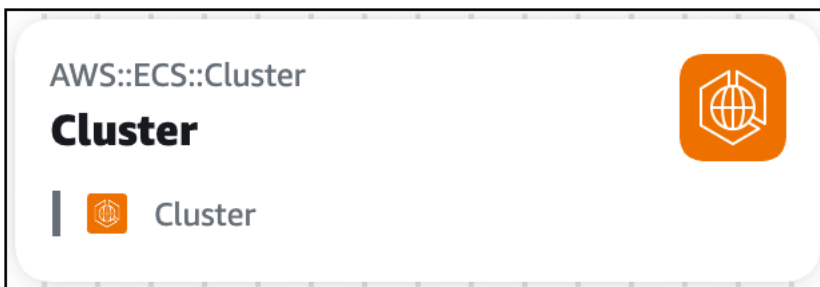
La scheda componente avanzata S3 Bucket rappresenta due CloudFormation risorse che sono entrambe necessarie per consentire a un bucket Amazon Simple Storage Service (Amazon S3) di interagire con altri servizi dell'applicazione.

## Schede componenti standard in Infrastructure Composer

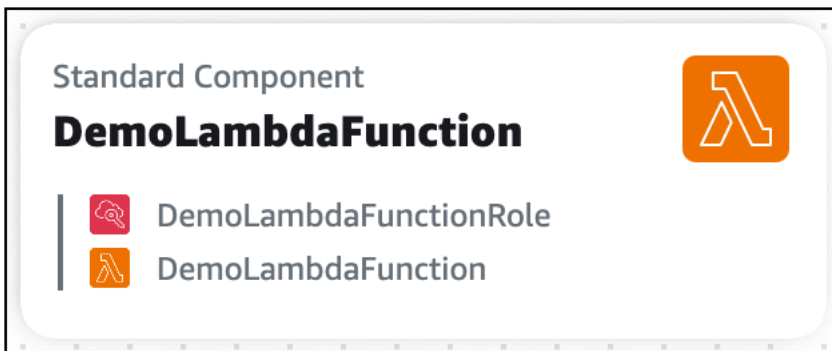
Prima che una scheda componente standard venga posizionata nell'area di disegno visiva di Infrastructure Composer, viene elencata come scheda di risorse Standard (IaC) nella palette Risorse di Infrastructure Composer. Una scheda di risorse standard (IaC) rappresenta una singola risorsa CloudFormation. Ogni scheda di risorse IaC standard, una volta posizionata sull'area di disegno, diventa una scheda denominata Componente standard e può essere combinata per rappresentare più risorse CloudFormation.



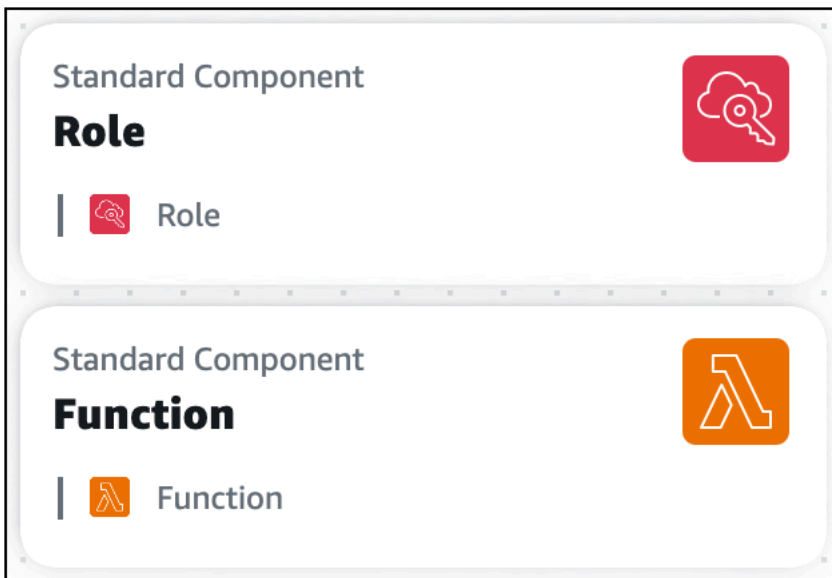
Ogni scheda di risorse IaC standard può essere identificata in base al tipo di CloudFormation risorsa. Di seguito è riportato un esempio di scheda di risorse IaC standard che rappresenta un tipo di `AWS::ECS::Cluster` CloudFormation risorsa:



Ogni scheda componente standard visualizza le CloudFormation risorse che contiene. Di seguito è riportato un esempio di scheda componente standard che include due risorse IaC standard:



Durante la configurazione delle proprietà delle schede componenti standard, Infrastructure Composer può combinare le schede correlate. Ad esempio, ecco due schede componenti standard:



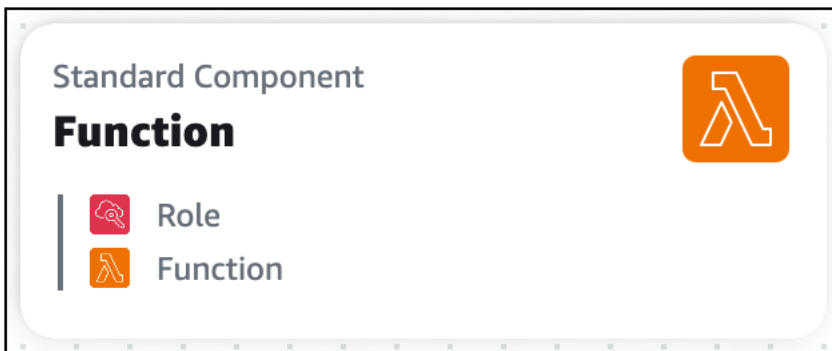
Nel pannello delle proprietà delle risorse della scheda componente standard che rappresenta una `AWS::Lambda::Function` risorsa, facciamo riferimento al ruolo AWS Identity and Access Management (IAM) tramite il relativo ID logico:

The image shows the AWS Infrastructure Composer interface. On the left is a canvas with a grid background. Two standard component cards are visible: a red 'Role' card and a blue 'Function' card. The 'Function' card is highlighted with a blue border. On the right is a 'Resource properties' panel for the 'AWS::Lambda::Function' resource. The panel includes an 'Editing' dropdown set to 'Function', a 'Logical ID' field containing 'Function', and a 'Resource configuration' section with a code editor. The code editor contains the following text:

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

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

Dopo aver salvato il nostro modello, le due schede componente standard si combinano in un'unica scheda componente standard.



## Connessioni a schede in Infrastructure Composer

In AWS Infrastructure Composer, una connessione tra due schede viene visualizzata visivamente da una linea. Queste righe rappresentano relazioni basate sugli eventi all'interno dell'applicazione.

Argomenti

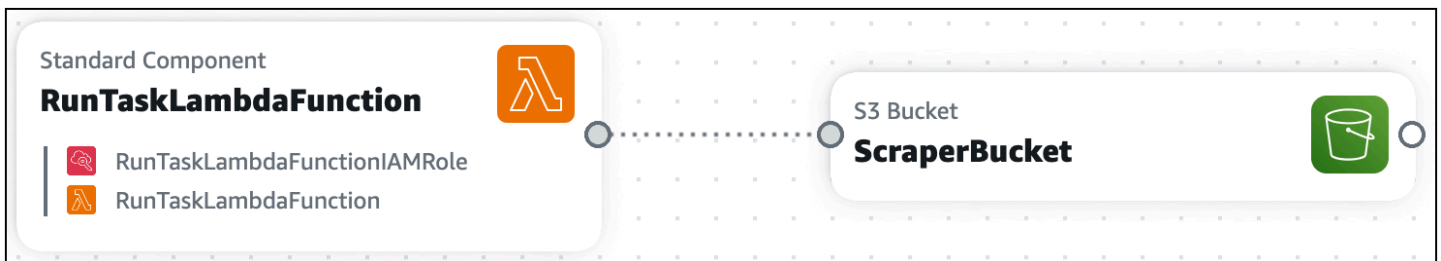
- [Connessioni tra schede](#)
- [Connessioni tra schede componenti avanzate](#)
- [Connessioni da e verso schede di risorse IAc standard](#)

## Connessioni tra schede

Il modo in cui si collegano le schede varia a seconda del tipo di scheda. Ogni scheda avanzata dispone di almeno una porta connettore. Per collegarle, è sufficiente selezionare una porta connettore e trascinarla sulla porta di un'altra scheda. Infrastructure Composer conetterà le due risorse o visualizzerà un messaggio indicante che questa configurazione non è supportata.



Come visto sopra, le linee tra le schede componenti avanzate sono solide. Al contrario, le schede di risorse IAC standard (chiamate anche schede componenti standard) non dispongono di porte di collegamento. Per queste schede, è necessario specificare queste relazioni basate sugli eventi nel modello dell'applicazione e Infrastructure Composer rileverà automaticamente le relative connessioni e le visualizzerà con una linea tratteggiata tra le schede.

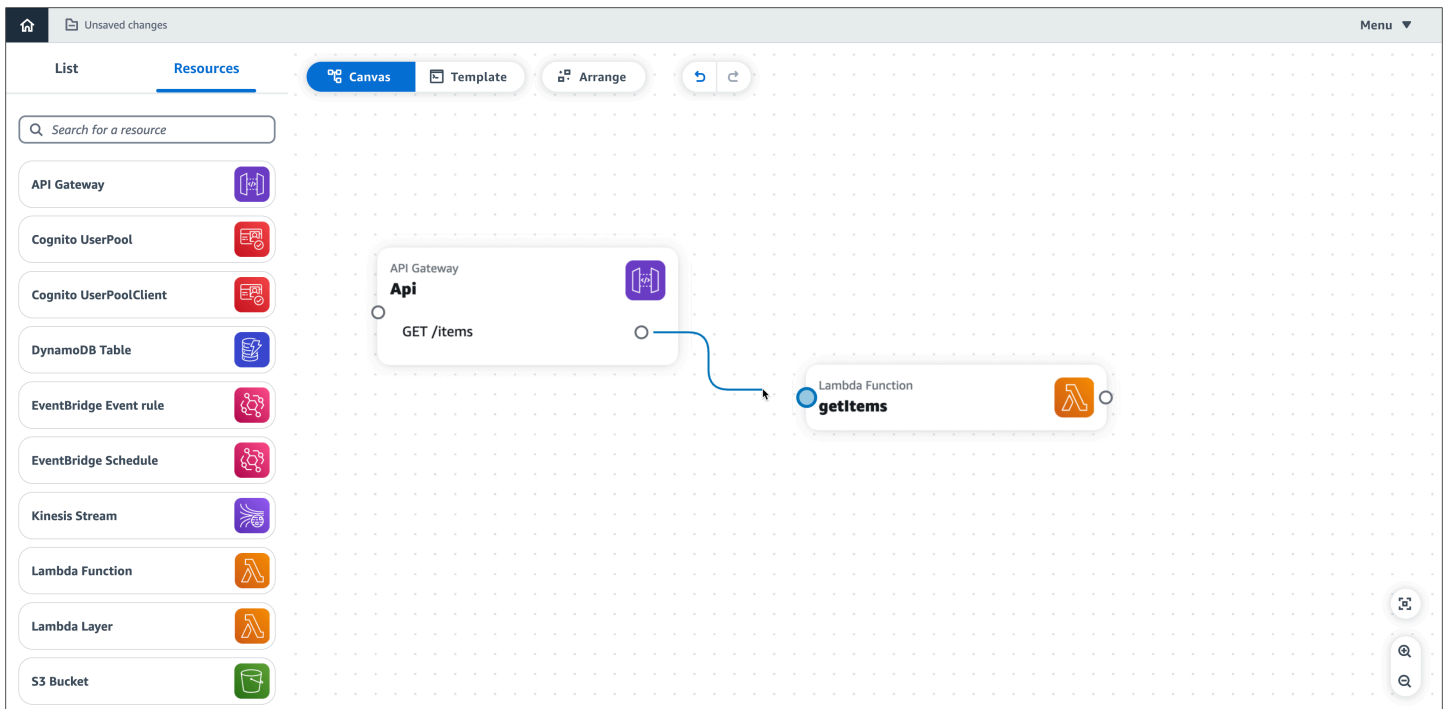


Per ulteriori informazioni, consulta le sezioni seguenti.

## Connessioni tra schede componenti avanzate

In Infrastructure Composer, una connessione tra due schede componenti avanzate viene visualizzata visivamente da una linea continua. Queste linee rappresentano relazioni basate sugli eventi all'interno dell'applicazione.

Per collegare due schede, fate clic su una porta di una scheda e trascinatela sulla porta di un'altra scheda.



### Note

Le schede di risorse IAc standard non dispongono di porte di collegamento. Per queste schede, è necessario specificare le relazioni basate sugli eventi nel modello dell'applicazione

e Infrastructure Composer rileverà automaticamente le relative connessioni e le visualizzerà con una linea tratteggiata tra le schede.

Per ulteriori informazioni, consulta [Connect card sulla tela visiva di Infrastructure Composer](#).

## Quali schede componenti avanzate forniscono le schede componenti

Le connessioni tra due schede, indicate visivamente da una linea, forniscono quanto segue quando necessario:

- AWS Identity and Access Management politiche (IAM)
- Variabili di ambiente
- Eventi

### Policy IAM

Quando una risorsa necessita dell'autorizzazione per richiamare un'altra risorsa, Infrastructure Composer fornisce politiche basate sulle risorse utilizzando AWS Serverless Application Model modelli di policy (AWS SAM).

- Per ulteriori informazioni sulle autorizzazioni e le politiche IAM, consulta [Panoramica della gestione degli accessi: autorizzazioni e politiche](#) nella Guida per l'utente IAM.
- Per saperne di più sui modelli di AWS SAM policy, consulta i modelli di [AWS SAM policy](#) nella AWS Serverless Application Model Developer Guide.

### Variabili di ambiente

Le variabili di ambiente sono valori temporanei che possono essere modificati per influire sul comportamento delle risorse. Se necessario, Infrastructure Composer definisce il codice dell'infrastruttura per utilizzare le variabili di ambiente tra le risorse.

### Eventi

Le risorse possono richiamare un'altra risorsa tramite diversi tipi di eventi. Se necessario, Infrastructure Composer definisce il codice di infrastruttura necessario affinché le risorse interagiscano attraverso tipi di eventi.

## Connessioni da e verso schede di risorse IAc standard

Tutte le CloudFormation risorse possono essere utilizzate come schede di risorse IaC standard dalla palette Risorse. Quando trascini una scheda di risorse IaC standard sull'area di lavoro, una scheda di risorse IaC standard diventa una scheda componente standard e ciò richiede a Infrastructure Composer di creare un modello di partenza per la risorsa nell'applicazione.

Per ulteriori informazioni, consulta [Schede standard in Infrastructure Composer](#).

# Guida introduttiva alla console Infrastructure Composer

Utilizza gli argomenti di questa sezione per configurare AWS Infrastructure Composer e imparare a progettare un'applicazione utilizzando la relativa tela visiva. Il tour e i tutorial di questa sezione vengono visualizzati nella console Infrastructure Composer, che è l'esperienza utente predefinita. Gli argomenti di questa sezione mostrano come completare i prerequisiti per l'utilizzo di Infrastructure Composer, utilizzare la console Infrastructure Composer, caricare e modificare un progetto e creare la prima applicazione.

Infrastructure Composer è disponibile anche da e in modalità console. AWS Toolkit for Visual Studio Code CloudFormation Le esperienze tra gli strumenti sono generalmente le stesse, ma ci sono alcune differenze tra gli strumenti. Per informazioni dettagliate sull'utilizzo di Infrastructure Composer in ciascuno di questi strumenti, consulta [Dove puoi usare Infrastructure Composer](#).

## Argomenti

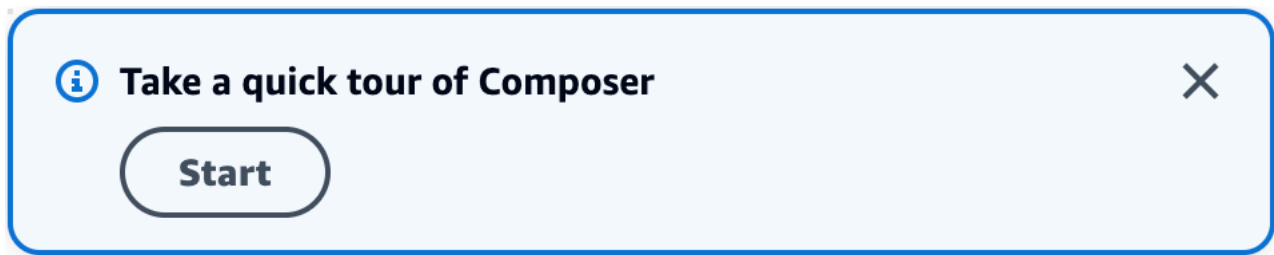
- [Fai un tour della console Infrastructure Composer](#)
- [Caricare e modificare il progetto demo di Infrastructure Composer](#)
- [Crea la tua prima applicazione con Infrastructure Composer](#)

## Fai un tour della console Infrastructure Composer

Per avere un'idea generale di come AWS Infrastructure Composer funziona, segui il tour integrato nella console Infrastructure Composer. Per una panoramica della console Infrastructure Composer, consulta [Fai un tour della console Infrastructure Composer](#) Per indicazioni dettagliate sull'uso di Infrastructure Composer, fare riferimento a [Come comporre in AWS Infrastructure Composer](#)

Per fare un tour di Infrastructure Composer

1. Accedi alla console [Infrastructure Composer](#).
2. Nella home page, scegli Apri demo.
3. Nell'angolo in alto a destra, nella finestra Fai una rapida panoramica di Composer, scegli Start.



4. Nella finestra del tour di Composer, procedi come segue:

- Per passare alla fase successiva, scegli Avanti.
- Per tornare al passaggio precedente, scegli Precedente.
- Nell'ultimo passaggio, per terminare il tour, scegli Fine.

Il tour offre una breve panoramica delle funzionalità di base di Infrastructure Composer, come l'uso, la configurazione e il collegamento delle schede. Per ulteriori informazioni, vedi [Come comporre in AWS Infrastructure Composer](#).

## Fasi successive

Per caricare e modificare un progetto in Infrastructure Composer, vedere. [Caricare e modificare il progetto demo di Infrastructure Composer](#)

## Caricare e modificare il progetto demo di Infrastructure Composer

Usa questo tutorial per acquisire familiarità con l'interfaccia utente di Infrastructure Composer e imparare a caricare, modificare e salvare il progetto demo di Infrastructure Composer.

Questo tutorial viene eseguito nella console Infrastructure Composer. Una volta completato, sarai pronto per iniziare [Crea la tua prima applicazione con Infrastructure Composer](#).

### Argomenti

- [Passaggio 1: apri la demo](#)
- [Fase 2: Esplora la tela visiva di Infrastructure Composer](#)
- [Fase 3: Espandi l'architettura dell'applicazione](#)
- [Fase 4: Salvare l'applicazione](#)
- [Fasi successive](#)

## Passaggio 1: apri la demo

Inizia a utilizzare Infrastructure Composer creando un progetto demo.

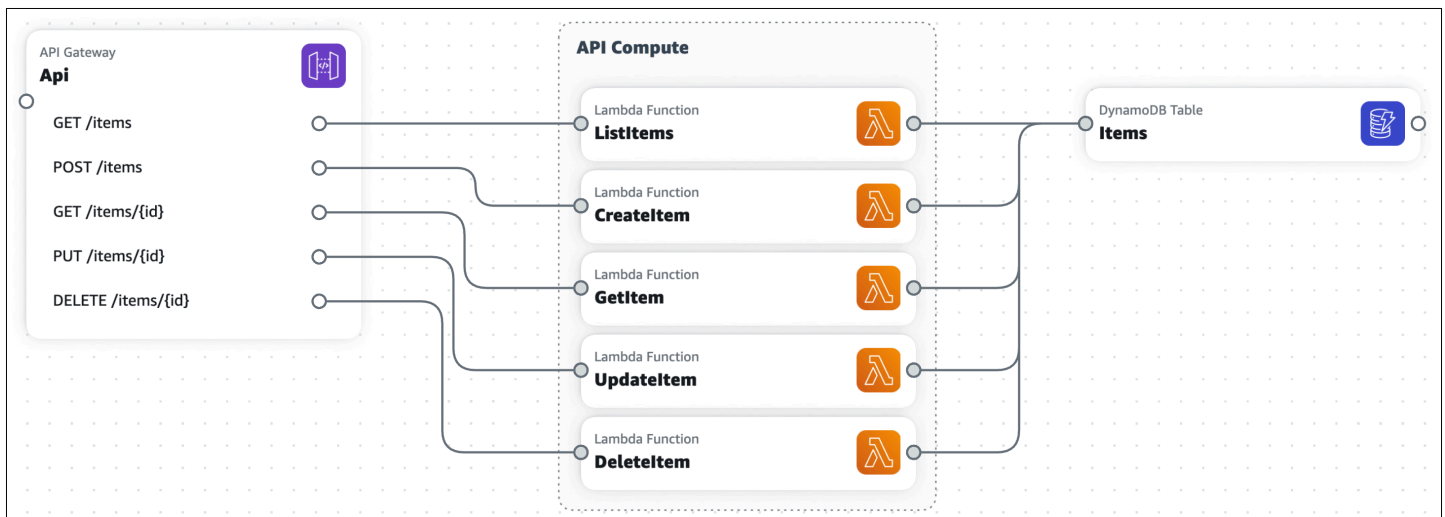
Per creare un progetto dimostrativo

1. Accedi alla [console Infrastructure Composer](#).
2. Nella home page, scegli Apri demo.

L'applicazione demo è un'applicazione serverless (CRUD) di base per la creazione, la lettura, l'eliminazione e l'aggiornamento che include:

- Una risorsa Amazon API Gateway con cinque percorsi.
- Cinque AWS Lambda funzioni.
- Una tabella Amazon DynamoDB.

L'immagine seguente è della demo:

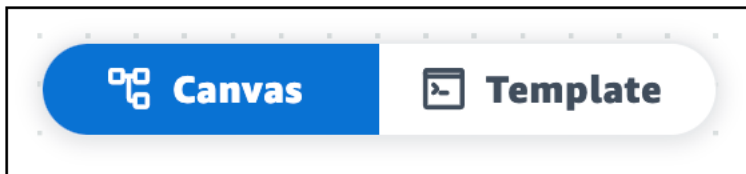


## Fase 2: Esplora la tela visiva di Infrastructure Composer

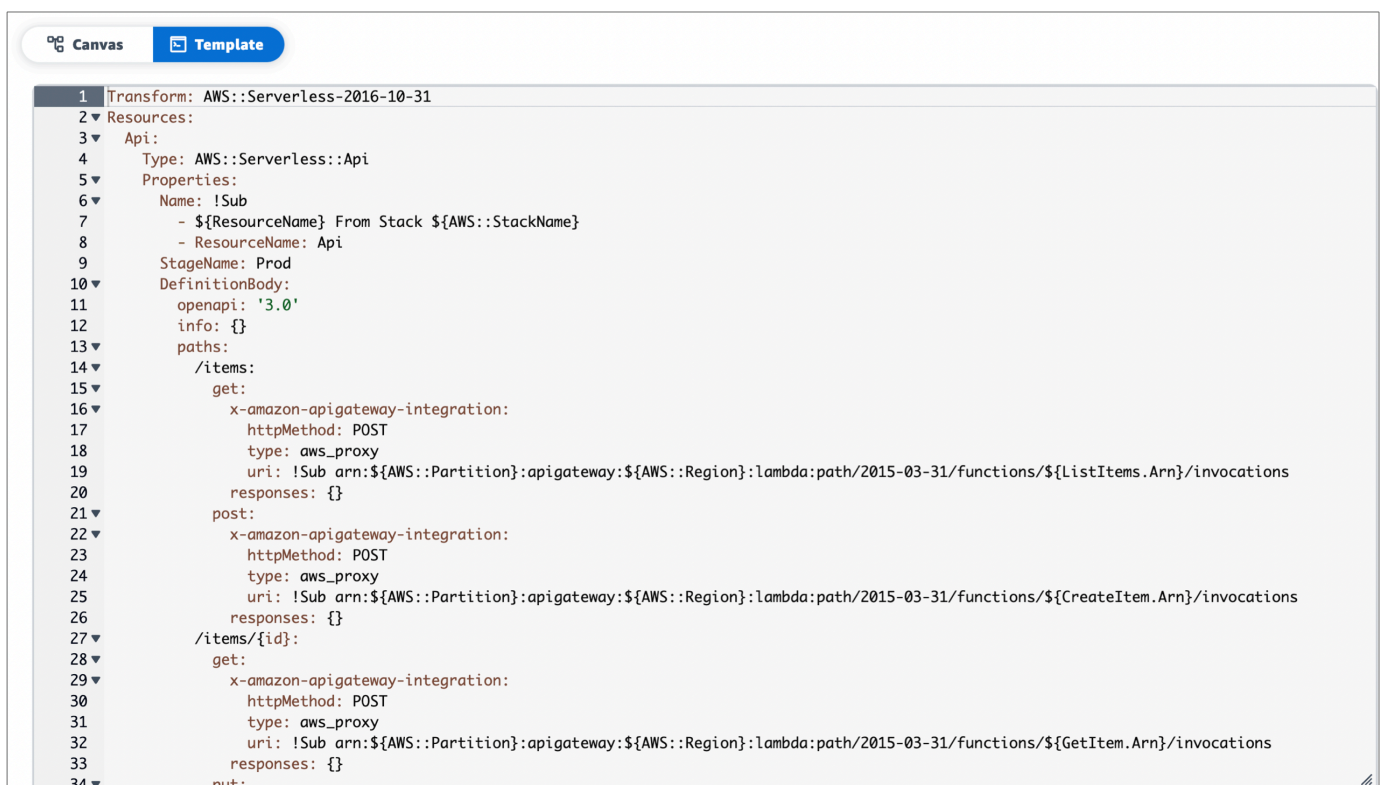
Scopri le funzionalità del visual canvas per creare il tuo progetto dimostrativo di Infrastructure Composer. Per una panoramica del layout della tela visiva, consulta [panoramica visiva](#).

## Per esplorare le caratteristiche della tela visiva

1. Quando aprite un progetto applicativo nuovo o esistente, Infrastructure Composer carica la vista canvas, come indicato sopra l'area di visualizzazione principale.

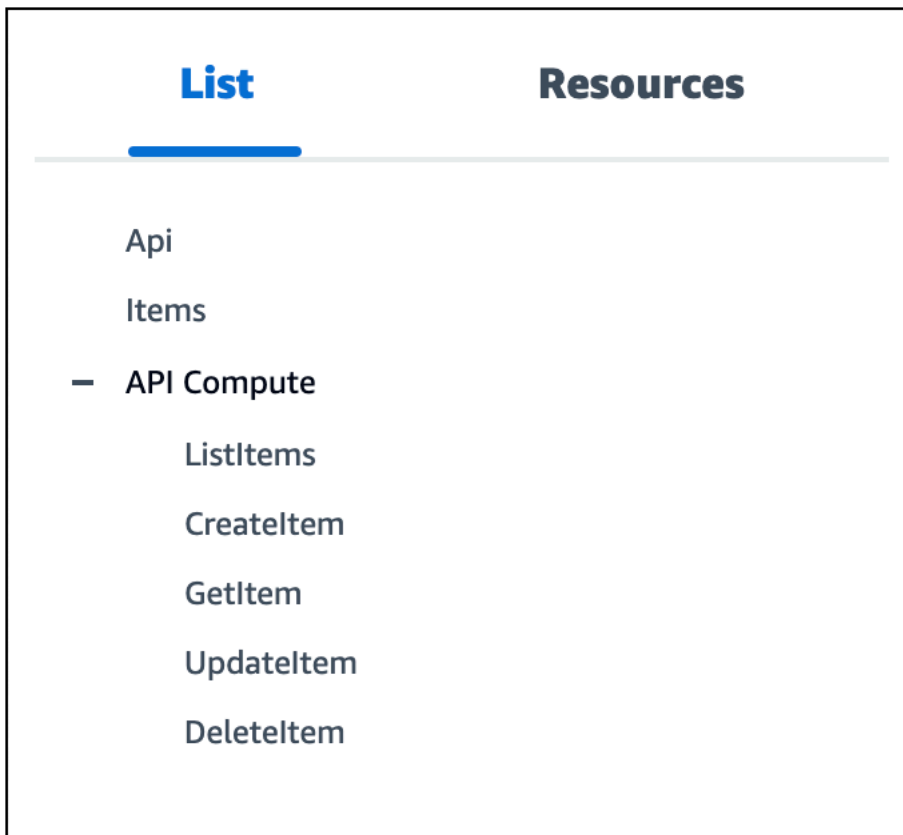


Per mostrare il codice dell'infrastruttura dell'applicazione nell'area di visualizzazione principale, scegli Template. Ad esempio, ecco la visualizzazione del modello AWS Serverless Application Model (AWS SAM) del progetto demo di Infrastructure Composer.

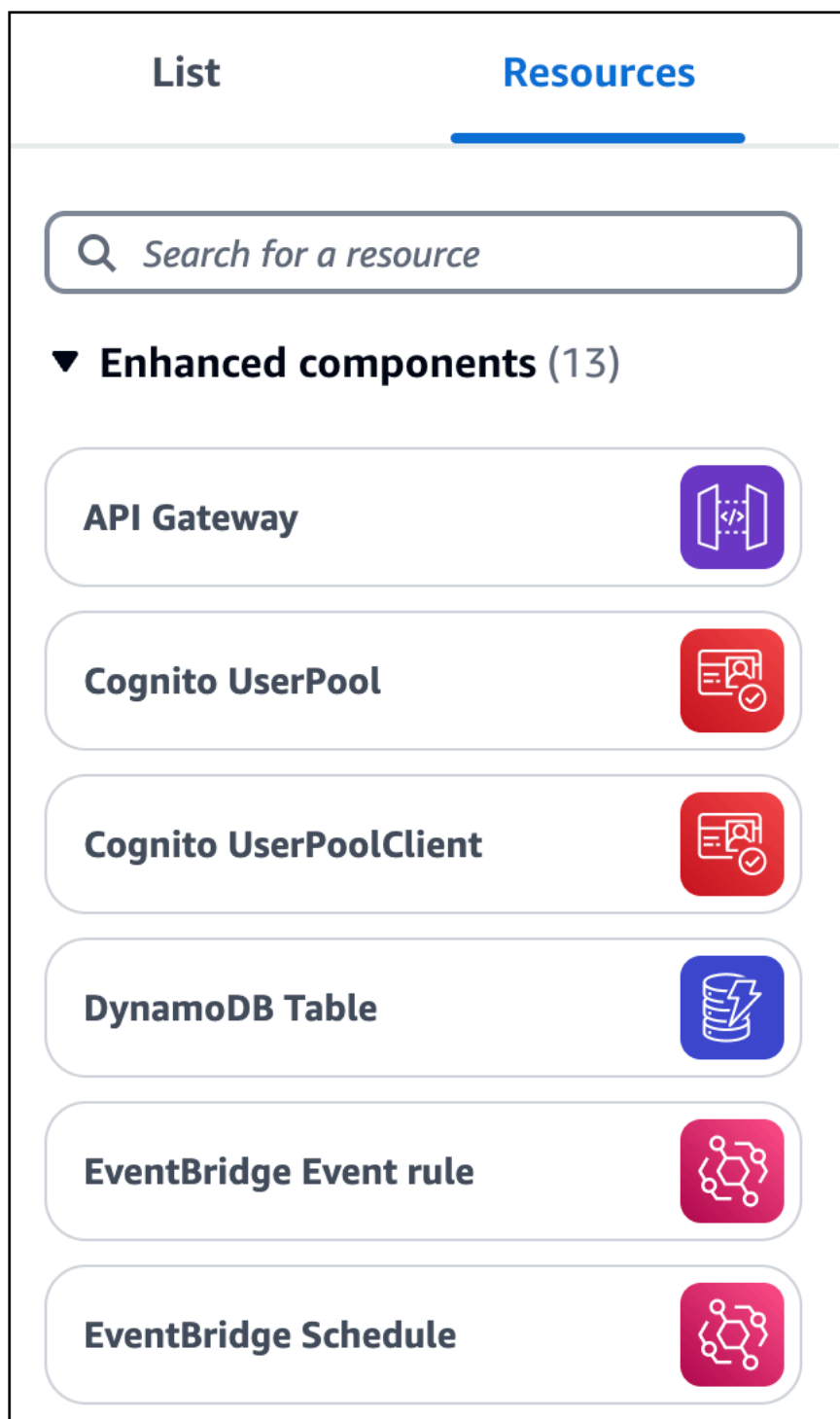
A screenshot of the Infrastructure Composer interface. At the top, there are two buttons: 'Canvas' and 'Template'. The 'Template' button is selected and highlighted in blue. Below the buttons is a code editor displaying the AWS SAM JSON code for a Serverless Application Model. The code is as follows:

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

2. Per mostrare nuovamente la visualizzazione canvas dell'applicazione, scegli Canvas.
3. Per mostrare le risorse dell'applicazione organizzate in una visualizzazione ad albero, scegliete Elenco.



4. Per mostrare la tavolozza delle risorse, scegliete Risorse. Questa palette include schede che puoi utilizzare per espandere l'architettura dell'applicazione. È possibile cercare le schede o scorrere l'elenco.



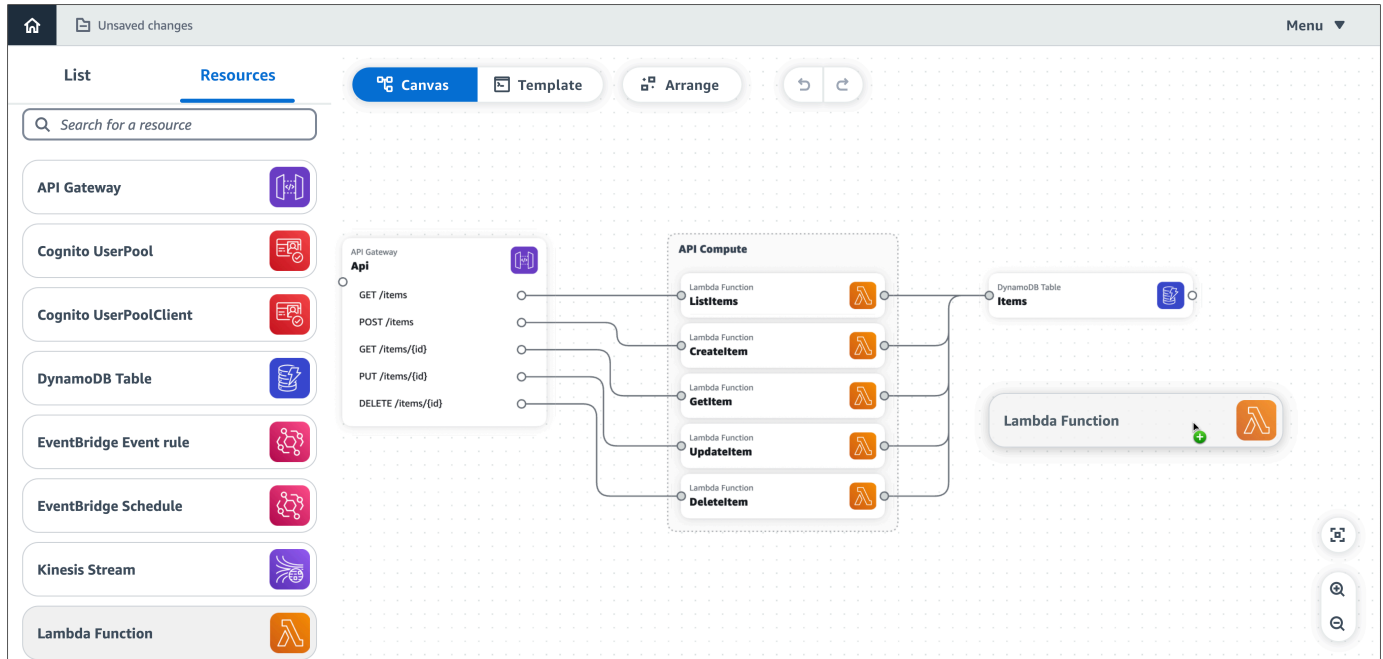
5. Per muoverti all'interno dell'area di visualizzazione, usa i gesti di base. Per ulteriori informazioni, consulta [Posiziona le carte sulla tela](#).

## Fase 3: Espandi l'architettura dell'applicazione

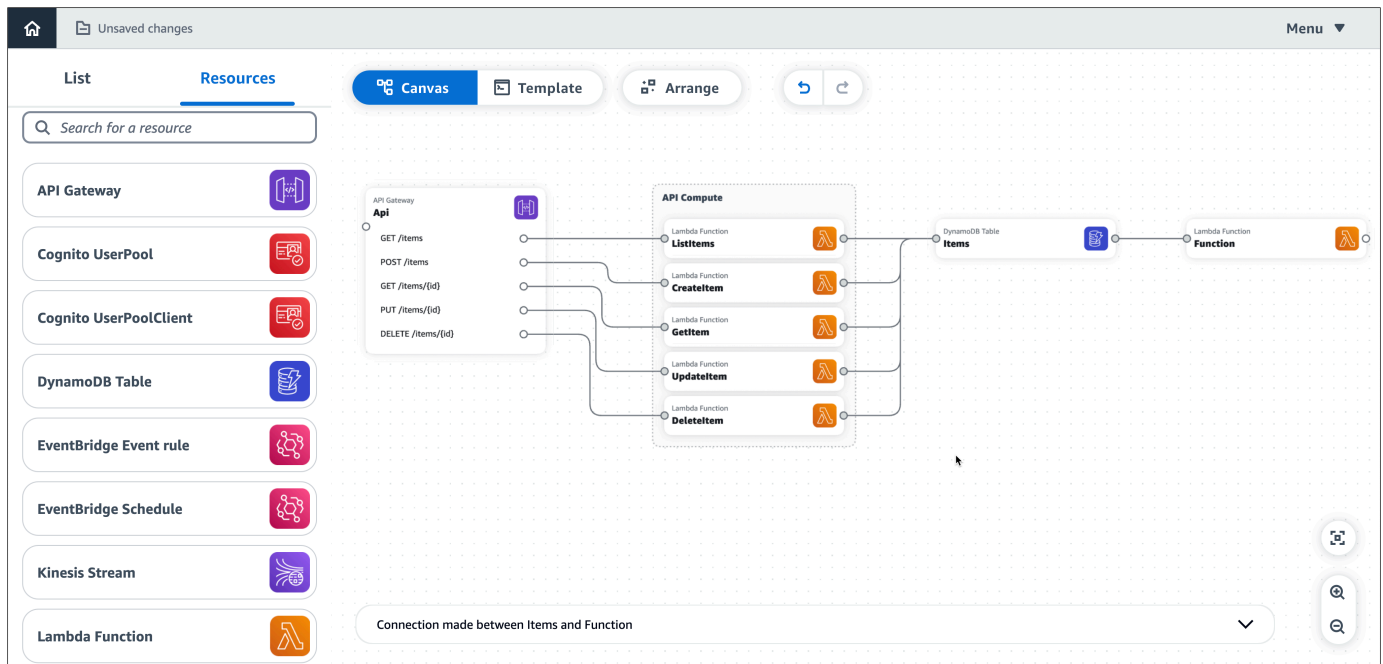
In questo passaggio, espanderai l'architettura dell'applicazione aggiungendo una funzione Lambda alla tabella DynamoDB.

Per aggiungere una funzione Lambda alla tabella DynamoDB

1. Dalla palette delle risorse (Risorse), trascina la scheda componente avanzata della funzione Lambda sull'area di disegno, a destra della scheda DynamoDB Table.



2. Connect la tabella DynamoDB alla funzione Lambda. Per collegarli, fai clic sulla porta destra della scheda DynamoDB Table e trascinala sulla porta sinistra della scheda Lambda Function.
3. Scegli Disponi per organizzare le schede nella visualizzazione dell'area di disegno.



4. Configura la tua funzione Lambda. Per configurarla, esegui una delle seguenti operazioni:
  - Nella vista canvas, modificate le proprietà della funzione nel pannello delle proprietà delle risorse. Per aprire il pannello, fai doppio clic sulla scheda Function Lambda. In alternativa, seleziona la scheda, quindi scegli Dettagli. Per ulteriori informazioni sulle proprietà configurabili della funzione Lambda elencate nel pannello delle proprietà delle risorse, consulta la Guida per [AWS Lambda gli sviluppatori](#).
  - Nella vista modello, modifica il codice per la tua funzione `()AWS::Serverless::Function`. Infrastructure Composer sincronizza automaticamente le modifiche sull'area di disegno. Per ulteriori informazioni sulla risorsa funzionale in un AWS SAM modello, vedere [AWS::Serverless::Function](#) nel riferimento alle AWS SAM risorse e alle proprietà.

## Fase 4: Salvare l'applicazione

Salvate l'applicazione salvando manualmente il modello dell'applicazione sul computer locale o attivando la sincronizzazione locale.

Per salvare manualmente il modello di applicazione

1. Dal menu, seleziona Salva > Salva file modello.
2. Fornisci un nome per il modello e scegli una posizione sul computer locale in cui salvare il modello. Premi Salva.

Per istruzioni sull'attivazione della sincronizzazione locale, consulta [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#).

## Fasi successive

Per iniziare a creare la tua prima applicazione, consulta [Crea la tua prima applicazione con Infrastructure Composer](#).

## Crea la tua prima applicazione con Infrastructure Composer

In questo tutorial, si usa AWS Infrastructure Composer per creare un'applicazione serverless CRUD (Create, Read, Update and Delete) che gestisce gli utenti in un database.

Per questo tutorial, utilizziamo Infrastructure Composer in. Console di gestione AWS Ti consigliamo di utilizzare Google Chrome or Microsoft Edge e una finestra del browser a schermo intero.

### Sei nuovo nel mondo serverless?

Ti consigliamo una conoscenza di base dei seguenti argomenti:

- [Architettura basata su eventi](#)
- [Infrastruttura come codice \(IaC\)](#)
- [Tecnologie serverless](#)

Per ulteriori informazioni, consulta [Concetti serverless per AWS Infrastructure Composer](#).

### Argomenti

- [Riferimento alle proprietà delle risorse](#)
- [Fase 1: Crea il tuo progetto](#)
- [Passaggio 2: aggiungi carte alla tela](#)
- [Fase 3: Configurazione dell'API Gateway REST API](#)
- [Fase 4: Configurazione delle funzioni Lambda](#)
- [Fase 5: Connect le tue carte](#)
- [Fase 6: Organizza la tela](#)

- [Fase 7: Aggiungere e connettere una tabella DynamoDB](#)
- [Fase 8: Rivedi il modello AWS CloudFormation](#)
- [Fase 9: Integrazione nei flussi di lavoro di sviluppo](#)
- [Fasi successive](#)

## Riferimento alle proprietà delle risorse

Durante la creazione dell'applicazione, utilizza questa tabella come riferimento per configurare le proprietà di Amazon API Gateway e AWS Lambda delle risorse.

Metodo	Path	Nome funzione
GET	/articoli	Ottieni articoli
GET	/items/ {id}	getItem
PUT	/oggetti/ {id}	Aggiorna elemento
POST	/articolo	Aggiungi elemento
DELETE	/items/ {id}	deleteItem

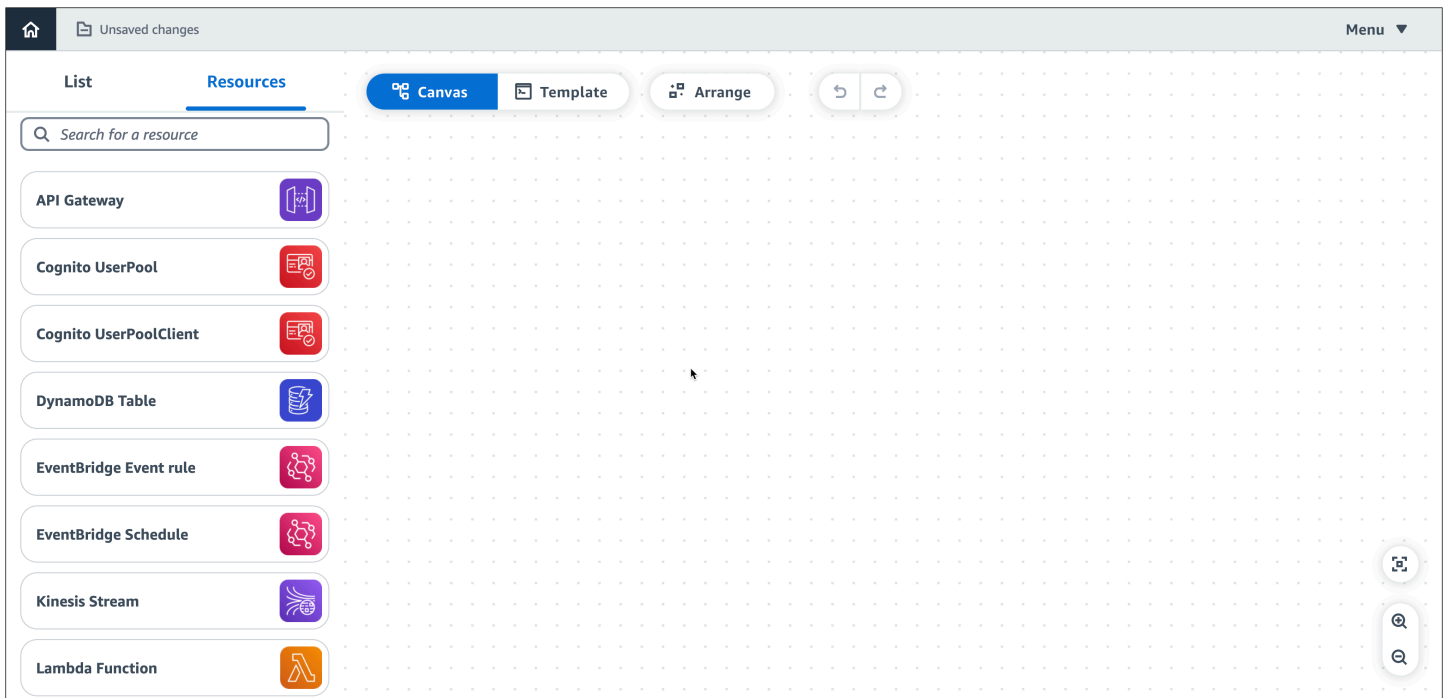
## Fase 1: Crea il tuo progetto

Per iniziare con la tua applicazione serverless CRUD, crea un nuovo progetto in Infrastructure Composer e attiva la sincronizzazione locale.

Per creare un nuovo progetto vuoto

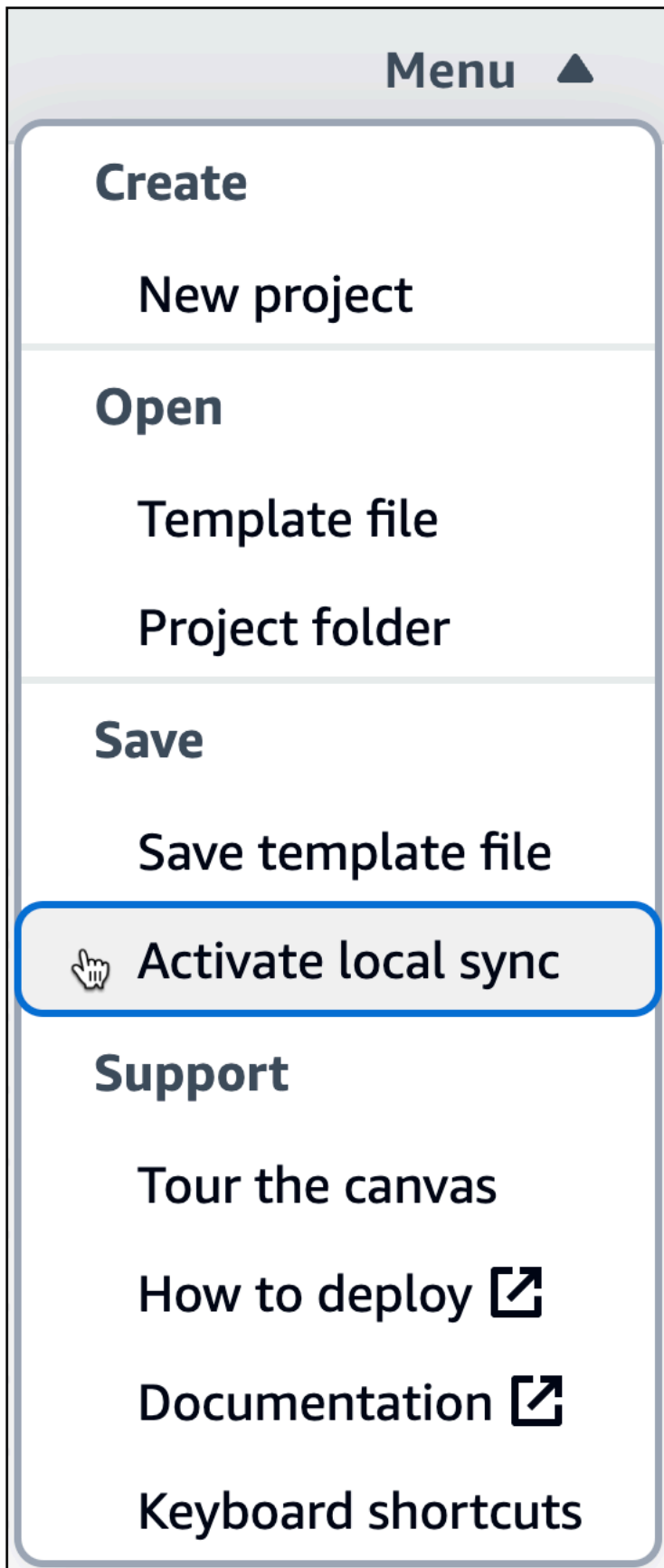
1. Accedi alla [console Infrastructure Composer](#).
2. Nella home page, scegli Crea progetto.

Come mostrato nell'immagine seguente, Infrastructure Composer apre la tela visiva e carica un modello di applicazione iniziale (vuoto).



Per attivare la sincronizzazione locale

1. Dal menu Infrastructure Composer, seleziona Salva > Attiva la sincronizzazione locale.



2. Per Posizione del progetto, premi Seleziona cartella e scegli una directory. Qui è dove Infrastructure Composer salverà e sincronizzerà i file e le cartelle del modello durante la progettazione.

La posizione del progetto non deve contenere un modello di applicazione esistente.

#### Note

La sincronizzazione locale richiede un browser che supporti l'API File System Access. Per ulteriori informazioni, consulta [Data Infrastructure Composer ottiene l'accesso a](#).

3. Quando viene richiesto di consentire l'accesso, seleziona Visualizza file.
4. Premi Attiva per attivare la sincronizzazione locale. Quando viene richiesto di salvare le modifiche, seleziona Salva modifiche.

Quando attivato, l'indicatore Autosave verrà visualizzato nell'area in alto a sinistra dell'area di disegno.

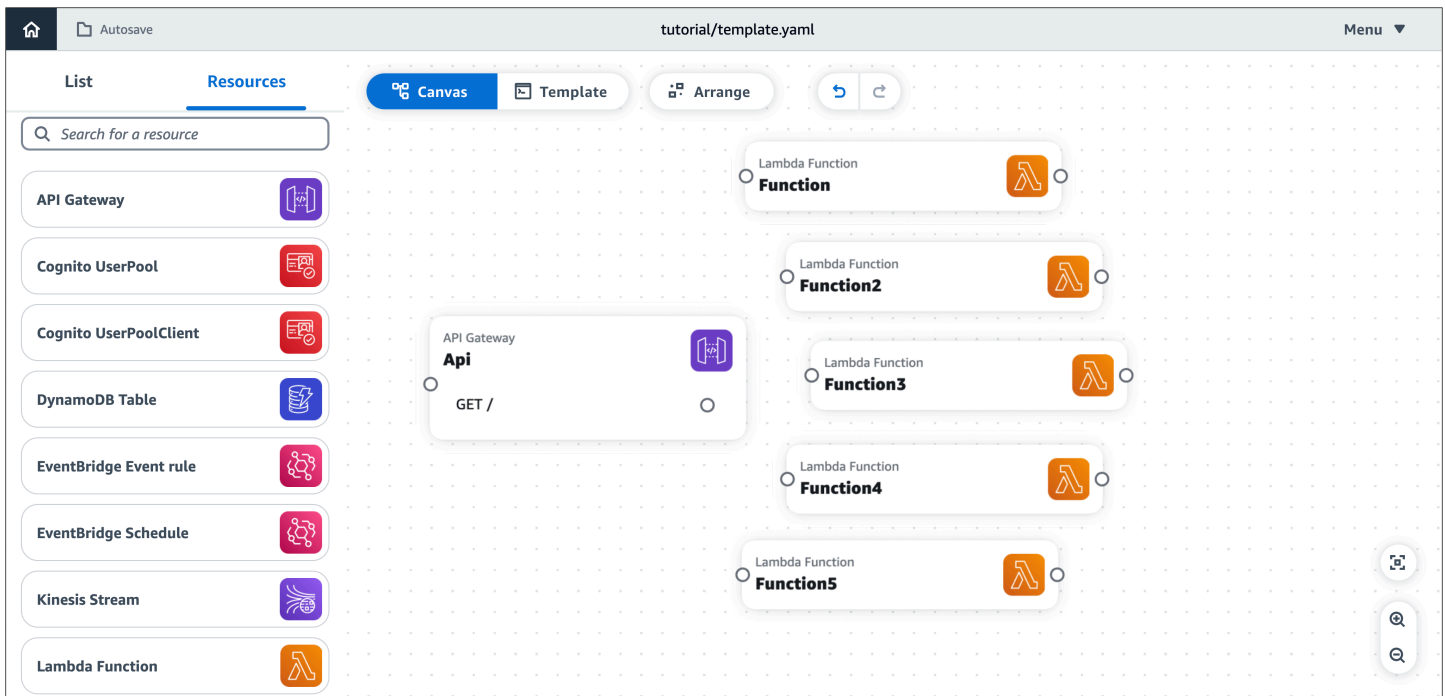
## Passaggio 2: aggiungi carte alla tela

Inizia a progettare l'architettura della tua applicazione utilizzando schede componenti avanzate, iniziando con un'API API Gateway REST e cinque funzioni Lambda.

Per aggiungere schede API Gateway e Lambda all'area di disegno

Dalla palette Risorse, nella sezione Componenti avanzati, procedi come segue:

1. Trascina una scheda API Gateway sull'area di disegno.
2. Trascina una scheda Lambda Function sull'area di disegno. Ripeti l'operazione fino ad aggiungere cinque schede Lambda Function all'area di disegno.



## Fase 3: Configurazione dell'API Gateway REST API

Quindi, aggiungi cinque percorsi nella tua scheda API Gateway.

Per aggiungere percorsi alla scheda API Gateway

1. Apri il pannello delle proprietà delle risorse per la scheda API Gateway. Per aprire il pannello, fate doppio clic sulla scheda. In alternativa, seleziona la carta, quindi scegli Dettagli.
2. Nel pannello delle proprietà delle risorse, in Percorsi, procedi come segue:

### Note

Per ciascuna delle seguenti rotte, utilizzate il metodo HTTP e i valori del percorso specificati nella [tabella di riferimento delle proprietà delle risorse](#).

- a. Per Metodo, scegliete il metodo HTTP specificato. Ad esempio, GET.
  - b. Per Path, immettete il percorso specificato. Ad esempio, **/items**.
  - c. Seleziona Aggiungi route.
  - d. Ripeti i passaggi precedenti finché non avrai aggiunto tutti e cinque i percorsi specificati.
3. Scegli Save (Salva).

The screenshot displays the AWS Infrastructure Composer interface for a template named 'tutorial/template.yaml'. On the left, a 'Resources' panel lists various AWS services. The central canvas shows a diagram with five Lambda Function resources (Function, Function2, Function3, Function4, Function5) and one API Gateway resource named 'Api'. The 'Api' resource is expanded to show its routes: GET /items, GET /items/{id}, PUT /items/{id}, POST /item, and DELETE /items/{id}. On the right, the 'Resource properties' panel is open for a selected Lambda Function, showing fields for Method (GET, PUT, POST) and Path (/items/{id}).

## Fase 4: Configurazione delle funzioni Lambda

Assegna un nome a ciascuna delle cinque funzioni Lambda come specificato nella tabella di [riferimento delle proprietà delle risorse](#).

Per assegnare un nome alle funzioni Lambda

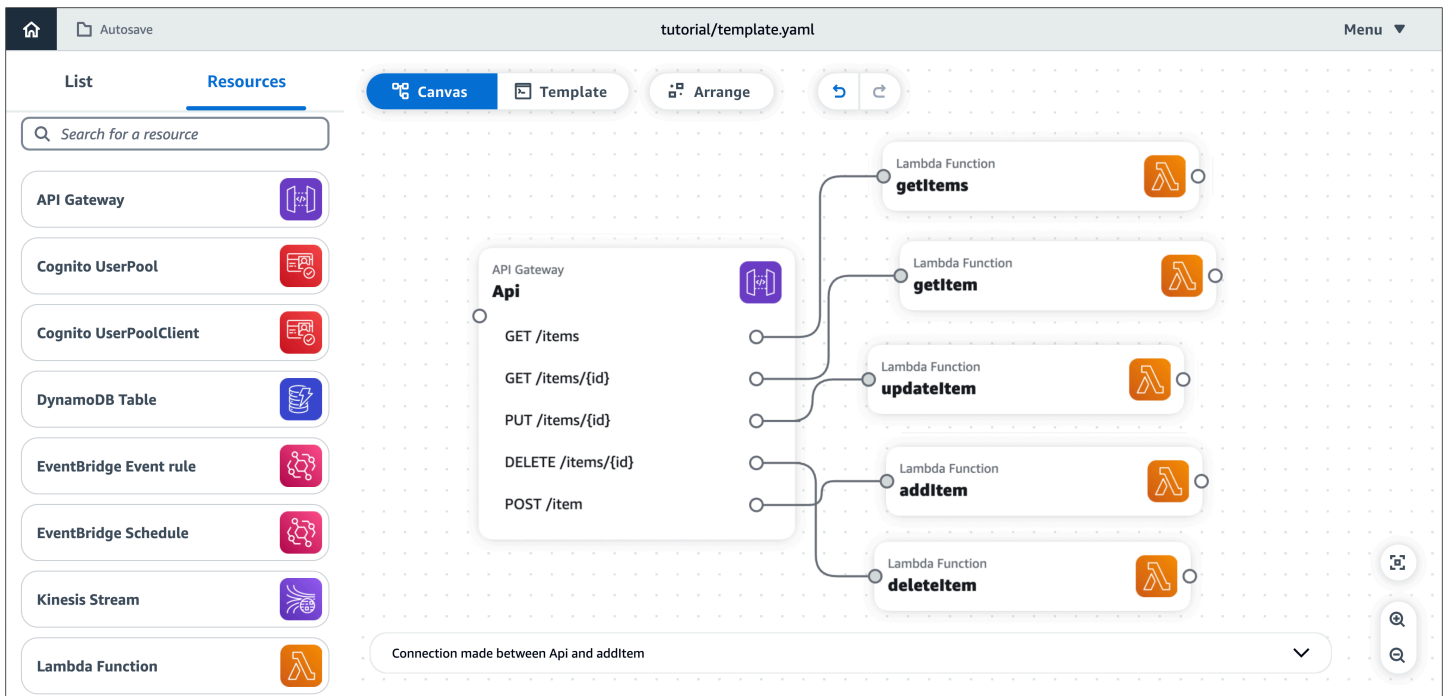
1. Apri il pannello delle proprietà delle risorse di una scheda Function Lambda. Per aprire il pannello, fai doppio clic sulla scheda. In alternativa, seleziona la carta, quindi scegli Dettagli.
2. Nel pannello delle proprietà della risorsa, in Logical ID, immettete il nome di una funzione specificata. Ad esempio, **getItems**.
3. Scegli Save (Salva).
4. Ripetete i passaggi precedenti finché non avete dato un nome a tutte e cinque le funzioni.

## Fase 5: Connect le tue carte

Connect ogni route sulla scheda API Gateway alla relativa scheda Lambda Function, come specificato nella tabella di [riferimento delle proprietà delle risorse](#).

Per connettere le tue carte

1. Fai clic su una porta destra sulla scheda API Gateway e trascinala sulla porta sinistra della scheda Lambda Function specificata. Ad esempio, fai clic sulla porta GET /items e trascinala nella porta sinistra di getItems.
2. Ripeti il passaggio precedente fino a connettere tutte e cinque le rotte sulla scheda API Gateway alle corrispondenti schede Lambda Function.



## Fase 6: Organizza la tela

Organizza l'area di visualizzazione raggruppando le funzioni Lambda e disponendo tutte le schede.

Per raggruppare le tue funzioni

1. Tieni premuto Shift, quindi seleziona ogni scheda Lambda Function sull'area di disegno.
2. Scegli Gruppo.

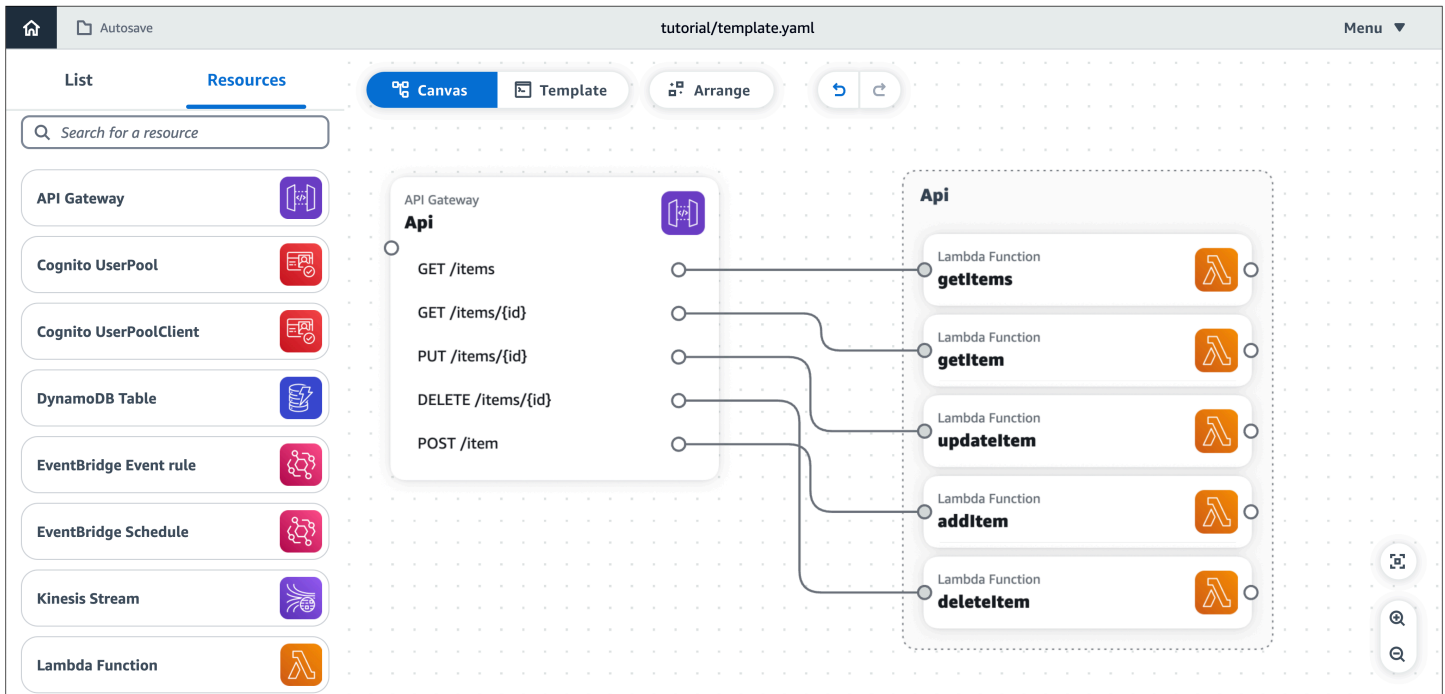
Per assegnare un nome al gruppo

1. Fai doppio clic sulla parte superiore del gruppo, vicino al nome del gruppo (Gruppo).  
Si apre il pannello delle proprietà del gruppo.
2. Nel pannello delle proprietà del gruppo, per Nome del gruppo, immettere **API**.
3. Scegli Save (Salva).

Per disporre le carte

Nell'area di disegno, sopra l'area di visualizzazione principale, scegli Disponi.

Infrastructure Composer organizza e allinea tutte le schede sull'area di disegno visuale, incluso il nuovo gruppo (API), come illustrato di seguito:

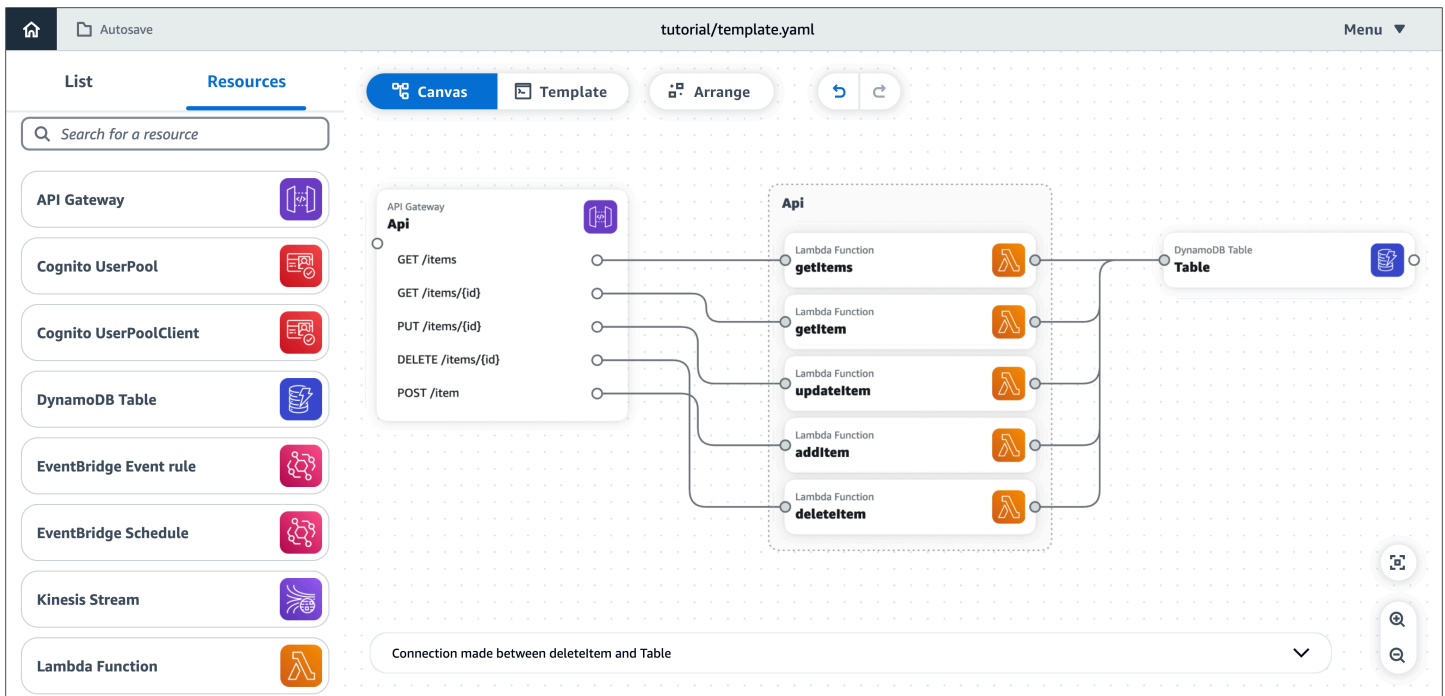


## Fase 7: Aggiungere e connettere una tabella DynamoDB

Ora aggiungete una tabella DynamoDB all'architettura dell'applicazione e collegatela alle funzioni Lambda.

Per aggiungere e connettere una tabella DynamoDB

1. Dalla palette delle risorse (Risorse), nella sezione Componenti avanzati, trascina una scheda DynamoDB Table sull'area di disegno.
2. Fai clic sulla porta destra di una scheda Lambda Function e trascinala sulla porta sinistra della scheda DynamoDB Table.
3. Ripeti il passaggio precedente fino a collegare tutte e cinque le schede Function Lambda alla scheda DynamoDB Table.
4. (Facoltativo) Per riorganizzare e riallineare le carte sulla tela, scegli Disponi.

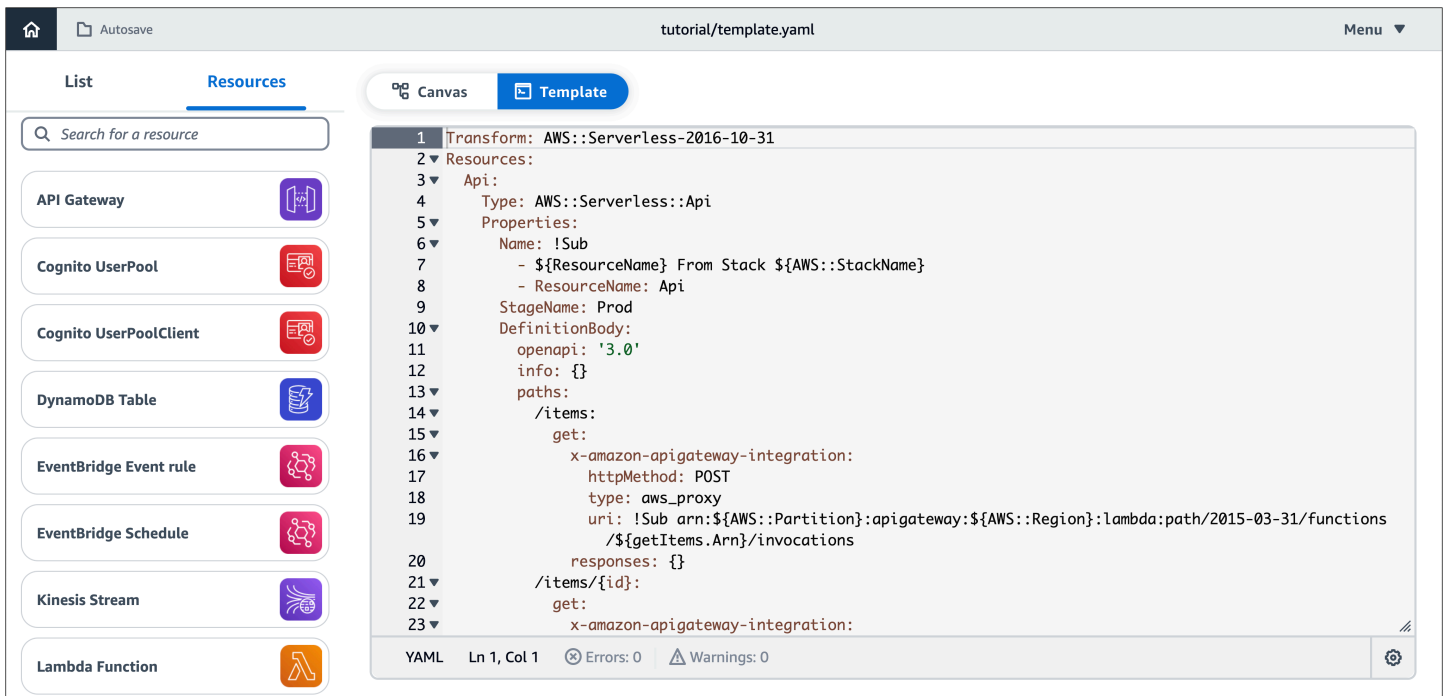


## Fase 8: Rivedi il modello AWS CloudFormation

Complimenti! Hai progettato con successo un'applicazione serverless pronta per l'implementazione. Infine, scegli Template per esaminare il AWS CloudFormation modello che Infrastructure Composer ha generato automaticamente per te.

Nel modello, Infrastructure Composer ha definito quanto segue:

- La Transform dichiarazione, che specifica il modello come modello AWS Serverless Application Model (AWS SAM). Per ulteriori informazioni, consulta l'[anatomia AWS SAM del modello nella Guida](#) per gli AWS Serverless Application Model sviluppatori.
- Una `AWS::Serverless::Api` risorsa che specifica l'API REST dell'API Gateway con le sue cinque rotte.
- Cinque `AWS::Serverless::Function` risorse, che specificano le configurazioni delle funzioni Lambda, comprese le variabili di ambiente e le politiche di autorizzazione.
- Una `AWS::DynamoDB::Table` risorsa che specifica la tabella DynamoDB e le relative proprietà.
- La **Metadata** sezione, che contiene informazioni sul tuo gruppo di risorse (API). Per ulteriori informazioni su questa sezione, consulta [Metadati](#) nella Guida per l'AWS CloudFormation utente.



## Fase 9: Integrazione nei flussi di lavoro di sviluppo

Utilizza il file modello e le directory di progetto create da Infrastructure Composer per ulteriori test e implementazioni.

- Con la sincronizzazione locale, puoi connettere Infrastructure Composer all'IDE sul tuo computer locale per accelerare lo sviluppo. Per ulteriori informazioni, consulta [Connect la console Infrastructure Composer all'IDE locale](#).
- Con la sincronizzazione locale, puoi utilizzare l'interfaccia a riga di AWS Serverless Application Model comando (AWS SAM CLI) sul tuo computer locale per testare e distribuire l'applicazione. Per ulteriori informazioni, consulta [Distribuisci la tua applicazione serverless Infrastructure Composer nel cloud AWS](#).

## Fasi successive

Ora sei pronto per creare le tue applicazioni con Infrastructure Composer. Per dettagli approfonditi sull'utilizzo di Infrastructure Composer, consulta. [Come comporre in AWS Infrastructure Composer](#) Quando siete pronti per distribuire l'applicazione, fate riferimento a. [Distribuisci la tua applicazione serverless Infrastructure Composer nel cloud AWS](#)

# Dove puoi usare Infrastructure Composer

È possibile utilizzare Infrastructure Composer dalla relativa console AWS Toolkit for Visual Studio Code, da e in Infrastructure Composer in CloudFormation modalità console. Sebbene ognuno vari a seconda dei casi d'uso leggermente diversi, nel complesso si tratta di esperienze simili. Questa sezione fornisce i dettagli di ogni esperienza.

L'argomento [Utilizzo della AWS Infrastructure Composer console](#) offre una panoramica completa dell'esperienza di console predefinita. L'argomento [CloudFormation modalità console](#) fornisce dettagli su una versione di Infrastructure Composer integrata con il flusso di lavoro dello CloudFormation stack. [AWS Toolkit for Visual Studio Code](#) fornisce informazioni sull'accesso e l'utilizzo di Infrastructure Composer in VS Code.

## Argomenti

- [Utilizzo della AWS Infrastructure Composer console](#)
- [Utilizzo di Infrastructure Composer in modalità CloudFormation console](#)
- [Utilizzo di Infrastructure Composer di AWS Toolkit for Visual Studio Code](#)

## Utilizzo della AWS Infrastructure Composer console

Questa sezione fornisce dettagli sull'accesso e sull'utilizzo AWS Infrastructure Composer dalla console Infrastructure Composer. Questa è l'esperienza predefinita per Infrastructure Composer ed è un buon modo per acquisire familiarità con Infrastructure Composer. È inoltre possibile integrare la console Infrastructure Composer con l'IDE locale. Per informazioni dettagliate, vedi [Connect la console Infrastructure Composer all'IDE locale](#).

Puoi anche [accedere a Infrastructure Composer dal AWS Toolkit in VS Code](#) e puoi utilizzare una [modalità di Infrastructure Composer progettata specificamente per essere utilizzata](#) in CloudFormation

Per la documentazione generale sull'uso di Infrastructure Composer, vedere. [Come comporre](#)

## Argomenti

- [AWS Infrastructure Composer panoramica visiva della console](#)
- [Gestisci il tuo progetto dalla console Infrastructure Composer](#)

- [Connect la console Infrastructure Composer all'IDE locale](#)
- [Consenti l'accesso alle pagine Web ai file locali in Infrastructure Composer](#)
- [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#)
- [Importa le funzioni in Infrastructure Composer dalla console Lambda](#)
- [Esporta un'immagine della tela visiva di Infrastructure Composer](#)

## AWS Infrastructure Composer panoramica visiva della console

Questa sezione fornisce una panoramica visiva della AWS Infrastructure Composer console.

### Argomenti

- [Home page](#)
- [Designer visivo e tela visiva](#)

### Home page

L'immagine seguente è della home page della console Infrastructure Composer:

The screenshot shows the AWS Infrastructure Composer console home page. On the left is a navigation sidebar with the following items: 'Infrastructure Composer', 'Canvas', 'Documentation', 'AWS Toolkit for VS Code', and 'Import a Lambda Function'. The main content area has a breadcrumb 'Infrastructure Composer > Home' and a 'New!' notification stating 'Infrastructure Composer is now available as part of the AWS Toolkit for Visual Studio Code'. Below the notification are three buttons: 'Download VS Code Extension', 'Open demo', and 'Create project'. The 'About' section describes the tool's purpose and lists key features:
 

- Drag and drop any CloudFormation resource on a visual canvas
- Connect and configure enhanced components to automatically build IaC for an application architecture
- Seamlessly transition between authoring workflows visually with Step Functions Workflow Studio and defining resources with Infrastructure Composer
- Integrate your browser with your project through "local sync" or use Composer in the AWS Toolkit for Visual Studio Code

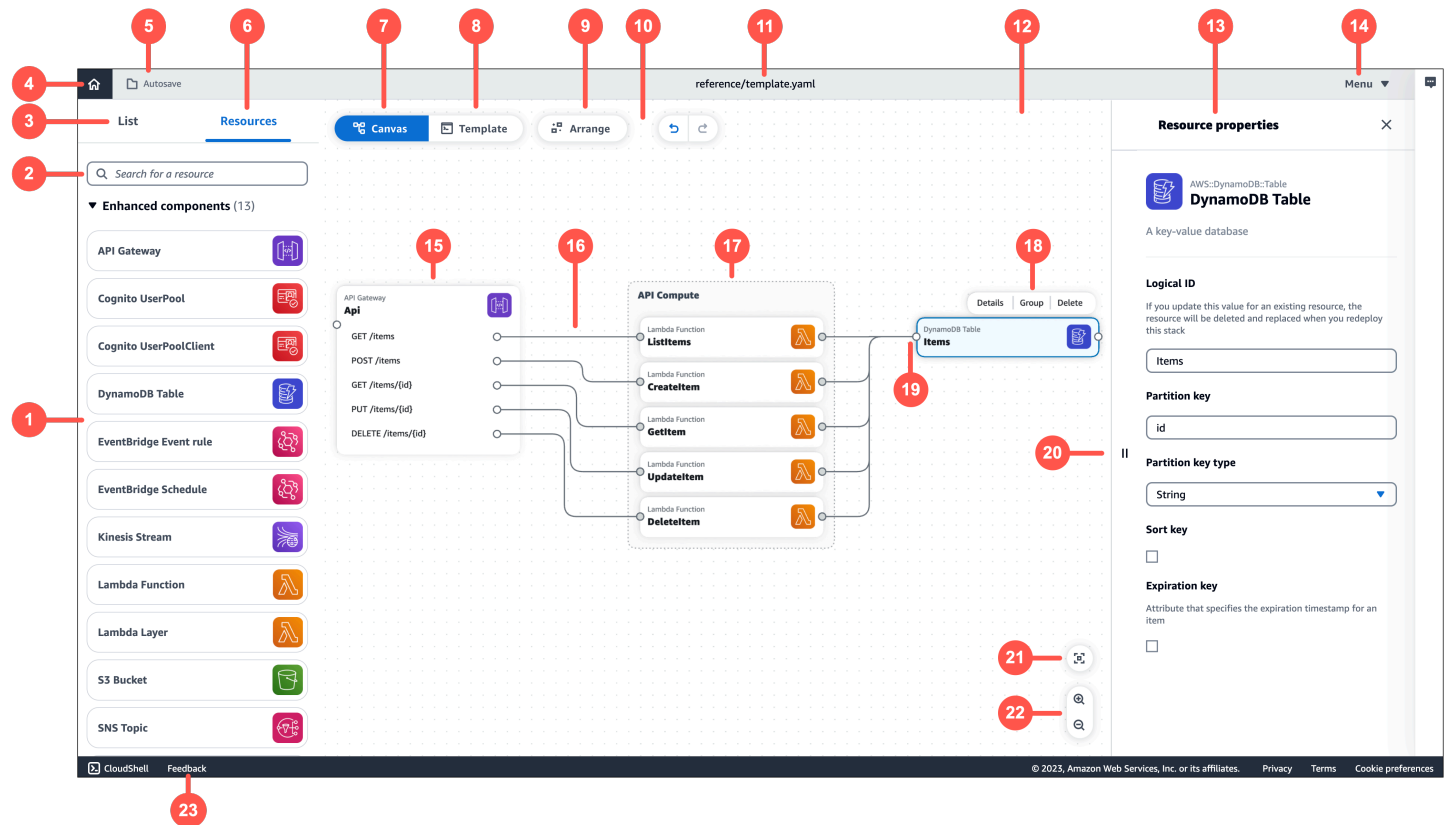
 At the bottom left is a 'Start building' section with a plus icon. On the right is a large visual canvas showing a complex architecture diagram with various AWS services like API Gateway, Lambda, S3, and IAM connected in a flow.

1. Documentazione: vai alla documentazione di Infrastructure Composer.
2. Canvas: vai all'area di disegno e crea o carica un progetto.
3. Demo: apri l'applicazione demo Infrastructure Composer.

4. Crea progetto: crea o carica un progetto.
5. Inizia a creare: collegamenti rapidi per iniziare a creare un'applicazione.
6. Feedback: vai qui per inviare feedback.

## Designer visivo e tela visiva

L'immagine seguente è del visual designer e visual canvas di Infrastructure Composer:



1. Resource palette: mostra le schede con cui è possibile progettare.
2. Barra di ricerca delle risorse: cerca le schede da aggiungere alla tela.
3. Elenco: mostra una visualizzazione ad albero delle risorse dell'applicazione.
4. Home — Seleziona qui per andare alla home page di Infrastructure Composer.
5. Stato di salvataggio: indica se le modifiche di Infrastructure Composer vengono salvate sul computer locale. Gli stati includono:
  - Salvataggio automatico: la sincronizzazione locale è attivata e il progetto viene sincronizzato e salvato automaticamente.
  - Modifiche salvate: il modello di applicazione viene salvato sul computer locale.

- Modifiche non salvate: il modello di applicazione contiene modifiche che non vengono salvate nel computer locale.
6. Risorse: visualizza la tavolozza delle risorse.
  7. Canvas: visualizza la visualizzazione della tela dell'applicazione nell'area di visualizzazione principale.
  8. Modello: visualizza la visualizzazione del modello dell'applicazione nell'area di visualizzazione principale.
  9. Arrange: organizza l'architettura dell'applicazione nell'area di disegno.
  10. Annulla e ripristina: esegue azioni di annullamento e ripristino se supportate.
  11. Nome modello: indica il nome del modello che si sta progettando.
  12. Area di visualizzazione principale: mostra l'area di disegno o il modello in base alla selezione effettuata.
  13. Pannello delle proprietà delle risorse: mostra le proprietà pertinenti per la scheda selezionata nell'area di disegno. Questo pannello è dinamico. Le proprietà visualizzate cambieranno man mano che configuri la scheda.
  14. Menu: fornisce opzioni generali come le seguenti:
    - Creazione di un progetto
    - Aprire un file modello o un progetto
    - Salvare un file modello
    - [Attiva la sincronizzazione locale](#)
    - [Esporta tela](#)
    - Richiedi assistenza
    - Tasti di scelta rapida
  15. Scheda: mostra una vista del biglietto sull'area di disegno.
  16. Linea: rappresenta una connessione tra le carte.
  17. Gruppo: raggruppa le carte selezionate per organizzarle visivamente.
  18. Azioni con le carte: fornisce le azioni che puoi eseguire sulla tua carta.
    - a. Dettagli: visualizza il pannello delle proprietà delle risorse.
    - b. Raggruppa: raggruppa le carte selezionate.
    - c. Elimina: elimina la scheda dall'area di disegno.
  19. Porta: punti di connessione ad altre schede.

- 20. Campi delle proprietà delle risorse: un set curato di campi di proprietà da configurare per le tue schede.
- 21. Ricentra: ricentra il diagramma dell'applicazione nell'area di disegno visiva.
- 22. Zoom: ingrandisci e rimpicciolisci la tela.
- 23. Feedback: vai qui per inviare feedback.

## Gestisci il tuo progetto dalla console Infrastructure Composer

Questo argomento fornisce indicazioni sulle attività di base da eseguire per gestire il progetto dalla console Infrastructure Composer. Ciò include attività comuni come la creazione di un nuovo progetto, il salvataggio di un progetto e l'importazione di un progetto o di un modello. Puoi anche caricare un progetto esistente se attivi la [modalità di sincronizzazione locale](#). Dopo aver attivato la modalità di sincronizzazione locale, puoi fare quanto segue:

- Crea un nuovo progetto costituito da un modello iniziale e da una struttura di cartelle.
- Carica un progetto esistente scegliendo una cartella principale che contiene il modello e i file del progetto.
- Usa Infrastructure Composer per gestire modelli e cartelle

Con la modalità di sincronizzazione locale, Infrastructure Composer salva automaticamente le modifiche al modello e alla cartella del progetto sul computer locale. Se il browser non supporta la modalità di sincronizzazione locale o se si preferisce utilizzare Infrastructure Composer senza attivare la modalità di sincronizzazione locale, è possibile creare un nuovo modello o caricare un modello esistente. Per salvare le modifiche, è necessario esportare il modello sul computer locale.

### Note

Infrastructure Composer supporta applicazioni costituite da quanto segue:

- Un AWS Serverless Application Model modello CloudFormation o che definisce il codice dell'infrastruttura.
- Una struttura di cartelle che organizza i file di progetto, come il codice della funzione Lambda, i file di configurazione e le cartelle di compilazione.

## Argomenti

- [Crea un nuovo progetto nella console Infrastructure Composer](#)
- [Importa una cartella di progetto esistente nella console Infrastructure Composer](#)
- [Importa un modello di progetto esistente nella console Infrastructure Composer](#)
- [Salva un modello di progetto esistente nella console Infrastructure Composer](#)

## Crea un nuovo progetto nella console Infrastructure Composer

Quando crei un nuovo progetto, Infrastructure Composer genera un modello iniziale. Man mano che si progetta l'applicazione sulla tela, il modello viene modificato. Per salvare il lavoro, è necessario esportare il modello o attivare la modalità di sincronizzazione locale.

Per creare un nuovo progetto

1. Accedi alla [console Infrastructure Composer](#).
2. Nella home page, scegli Crea progetto.

### Note


Puoi anche caricarne uno esistente in Infrastructure Composer, ma devi prima [attivare la modalità di sincronizzazione locale](#). Una volta attivata, consulta [Carica un progetto Infrastructure Composer esistente con la sincronizzazione locale attivata](#) per caricare un progetto esistente.

## Importa una cartella di progetto esistente nella console Infrastructure Composer

Utilizzando la modalità di sincronizzazione locale, è possibile importare la cartella principale di un progetto esistente. Se il progetto contiene più modelli, puoi scegliere il modello da caricare.

Per importare un progetto esistente dalla home page

1. Accedi alla [console Infrastructure Composer](#).
2. Nella home page, scegli Carica un CloudFormation modello.
3. Per Posizione del progetto, scegli Seleziona cartella. Seleziona la cartella principale del progetto e scegli Seleziona.


 Note

Se non ricevete questo messaggio, il vostro browser potrebbe non supportare l'API File System Access, necessaria per la modalità di sincronizzazione locale. Per ulteriori informazioni, consulta [Consenti l'accesso alle pagine Web ai file locali in Infrastructure Composer](#).

4. Quando richiesto dal browser, seleziona Visualizza file.
5. Per File modello, scegli il modello dall'elenco a discesa. Se il progetto contiene un solo modello, Infrastructure Composer lo seleziona automaticamente per te.
6. Scegli Create (Crea).

Per importare un progetto esistente dall'area di disegno

1. Dall'area di disegno, scegliete Menu per aprire il menu.
2. Nella sezione Apri, scegli la cartella del progetto.

 Note

Se l'opzione Cartella del progetto non è disponibile, il browser potrebbe non supportare l'API File System Access, necessaria per la modalità di sincronizzazione locale. Per ulteriori informazioni, consulta [Consenti l'accesso alle pagine Web ai file locali in Infrastructure Composer](#).

3. Per Posizione del progetto, scegli Seleziona cartella. Seleziona la cartella principale del progetto e scegli Seleziona.
4. Quando richiesto dal browser, seleziona Visualizza file.
5. Per File modello, scegli il modello dall'elenco a discesa. Se il progetto contiene un solo modello, Infrastructure Composer lo seleziona automaticamente per te.
6. Scegli Create (Crea).

Quando importate una cartella di progetto esistente, Infrastructure Composer attiva la modalità di sincronizzazione locale. Le modifiche apportate al modello o ai file del progetto vengono salvate automaticamente sul computer locale.

## Importa un modello di progetto esistente nella console Infrastructure Composer

Quando si importa un AWS SAM modello CloudFormation o un modello esistente, Infrastructure Composer genera automaticamente una visualizzazione dell'architettura dell'applicazione sull'area di lavoro.

È possibile importare un modello di progetto dal computer locale.

Per importare un modello di progetto esistente

1. Accedi alla [console Infrastructure Composer](#).
2. Scegli Crea progetto per aprire un'area di disegno vuota.
3. Scegli Menu per aprire il menu.
4. Nella sezione Apri, scegli File modello.
5. Seleziona il modello e scegli Apri.

Per salvare le modifiche al modello, devi esportare il modello o attivare la modalità di sincronizzazione locale.

## Salva un modello di progetto esistente nella console Infrastructure Composer

Se non utilizzi la modalità di sincronizzazione locale, devi esportare il modello per salvare le modifiche. Se la modalità di sincronizzazione locale è attivata, non è necessario salvare manualmente il modello. Le modifiche vengono salvate automaticamente sul computer locale.

Per salvare un modello di progetto esistente

1. Dalla tela di Infrastructure Composer, scegliete Menu per aprire il menu.
2. Nella sezione Salva, scegli Salva il file modello.
3. Fornisci un nome per il tuo modello.
4. Seleziona una posizione in cui salvare il modello.
5. Scegli Save (Salva).

## Connect la console Infrastructure Composer all'IDE locale

Per connettere la console Infrastructure Composer all'ambiente di sviluppo integrato (IDE) locale, utilizza la modalità di sincronizzazione locale. Questa modalità sincronizza e salva automaticamente

i dati sul computer locale. Per ulteriori informazioni sulla modalità di sincronizzazione locale, vedere [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#). Per istruzioni sull'uso della modalità di sincronizzazione locale, vedere [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#).

### Note

L'opzione Attiva sincronizzazione locale non è disponibile in tutti i browser. È disponibile in Google Chrome e Microsoft Edge.

## Vantaggi dell'utilizzo di Infrastructure Composer con l'IDE locale

Durante la progettazione in Infrastructure Composer, il modello locale e la directory del progetto vengono sincronizzati e salvati automaticamente.

È possibile utilizzare l'IDE locale per visualizzare le modifiche e modificare i modelli. Le modifiche apportate localmente vengono sincronizzate automaticamente con Infrastructure Composer.

È possibile utilizzare strumenti locali come l'interfaccia a riga di AWS Serverless Application Model comando (AWS SAM CLI) per creare, testare, distribuire l'applicazione e altro ancora. L'esempio seguente mostra come trascinare le risorse sulla tela visiva di Infrastructure Composer che, a sua volta, crea il markup nel AWS SAM modello nell'IDE locale.

The screenshot displays the Infrastructure Composer IDE interface. On the left, a 'Resources' panel lists various AWS services like DynamoDB, EventBridge, Kinesis, Lambda, and S3. The central 'Canvas' shows a visual diagram where an 'S3 Bucket' resource is connected to a 'Lambda Function' resource. On the right, the 'EXPLORER' panel shows the 'template.yaml' file with the following YAML content:

```

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

## Integra Infrastructure Composer con il tuo IDE locale

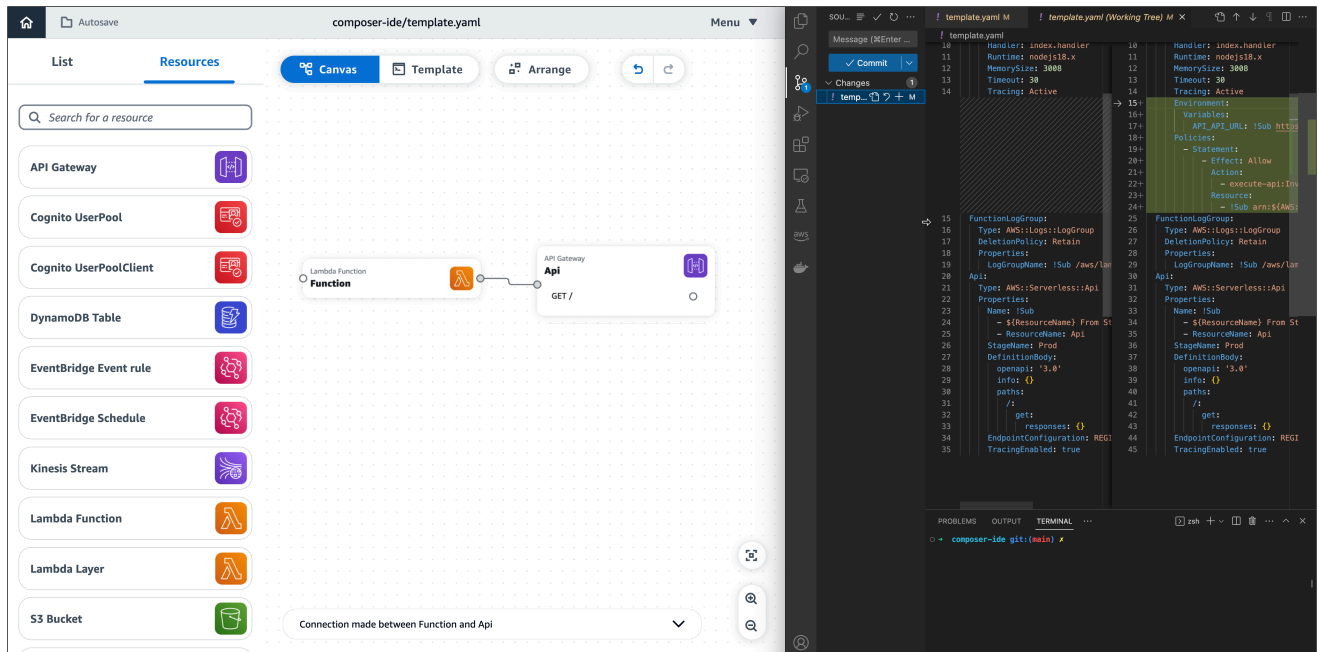
Per integrare Infrastructure Composer con il tuo IDE locale

1. In Infrastructure Composer, crea o carica un progetto e attiva la sincronizzazione locale selezionando il pulsante Menu nella parte superiore destra dello schermo e scegliendo Attiva sincronizzazione locale.

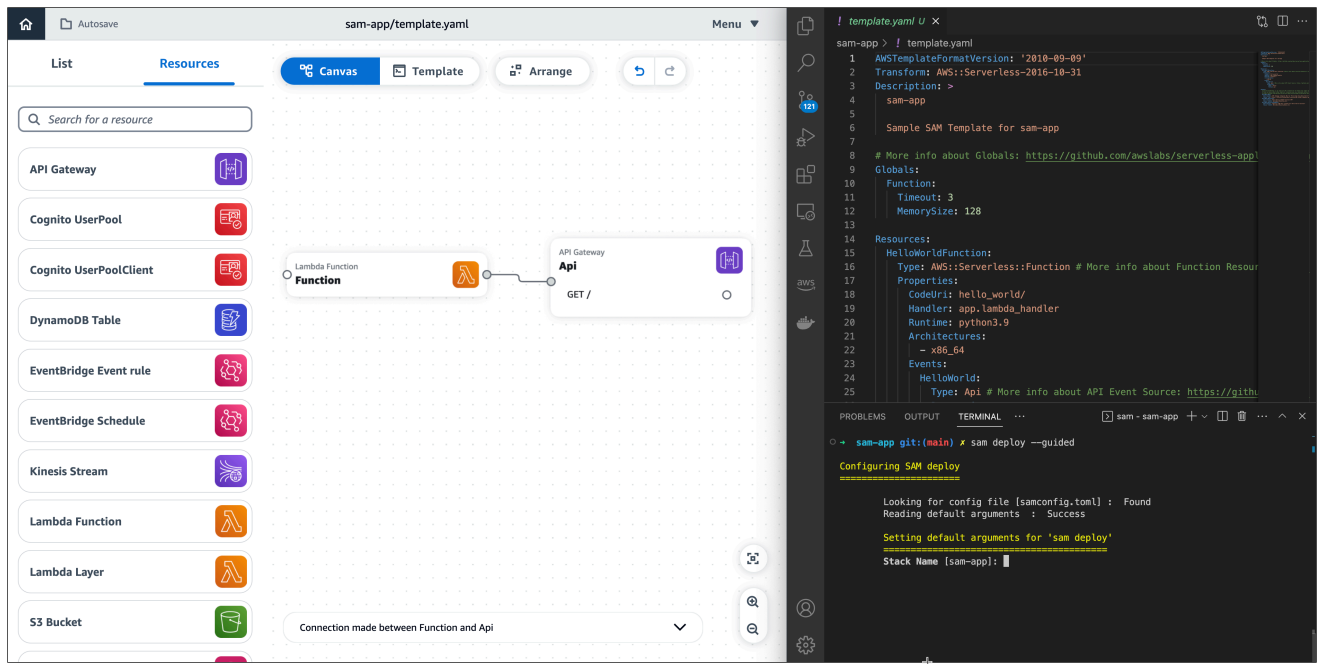
### Note

L'opzione Attiva sincronizzazione locale non è disponibile in tutti i browser. È disponibile in Google Chrome e Microsoft Edge.

2. Nel tuo IDE locale, apri la stessa cartella di progetto di Infrastructure Composer.
3. Usa Infrastructure Composer con il tuo IDE locale. Gli aggiornamenti effettuati in Infrastructure Composer verranno sincronizzati automaticamente con il computer locale. Ecco alcuni esempi di cosa puoi fare:
  - a. Utilizzate il vostro sistema di controllo delle versioni preferito per tenere traccia degli aggiornamenti eseguiti da Infrastructure Composer.



- b. Usa la AWS SAM CLI localmente per creare, testare, distribuire la tua applicazione e altro ancora. Per ulteriori informazioni, consulta [Distribuisci la tua applicazione serverless Infrastructure Composer nel cloud AWS.](#)



## Consenti l'accesso alle pagine Web ai file locali in Infrastructure Composer

La console Infrastructure Composer supporta la [modalità di sincronizzazione locale](#) e [le funzioni di importazione dalla console Lambda](#). Per utilizzare queste funzionalità, è necessario un browser Web che supporti l'API File System Access. Qualsiasi versione recente di Google Chrome e Microsoft Edge supporta tutte le funzionalità dell'API File System Access e può essere utilizzata con la modalità di sincronizzazione locale in Infrastructure Composer.

L'API File System Access consente alle pagine Web di accedere al file system locale per leggere, scrivere o salvare file. Questa funzionalità è disattivata per impostazione predefinita e richiede l'autorizzazione dell'utente tramite un prompt visivo per consentirla. Una volta concesso, questo accesso rimane valido per tutta la durata della sessione del browser della pagina Web.

Per ulteriori informazioni sull'API File System Access, consulta:

- [API File System Access](#) nei documenti web di mdn.
- [L'API File System Access: semplifica l'accesso ai file locali](#) nel sito web.dev.

## modalità di sincronizzazione locale

La modalità di sincronizzazione locale consente di sincronizzare e salvare automaticamente i file modello e le cartelle di progetto localmente durante la progettazione in Infrastructure Composer. Per utilizzare questa funzionalità, è necessario un browser Web che supporti l'API File System Access.

### Data Infrastructure Composer ottiene l'accesso a

Infrastructure Composer ottiene l'accesso in lettura e scrittura alla cartella di progetto consentita, insieme a tutte le cartelle secondarie di quella cartella di progetto. Questo accesso viene utilizzato per creare, aggiornare e salvare qualsiasi file modello, cartella di progetto e directory di backup generati durante la progettazione. I dati a cui accede Infrastructure Composer non vengono utilizzati per altri scopi e non vengono archiviati al di fuori del file system locale.

### Accesso a dati sensibili

L'API File System Access esclude o limita l'accesso a directory specifiche che possono contenere dati sensibili. Si verificherà un errore se si seleziona una di queste directory da utilizzare con la modalità di sincronizzazione locale di Infrastructure Composer. È possibile scegliere un'altra directory locale a cui connettersi o utilizzare Infrastructure Composer nella modalità predefinita con la sincronizzazione locale disattivata.

Per ulteriori informazioni, inclusi esempi di directory riservate, consultate [Users conceder access a più o più file sensibili di quanto previsto nel File System Access W3C Draft Community Group Report](#).

Se si utilizza Windows Subsystem for Linux (WSL), l'API File System Access esclude l'accesso all'intera Linux directory a causa della sua posizione all'interno del sistema. Windows È possibile utilizzare Infrastructure Composer con la sincronizzazione locale disattivata o configurare una soluzione per sincronizzare i file di progetto dalla directory a una WSL directory di lavoro in Windows. Quindi, utilizzate la modalità di sincronizzazione locale di Infrastructure Composer con la directory Windows.

## Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer

Questa sezione fornisce informazioni sull'utilizzo della modalità di sincronizzazione locale di Infrastructure Composer per sincronizzare e salvare automaticamente il progetto sul computer locale.

Ti consigliamo di utilizzare la sincronizzazione locale per i seguenti motivi:

Puoi attivare la sincronizzazione locale per un nuovo progetto o caricare un progetto esistente con la sincronizzazione locale attivata.

- Per impostazione predefinita, è necessario salvare manualmente il modello di applicazione durante la progettazione. Utilizzate la sincronizzazione locale per salvare automaticamente il modello di applicazione sul computer locale quando apportate le modifiche.
- La sincronizzazione locale gestisce e sincronizza automaticamente le cartelle di progetto, la cartella di backup e [i file esterni supportati](#) sul computer locale.
- Quando si utilizza la sincronizzazione locale, è possibile connettere Infrastructure Composer all'IDE locale per accelerare lo sviluppo. Per ulteriori informazioni, consulta [Connect la console Infrastructure Composer all'IDE locale](#).

## Cosa salva la modalità di sincronizzazione locale

La modalità di sincronizzazione locale sincronizza e salva automaticamente quanto segue sul computer locale:

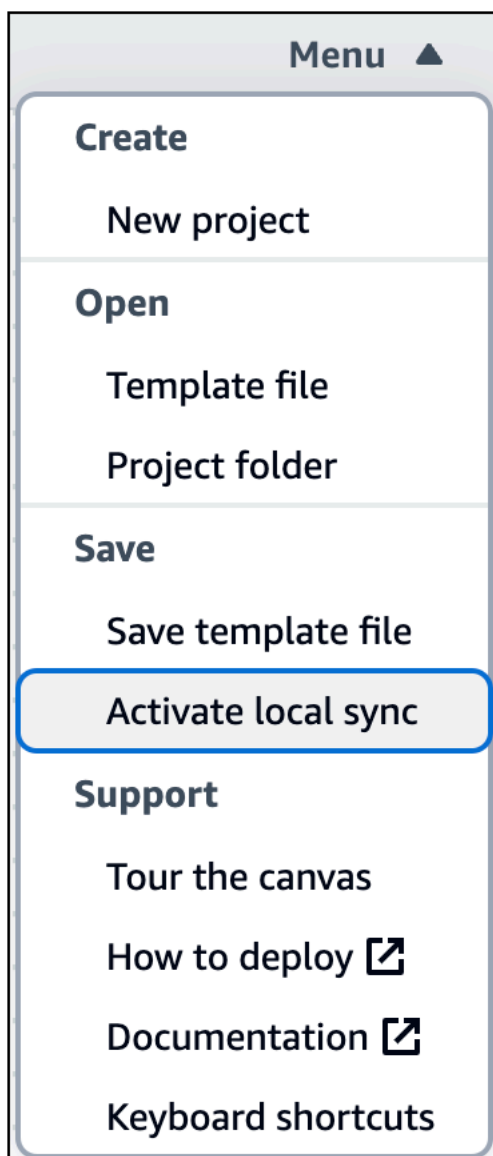
- File modello di applicazione: il modello AWS CloudFormation or AWS Serverless Application Model (AWS SAM) che contiene l'infrastruttura come codice (IaC).
- Cartelle di progetto: una struttura di directory generale che organizza le AWS Lambda funzioni.
- Directory di backup: una directory di backup denominata `.aws-composer`, creata nella radice della posizione del progetto. Questa directory contiene una copia di backup del file modello dell'applicazione e delle cartelle di progetto.
- File esterni: file esterni supportati che è possibile utilizzare in Infrastructure Composer. Per ulteriori informazioni, consulta [Fai riferimento ai file esterni in Infrastructure Composer](#).

## Requisiti browser

La modalità di sincronizzazione locale richiede un browser che supporti l'API File System Access. Per ulteriori informazioni, consulta [Consenti l'accesso alle pagine Web ai file locali in Infrastructure Composer](#).

## Attivazione della modalità di sincronizzazione locale

La modalità di sincronizzazione locale è disattivata per impostazione predefinita. È possibile attivare la modalità di sincronizzazione locale tramite il menu Infrastructure Composer.



Per istruzioni sull'attivazione della sincronizzazione locale e sui progetti di caricamento esistenti, consulta i seguenti argomenti:


- [Attiva la sincronizzazione locale in Infrastructure Composer](#)
- [Carica un progetto Infrastructure Composer esistente con la sincronizzazione locale attivata](#)

## Attiva la sincronizzazione locale in Infrastructure Composer

Per attivare la sincronizzazione locale, completa i seguenti passaggi:

1. Dalla [home](#) page di Infrastructure Composer, seleziona Crea progetto.

2. Dal menu Infrastructure Composer, seleziona Attiva la sincronizzazione locale.
3. Per la posizione del progetto, premi Seleziona cartella e scegli una directory. Qui è dove Infrastructure Composer salverà e sincronizzerà i file e le cartelle del modello durante la progettazione.

 Note

La posizione del progetto non deve contenere un modello di applicazione esistente.

4. Quando viene richiesto di consentire l'accesso, selezionate Visualizza file.
5. Premi Attiva. Quando viene richiesto di salvare le modifiche, seleziona Salva modifiche.

Quando attivato, l'indicatore Autosave verrà visualizzato nell'area in alto a sinistra dell'area di disegno.

## Carica un progetto Infrastructure Composer esistente con la sincronizzazione locale attivata

Per caricare un progetto esistente con la sincronizzazione locale attivata, completa i seguenti passaggi:

1. Dalla [home](#) page di Infrastructure Composer, seleziona Carica un CloudFormation modello.
2. Dal menu Infrastructure Composer, selezionate Apri > Cartella del progetto.
3. Per localizzare il progetto, premete Seleziona cartella e scegliete la cartella principale del progetto.
4. Quando viene richiesto di consentire l'accesso, seleziona Visualizza file.
5. Per File modello, seleziona il modello dell'applicazione e premi Crea.
6. Quando viene richiesto di salvare le modifiche, seleziona Salva modifiche.

Quando attivato, l'indicatore Autosave verrà visualizzato nell'area in alto a sinistra dell'area di disegno.

## Importa le funzioni in Infrastructure Composer dalla console Lambda

Infrastructure Composer fornisce un'integrazione con la AWS Lambda console. È possibile importare una funzione Lambda dalla console Lambda nella console Infrastructure Composer. Quindi, usa il canvas di Infrastructure Composer per progettare ulteriormente l'architettura dell'applicazione.

- Questa integrazione richiede un browser che supporti l'API File System Access. Per ulteriori informazioni, consulta [Consenti l'accesso alle pagine Web ai file locali in Infrastructure Composer](#).
- Quando importi la funzione Lambda in Infrastructure Composer, devi attivare la modalità di sincronizzazione locale per salvare eventuali modifiche. Per ulteriori informazioni, consulta [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#).

Per iniziare a utilizzare questa integrazione, consulta [Using AWS Lambda with AWS Infrastructure Composer](#) nella AWS Lambda Developer Guide.

## Esporta un'immagine della tela visiva di Infrastructure Composer

Questo argomento descrive la funzionalità di esportazione della AWS Infrastructure Composer console su tela.

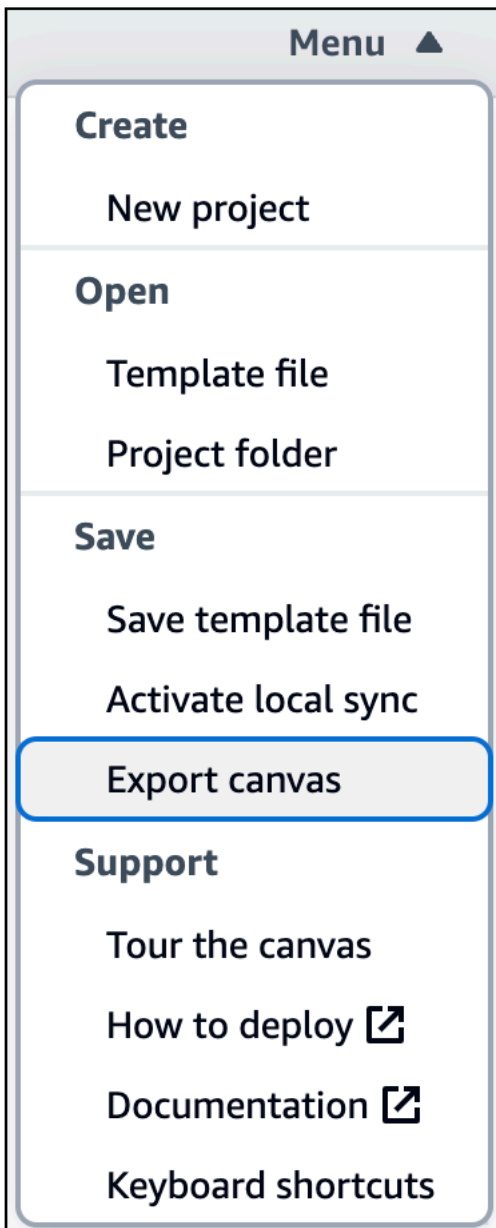
Per una panoramica visiva di tutte le funzionalità di Infrastructure Composer, consulta [AWS Infrastructure Composer panoramica visiva della console](#).

### Informazioni su Export Canvas

La funzione di esportazione della tela esporta la tela dell'applicazione come immagine sul computer locale.

- Infrastructure Composer rimuove gli elementi dell'interfaccia utente di Visual Designer ed esporta solo il diagramma dell'applicazione.
- Il formato di file di immagine predefinito è .png
- Il file viene esportato nella posizione di download predefinita del computer locale.

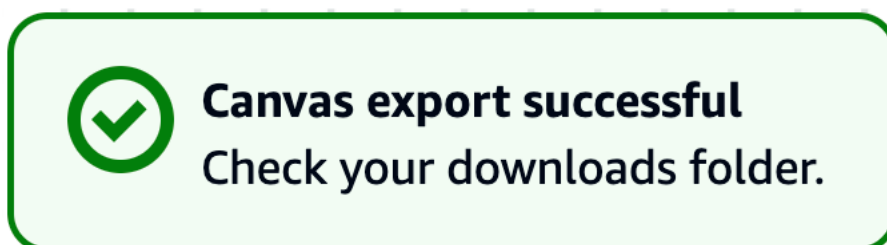
Puoi accedere alla funzione di esportazione della tela dal menu.



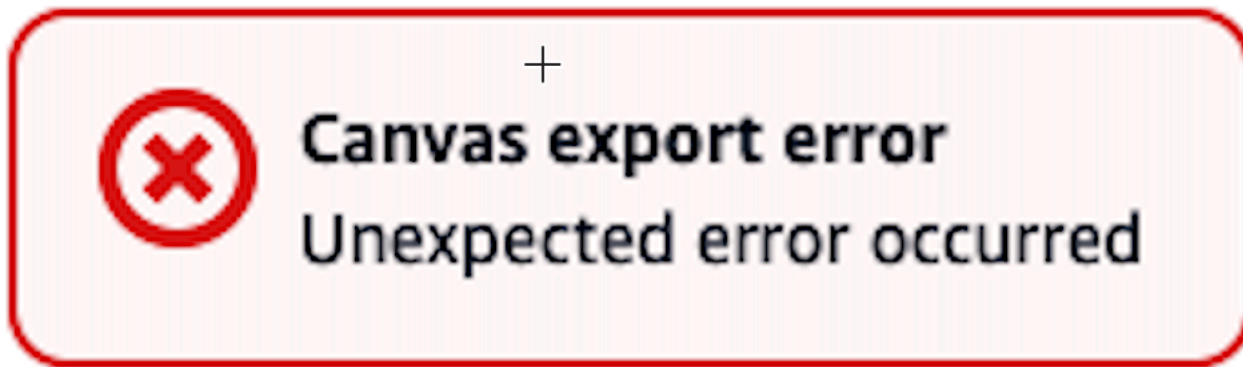
## Esportazione della tela

Quando esporti il tuo canvas, Infrastructure Composer visualizza un messaggio di stato.

Se l'esportazione ha esito positivo, verrà visualizzato il seguente messaggio:



Se l'esportazione non è riuscita, verrà visualizzato un messaggio di errore. Se ricevi un errore, prova a esportare di nuovo.



## Utilizzo di Infrastructure Composer in modalità CloudFormation console

Infrastructure Composer in modalità CloudFormation console è lo strumento consigliato per visualizzare i modelli. CloudFormation Puoi utilizzare questo strumento anche per creare e modificare CloudFormation modelli.

### In che modo questa modalità è diversa dalla console di Infrastructure Composer?

Infrastructure Composer in modalità CloudFormation console offre in genere le stesse funzionalità della [console Infrastructure Composer predefinita](#), ma ci sono alcune differenze da notare.

- Questa modalità è integrata con il flusso di lavoro dello stack nella console. CloudFormation Ciò consente di utilizzare Infrastructure Composer direttamente in. CloudFormation
- [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#), una funzionalità che sincronizza e salva automaticamente i dati sul computer locale, non è supportata.
- Le schede relative a Lambda (Lambda Function e Lambda Layer) richiedono build di codice e soluzioni di packaging che non sono disponibili in questa modalità.

#### Note

[Queste schede e la sincronizzazione locale possono essere utilizzate nella console Infrastructure Composer o in. AWS Toolkit for Visual Studio Code](#)

Quando si apre Infrastructure Composer dalla CloudFormation console, Infrastructure Composer si apre in CloudFormation modalità console. In questa modalità, puoi utilizzare Infrastructure Composer per visualizzare, creare e aggiornare i tuoi modelli.

## Come accedere a Infrastructure Composer in modalità console CloudFormation

Infrastructure Composer in modalità CloudFormation console è un aggiornamento di CloudFormation Designer. Ti consigliamo di utilizzare Infrastructure Composer per visualizzare i tuoi modelli. CloudFormation Puoi utilizzare questo strumento anche per creare e modificare CloudFormation modelli.

1. Vai alla [console Cloudformation](#) e accedi.
2. Seleziona Infrastructure Composer dal menu di navigazione a sinistra. Questo ti porterà a Infrastructure Composer in modalità console. CloudFormation

### Note

Per informazioni sull'utilizzo di Infrastructure Composer in modalità CloudFormation console, consulta. [Utilizzo di Infrastructure Composer in modalità CloudFormation console](#)

## Visualizza una distribuzione in Infrastructure Composer in modalità console CloudFormation

Segui le istruzioni riportate in questo argomento per visualizzare un modello di CloudFormation stack/ Infrastructure Composer distribuito.

1. [Vai alla console e accedi.CloudFormation](#)
2. Seleziona lo stack che desideri modificare.
3. Seleziona la scheda Modello.
4. Seleziona Infrastructure Composer.

Infrastructure Composer visualizzerà il tuo stack/modello. Le modifiche possono essere apportate anche qui.

## Crea un nuovo modello in Infrastructure Composer in modalità CloudFormation console

Segui le istruzioni riportate in questo argomento per creare un nuovo modello.

1. Vai alla [CloudFormation console](#) e accedi.
2. Seleziona Infrastructure Composer dal menu di navigazione a sinistra. Questo aprirà Infrastructure Composer in modalità console. CloudFormation
3. Trascina, rilascia, configura e collega le risorse ([schede](#)) di cui hai bisogno dal palette Risorse.

### Note

Vedi [Come comporre](#) i dettagli sull'uso di Infrastructure Composer e tieni presente che le schede relative a Lambda (Lambda Function e Lambda Layer) richiedono build di codice e soluzioni di packaging che non sono disponibili in Infrastructure Composer in modalità console. CloudFormation [Queste schede possono essere utilizzate nella console Infrastructure Composer o in](#) AWS Toolkit for Visual Studio Code Per informazioni sull'utilizzo di questi strumenti, fare riferimento a [Dove puoi usare Infrastructure Composer](#).

4. Fate doppio clic sulle schede per utilizzare il pannello delle proprietà delle risorse e specificare come sono configurate le schede.
5. [Connect le schede](#) per specificare il flusso di lavoro basato sugli eventi dell'applicazione.
6. Seleziona Modello per visualizzare e modificare il codice dell'infrastruttura. Le modifiche vengono sincronizzate automaticamente con la visualizzazione dell'area di disegno.
7. Quando il modello è pronto per essere esportato in una pila, seleziona Crea modello.
8. Seleziona il pulsante Conferma ed esporta in. CloudFormation In questo modo tornerai al flusso di lavoro di creazione dello stack con un messaggio di conferma che il modello è stato importato correttamente.

### Note

È possibile esportare solo i modelli che contengono risorse.

9. Nel flusso di lavoro Create stack, seleziona Avanti.
10. Fornisci un nome per lo stack, esamina tutti i parametri elencati e seleziona Avanti.

**Note**

Il nome dello stack deve iniziare con una lettera e contenere solo lettere, numeri e trattini.

11. Seleziona Avanti dopo aver fornito le seguenti informazioni:

- Tag associati allo stack
- Autorizzazioni dello stack
- Le opzioni di errore dello stack

**Note**

Per indicazioni sulla gestione degli stack, consulta [le CloudFormation best practice](#) nella Guida per l'CloudFormation utente.

12. Verifica che i dettagli dello stack siano corretti, controlla i riconoscimenti in fondo alla pagina e seleziona il pulsante Invia.

CloudFormation inizierà a creare lo stack in base ai dati del modello.

## Aggiorna uno stack esistente in Infrastructure Composer in modalità console CloudFormation

Segui le istruzioni riportate in questo argomento per aggiornare uno CloudFormation stack esistente.

**Note**

Se il file viene salvato localmente, ti consigliamo di utilizzare [AWS Toolkit for Visual Studio Code](#).

1. Vai alla [CloudFormation console](#) e accedi.
2. Seleziona lo stack che desideri modificare.
3. Seleziona il pulsante Aggiorna. In questo modo verrai indirizzato alla procedura guidata di aggiornamento dello stack.

4. A destra, seleziona Modifica in Infrastructure Composer.
5. Seleziona il pulsante sotto denominato Modifica in Infrastructure Composer. Questo ti porterà a Infrastructure Composer in CloudFormation modalità console.
6. Qui puoi trascinare, rilasciare, configurare e connettere le risorse ([schede](#)) dal pannello Risorse.

#### Note

Vedi [Come comporre](#) i dettagli sull'uso di Infrastructure Composer e tieni presente che le schede relative a Lambda (Lambda Function e Lambda Layer) richiedono build di codice e soluzioni di packaging che non sono disponibili in Infrastructure Composer in modalità console. CloudFormation [Queste schede possono essere utilizzate nella console Infrastructure Composer o in](#). AWS Toolkit for Visual Studio Code Per informazioni sull'utilizzo di questi strumenti, fare riferimento a [Dove puoi usare Infrastructure Composer](#).

7. Quando sei pronto per esportare le modifiche in CloudFormation, seleziona Aggiorna modello.
8. Seleziona Conferma e continua con CloudFormation. In questo modo tornerai al flusso di lavoro Update stack con un messaggio di conferma che il modello è stato importato correttamente.

#### Note

È possibile esportare solo i modelli che contengono risorse.

9. Nel flusso di lavoro Update stack, selezionate Avanti.
10. Esamina tutti i parametri elencati e seleziona Avanti.
11. Seleziona Avanti dopo aver fornito le seguenti informazioni:
  - Tag associati allo stack
  - Autorizzazioni dello stack
  - Le opzioni di errore dello stack

#### Note

Per indicazioni sulla gestione degli stack, consulta [le CloudFormation best practice](#) nella Guida per l'CloudFormation utente.

12. Verifica che i dettagli dello stack siano corretti, controlla i riconoscimenti in fondo alla pagina e seleziona il pulsante Invia.

CloudFormation inizierà ad aggiornare lo stack in base agli aggiornamenti apportati al modello.

## Utilizzo di Infrastructure Composer di AWS Toolkit for Visual Studio Code

Questa sezione descrive come utilizzare AWS Infrastructure Composer from. [AWS Toolkit for Visual Studio Code](#) Ciò include una panoramica visiva di Infrastructure Composer fornita da. AWS Toolkit for Visual Studio Code Include anche istruzioni che mostrano come accedere a questa esperienza e sincronizzare il progetto da VS Code al AWS cloud. Per la sincronizzazione, si utilizza il sam sync comando di AWS SAMCLI. Questa sezione fornisce inoltre indicazioni sull'utilizzo Amazon Q di Infrastructure Composer da. AWS Toolkit for Visual Studio Code

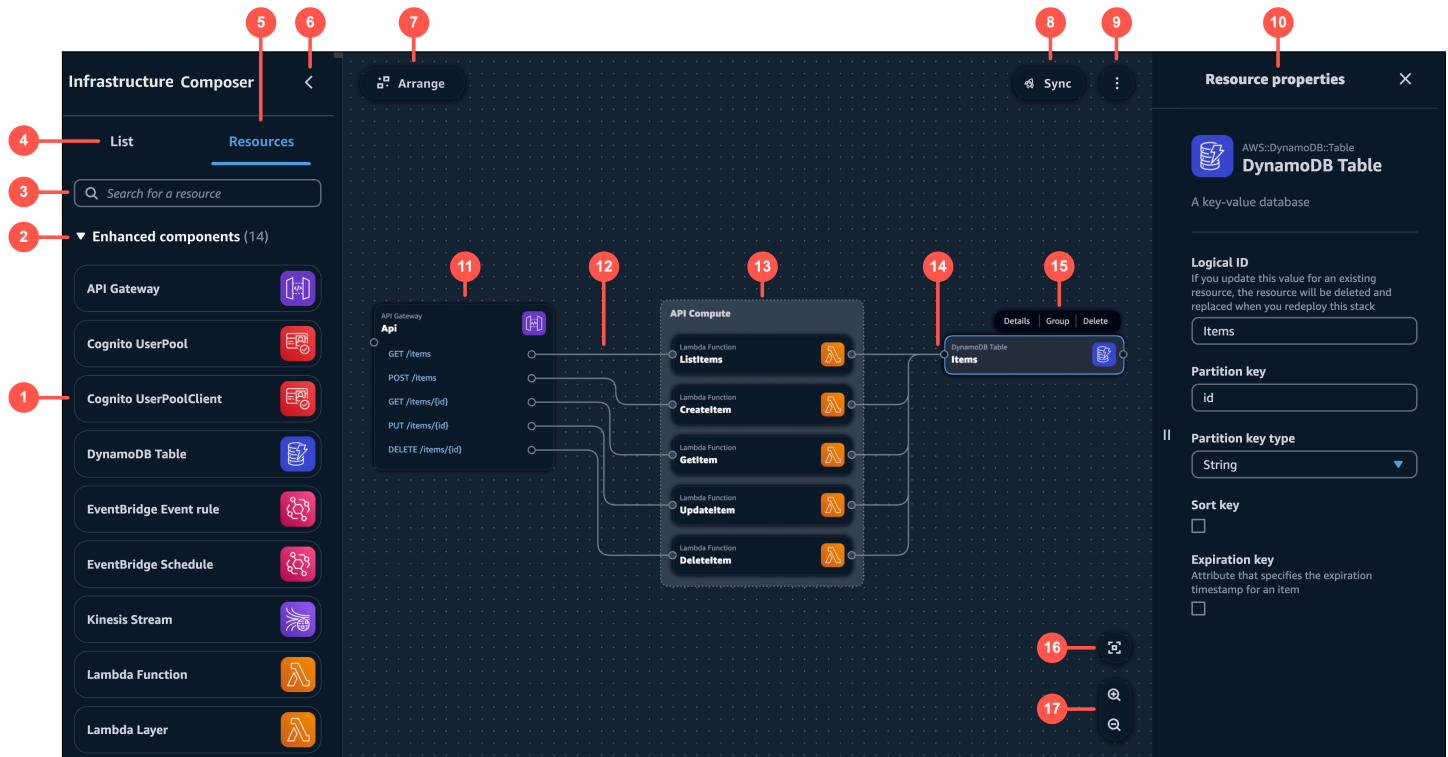
Per ulteriori indicazioni sull'utilizzo di Infrastructure Composer di AWS Toolkit for Visual Studio Code, fare riferimento a. [Come comporre](#) Il contenuto di questa sezione si applica a questa esperienza, oltre che all'esperienza della console Infrastructure Composer.

### Argomenti

- [Panoramica visiva di Infrastructure Composer dal AWS Toolkit for Visual Studio Code](#)
- [Accedi a Infrastructure Composer dal AWS Toolkit for Visual Studio Code](#)
- [Sync Infrastructure Composer per la distribuzione su Cloud AWS](#)
- [Utilizzo AWS Infrastructure Composer con Amazon Q Developer](#)

## Panoramica visiva di Infrastructure Composer dal AWS Toolkit for Visual Studio Code

Il visual designer di Infrastructure Composer AWS Toolkit for Visual Studio Code include una tela visiva, che include componenti numerati nell'immagine seguente ed elencati di seguito.



1. Resource Palette: mostra le schede con cui è possibile progettare.
2. Categorie di schede: le schede sono organizzate per categorie esclusive di Infrastructure Composer.
3. Barra di ricerca delle risorse: cerca le schede da aggiungere all'area di disegno.
4. Elenco: mostra una visualizzazione ad albero delle risorse dell'applicazione.
5. Risorse: visualizza la tavolozza delle risorse.
6. Attiva/disattiva il riquadro sinistro: nasconde o mostra il riquadro sinistro.
7. Arrange: organizza l'architettura dell'applicazione nell'area di disegno.
8. Sincronizza: avvia il CLI `sam sync` comando AWS Serverless Application Model (AWS SAM) per distribuire l'applicazione.
9. Menu: fornisce opzioni generali come le seguenti:
  - Esporta tela
  - Visita la tela
  - Collegamenti alla documentazione
  - Tasti di scelta rapida

10 Pannello delle proprietà delle risorse: mostra le proprietà pertinenti per la scheda selezionata nell'area di disegno. Questo pannello è dinamico. Le proprietà visualizzate cambieranno man mano che configuri la scheda.

11 Scheda: mostra una vista della scheda sulla tela.

12 Linea: rappresenta una connessione tra le carte.

13 Gruppo: un gruppo di carte. È possibile raggruppare le carte per organizzarle visivamente.

14 Porta: punti di connessione ad altre schede.

15 Azioni relative alla carta: fornisce le azioni che puoi eseguire sulla tua carta.

- Dettagli: visualizza il pannello delle proprietà della risorsa.
- Gruppo: raggruppa le carte selezionate.
- Elimina: elimina la scheda dalla tela e dal modello.

16 Ricentra: ricentra il diagramma dell'applicazione sull'area di disegno visiva.

17 Zoom: ingrandisci e rimpicciolisci la tela.

## Accedi a Infrastructure Composer dal AWS Toolkit for Visual Studio Code

Segui le istruzioni riportate in questo argomento per accedere a Infrastructure Composer da AWS Toolkit for Visual Studio Code

### Note

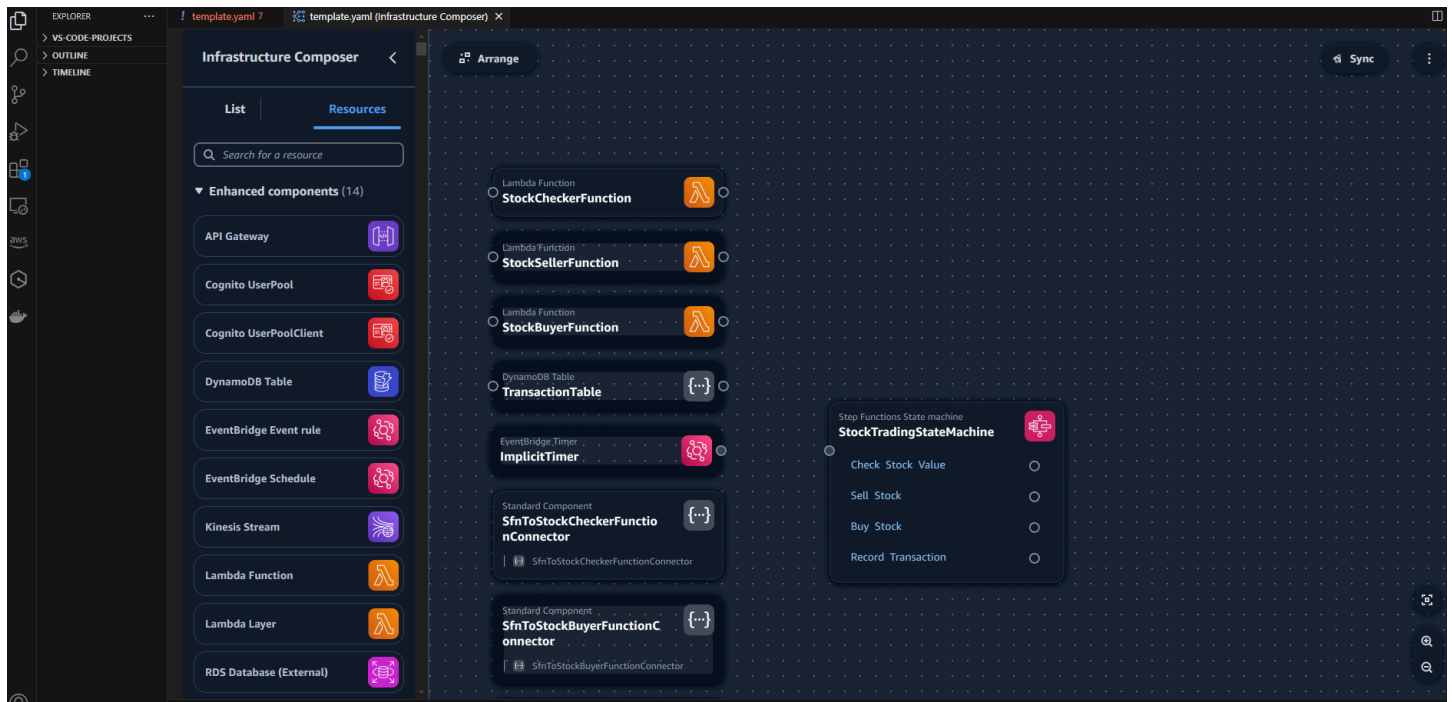
Prima di poter accedere a Infrastructure Composer da AWS Toolkit for Visual Studio Code, devi prima scaricare e installare il Toolkit for VS Code. Per istruzioni, consulta [Download del Toolkit for VS Code](#).

Per accedere a Infrastructure Composer dal Toolkit for VS Code

È possibile accedere a Infrastructure Composer in uno dei seguenti modi:

1. Selezionando il pulsante Infrastructure Composer da CloudFormation qualsiasi AWS SAM modello.
2. Tramite il menu contestuale facendo clic con il pulsante destro del mouse sul modello CloudFormation o sul modello AWS SAM .
3. Dalla palette dei comandi VS Code.

Di seguito è riportato un esempio di accesso a Infrastructure Composer dal pulsante Infrastructure Composer:



Per ulteriori informazioni sull'accesso a Infrastructure Composer, vedere [Accesso AWS Infrastructure Composer dal Toolkit](#).

## Sync Infrastructure Composer per la distribuzione su Cloud AWS

Usa il pulsante di sincronizzazione AWS Infrastructure Composer di AWS Toolkit for Visual Studio Code per distribuire la tua applicazione su Cloud AWS.

Il pulsante di sincronizzazione avvia il `sam sync` comando dall'interfaccia a riga di comando AWS SAM (`aws sam sync`).

Il `sam sync` comando può distribuire nuove applicazioni o sincronizzare rapidamente le modifiche apportate localmente a Cloud AWS. L'esecuzione `sam sync` può includere quanto segue:

- Creazione dell'applicazione con `sam build` cui preparare i file dell'applicazione locale per la distribuzione creando o aggiornando una `.aws-sam` directory locale.
- Per le risorse che supportano il servizio AWS APIs, AWS SAM CLI utilizzeranno le APIs per distribuire le modifiche. Lo AWS SAM CLI fa per aggiornare rapidamente le risorse nel cloud.
- Se necessario, AWS SAM CLI esegue una AWS CloudFormation distribuzione per aggiornare l'intero stack tramite un set di modifiche.

Il `sam sync` comando è più adatto per ambienti di sviluppo rapido quando l'aggiornamento rapido delle risorse cloud può favorire i flussi di lavoro di sviluppo e test.

Per ulteriori informazioni su `sam sync`, consulta [Using sam sync](#) nella AWS Serverless Application Model Developer Guide.

## Configurazione

Per utilizzare la funzionalità di sincronizzazione in Infrastructure Composer, è necessario averla AWS SAM CLI installata sul computer locale. Per istruzioni, consulta [Installazione di AWS SAMCLI nella Guida per gli AWS Serverless Application Model sviluppatori](#).

Quando si utilizza la funzionalità di sincronizzazione in Infrastructure Composer, AWS SAM CLI fa riferimento al file di configurazione per ottenere le informazioni necessarie per sincronizzare l' Cloud AWS applicazione con. Per istruzioni sulla creazione, la modifica e l'utilizzo dei file di configurazione, consulta [Configurare le impostazioni del progetto nella Guida](#) per gli AWS Serverless Application Model sviluppatori.

## Sincronizza e distribuisci la tua applicazione

Per sincronizzare l'applicazione con Cloud AWS

1. Seleziona il pulsante di sincronizzazione nell'area di disegno di Infrastructure Composer.
2. È possibile che venga richiesto di confermare che si sta lavorando con uno stack di sviluppo. Seleziona OK per continuare.
3. Infrastructure Composer potrebbe richiedere all'utente di configurare le seguenti opzioni:
  - Regione AWS— La regione con cui sincronizzare l'applicazione.
  - CloudFormation nome dello stack: il nome del tuo CloudFormation stack. Puoi selezionare il nome di uno stack esistente o crearne uno nuovo.
  - Bucket Amazon Simple Storage Service (Amazon S3): il nome del bucket Amazon S3. Qui AWS SAM CLI verranno impacchettati e archiviati i file dell'applicazione e il codice della funzione. È possibile selezionare un bucket esistente o crearne uno nuovo.

Infrastructure Composer avvierà il AWS SAM CLI `sam sync` comando e aprirà una finestra di terminale nell'IDE per visualizzarne l'avanzamento.

## Utilizzo AWS Infrastructure Composer con Amazon Q Developer

AWS Infrastructure Composer da AWS Toolkit for Visual Studio Code fornisce un'integrazione con Amazon Q. Puoi utilizzarlo Amazon Q all'interno di Infrastructure Composer per generare il codice di infrastruttura per AWS le tue risorse durante la progettazione dell'applicazione.

Amazon Q è un generatore di codice generico basato sull'apprendimento automatico. [Per ulteriori informazioni, vedi Cos'è? Amazon Q](#) nella Guida Amazon Q Developer per l'utente.

Per le schede di risorse e componenti standard, puoi utilizzarle Amazon Q per generare suggerimenti sul codice di infrastruttura per le tue risorse.

The screenshot displays the AWS Infrastructure Composer interface. On the left, a grid of resource cards is shown. A card for a 'Standard Component VPC' is highlighted with a blue border. The card includes a 'VPC' icon and a 'VPC' label. Above the card are buttons for 'Details', 'Group', and 'Delete'. A 'Sync' button is visible at the top of the grid. On the right, the 'Resource properties' panel is open for the selected VPC resource. The panel shows the resource type 'AWS::EC2::VPC' and 'CFN Resource'. Below this, there is an 'Editing' dropdown menu set to 'VPC'. The 'Logical ID' section explains that updating this value will generate a new resource. A text input field contains the value 'VPC'. The 'Resource configuration' section explains that updating this value will change the resource's properties and that placeholder values should be replaced. A modal dialog is open in the foreground, displaying a loading spinner and the text 'Generating can take up to 30 seconds.' with a 'Stop generating' button. A 'Resource reference' button is also visible at the bottom right of the properties panel.

Le schede di risorse standard e le schede componenti standard possono rappresentare una CloudFormation risorsa o una raccolta di CloudFormation risorse. Per ulteriori informazioni, consulta [Configurazione e modifica delle schede in Infrastructure Composer](#).

## Configurazione

Per utilizzarlo Amazon Q in Infrastructure Composer, è necessario autenticarsi con Amazon Q il Toolkit. Per istruzioni, consulta [Guida introduttiva a VS Code e Amazon Q JetBrains nella Guida per l'Amazon Q Developer](#) utente.

## Utilizzo Amazon Q Developer in Infrastructure Composer

È possibile utilizzare Amazon Q Developer dal pannello delle proprietà delle risorse di qualsiasi risorsa standard o scheda componente standard.

Da utilizzare Amazon Q in Infrastructure Composer

1. Da una risorsa standard o da una scheda componente standard, aprite il pannello delle proprietà della risorsa.
2. Individuate il campo di configurazione delle risorse. Questo campo contiene il codice di infrastruttura per la scheda.
3. Seleziona il pulsante Genera suggerimenti. Amazon Q genererà un suggerimento.

### Note

Il codice generato in questa fase non sovrascriverà il codice di infrastruttura esistente del modello.

4. Per generare altri suggerimenti, seleziona Rigenera. Puoi passare da un campione all'altro per confrontare i risultati.
5. Per selezionare un'opzione, scegliete Seleziona. Puoi modificare il codice qui prima di salvarlo nella tua applicazione. Per uscire senza salvare, seleziona l'icona di uscita (X).
6. Per salvare il codice nel modello dell'applicazione, selezionate Salva dal pannello delle proprietà della risorsa.

## Ulteriori informazioni

Per saperne di più Amazon Q, vedi [Cos'è Amazon Q?](#) nella Guida Amazon Q Developer per l'utente.

# Come comporre in AWS Infrastructure Composer

Questa sezione illustra le nozioni di base sull'utilizzo di Infrastructure Composer di [Console Infrastructure Composer](#) e [AWS Toolkit for Visual Studio Code](#). Più specificamente, gli argomenti di questa sezione forniscono dettagli chiave su come comporre un'applicazione con Infrastructure Composer e includono dettagli su funzionalità e scorciatoie aggiuntive. Esistono alcune variazioni di funzionalità tra le esperienze console e VS Code e gli argomenti di questa sezione identificano e descrivono queste variazioni laddove si verificano.

Dopo aver composto l'applicazione, sarete pronti a leggere [Distribuisci la tua applicazione serverless Infrastructure Composer nel cloud AWS](#) le informazioni sulla distribuzione dell'applicazione.

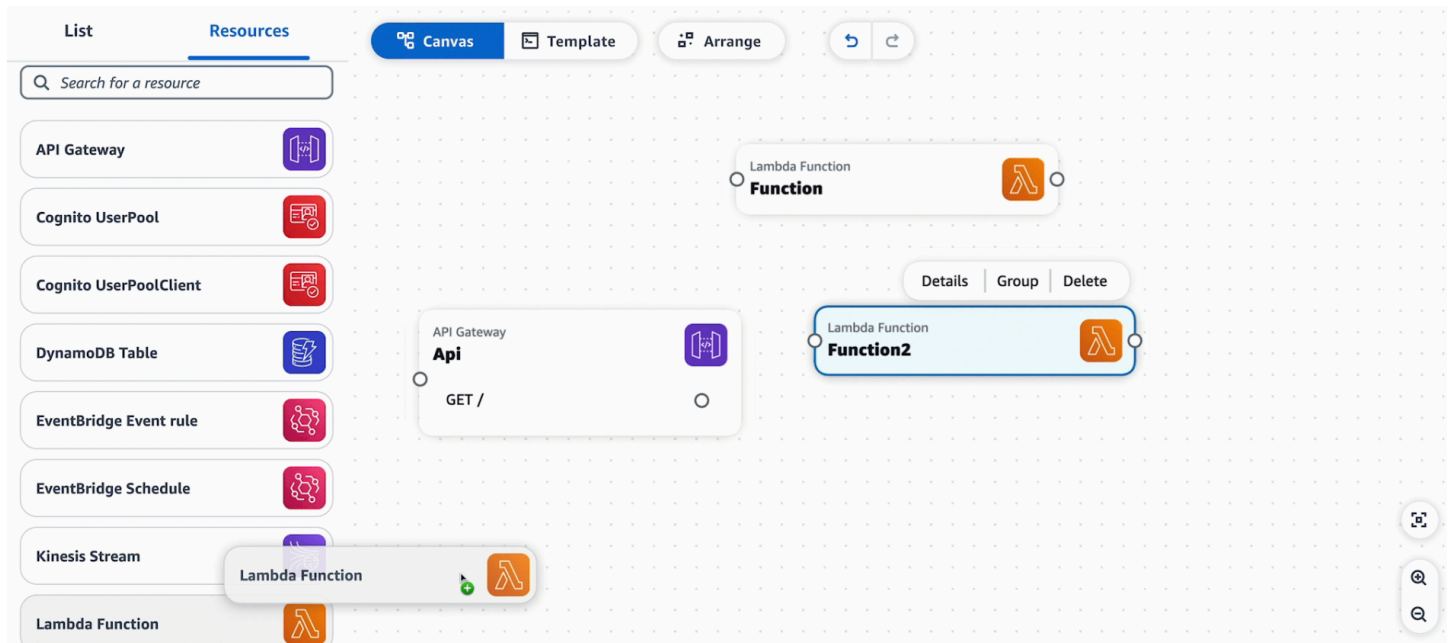
## Argomenti

- [Posiziona le carte sulla tela visiva di Infrastructure Composer](#)
- [Raggruppa le schede nella tela visiva di Infrastructure Composer](#)
- [Connect card sulla tela visiva di Infrastructure Composer](#)
- [Disconnetti le schede in Infrastructure Composer](#)
- [Disponi le carte sulla tela visiva di Infrastructure Composer](#)
- [Configurazione e modifica delle schede in Infrastructure Composer](#)
- [Eliminare schede in Infrastructure Composer](#)
- [Visualizza gli aggiornamenti del codice con Change Inspector in Infrastructure Composer](#)
- [Fai riferimento ai file esterni in Infrastructure Composer](#)
- [Integra Infrastructure Composer con Amazon Virtual Private Cloud \(Amazon VPC\)](#)

## Posiziona le carte sulla tela visiva di Infrastructure Composer

Questa sezione descrive come selezionare e trascinare le [schede](#) di Infrastructure Composer nella relativa area di visualizzazione. Prima di iniziare, identifica le risorse di cui ha bisogno la tua applicazione e come devono interagire. Per suggerimenti su come eseguire questa operazione, consulta [Crea la tua prima applicazione con Infrastructure Composer](#).

Per aggiungere una scheda all'applicazione, trascinala dalla palette delle risorse e rilasciala nell'area di disegno.



È possibile scegliere tra due tipi di schede: schede [componenti avanzate e schede](#) di risorse [IAC standard](#).

Dopo aver posizionato le carte sull'area di disegno, sarete pronti per raggrupparle, connetterle, disporle e configurarle. Per informazioni su questa operazione, consulta i seguenti argomenti:

- [Raggruppa le schede nella tela visiva di Infrastructure Composer](#)
- [Connect card sulla tela visiva di Infrastructure Composer](#)
- [Disponi le carte sulla tela visiva di Infrastructure Composer](#)
- [Configurazione e modifica delle schede in Infrastructure Composer](#)

## Raggruppa le schede nella tela visiva di Infrastructure Composer

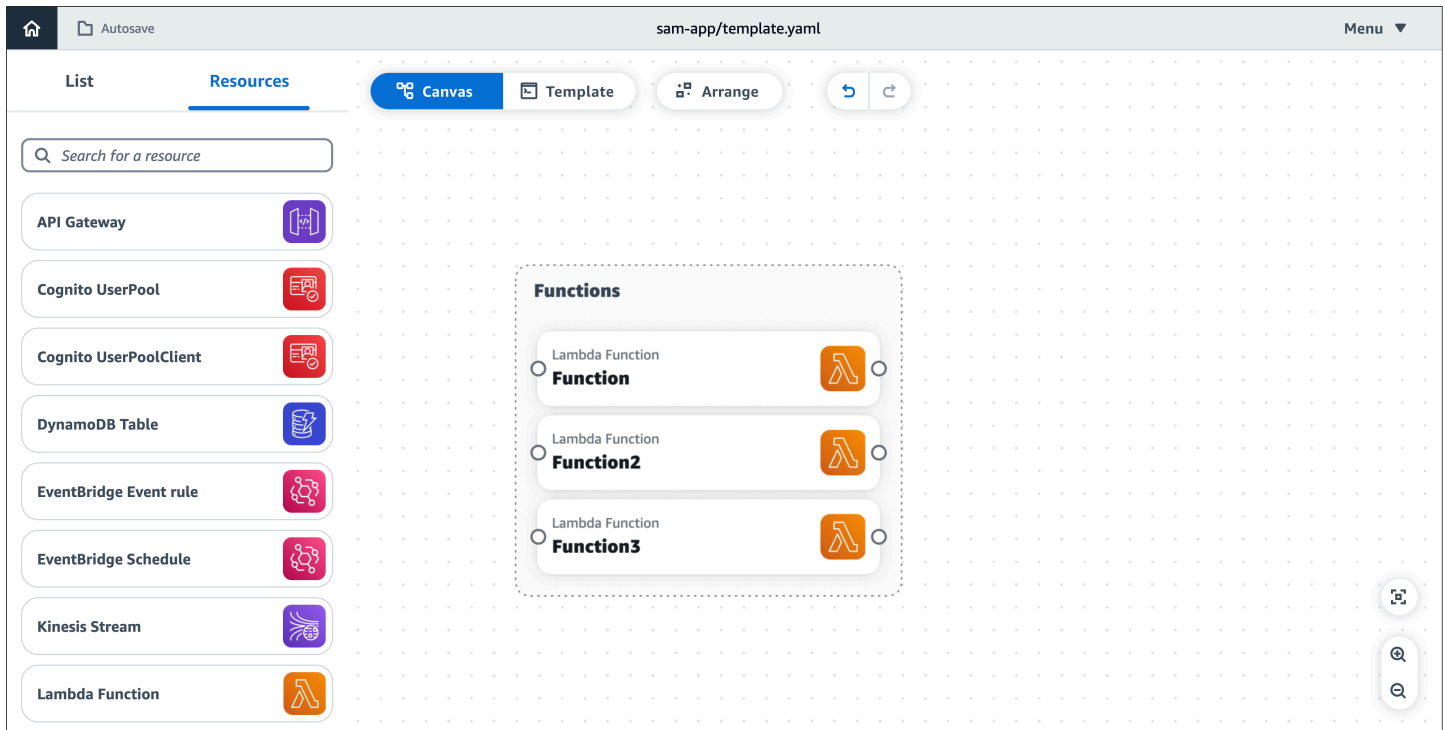
Questo argomento contiene dettagli sul raggruppamento di schede componenti avanzate e schede componenti standard. Il raggruppamento delle schede consente di classificare e organizzare le risorse senza dover pensare al codice o al markup da scrivere.

### Raggruppamento di schede componenti avanzate

Esistono due modi per raggruppare le schede componenti avanzate:

- Premendo Shift, seleziona le carte da raggruppare. Quindi, scegli Raggruppa dal menu delle azioni sulle risorse.

- seleziona una carta che desideri inserire in un gruppo. Dal menu visualizzato, seleziona Gruppo. Questo creerà un gruppo in cui puoi trascinare e rilasciare altre carte.



## Raggruppamento di una scheda componente standard in un'altra

L'esempio seguente mostra un modo in cui una scheda componente standard può essere raggruppata in un'altra scheda dal pannello delle proprietà delle risorse:

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

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

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

Nel campo Configurazione delle risorse del pannello delle proprietà delle risorse, `Role` è stato fatto riferimento alla funzione Lambda. Ciò fa sì che la scheda Ruolo venga raggruppata nella scheda Funzione sull'area di disegno.

# Connect card sulla tela visiva di Infrastructure Composer

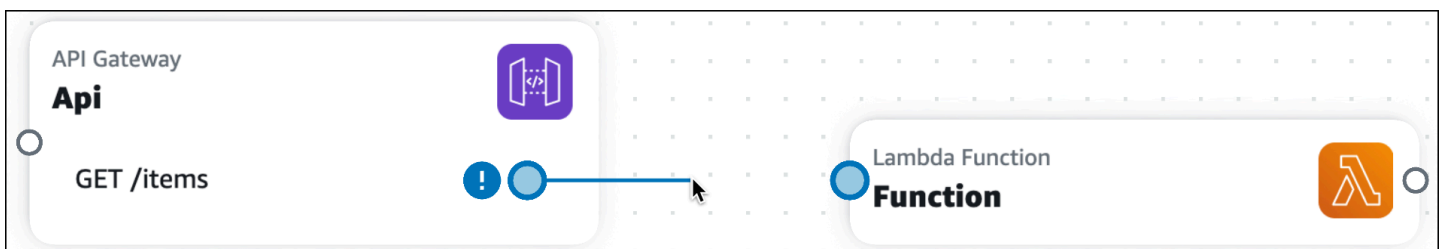
Usa questo argomento per capire come collegare le schede in Infrastructure Composer. Questa sezione include dettagli sul collegamento di schede componenti avanzate e schede componenti standard. Fornisce inoltre alcuni esempi che illustrano i diversi modi in cui le schede possono essere collegate.

## Collegamento di schede componenti avanzate

Nelle schede componenti avanzate, le porte identificano visivamente dove è possibile effettuare le connessioni.

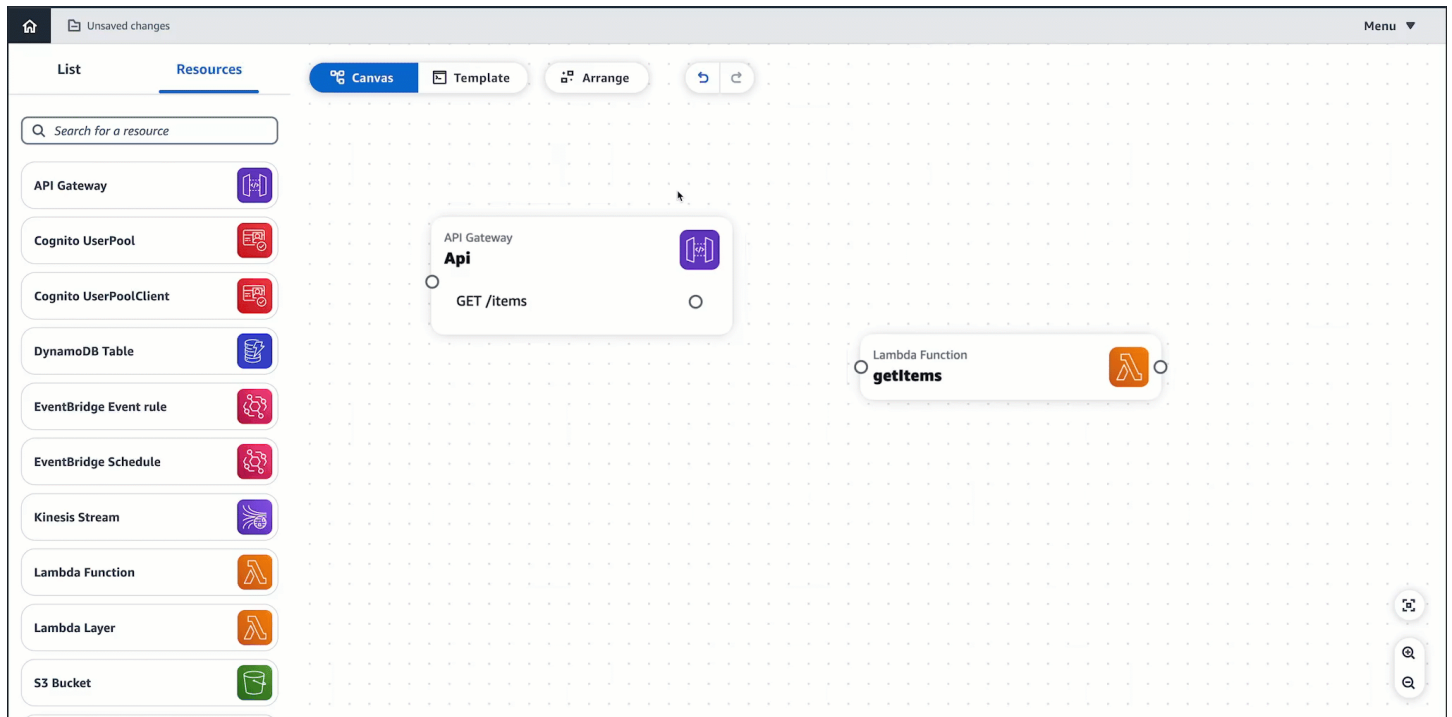
- Una porta sul lato destro di una scheda indica la possibilità per la scheda di richiamare un'altra carta.
- Una porta sul lato sinistro di una carta indica la possibilità che la carta venga richiamata da un'altra carta.

Connect le schede tra loro facendo clic sulla porta destra di una scheda e trascinandola sulla porta sinistra di un'altra scheda.



Quando crei una connessione, viene visualizzato un messaggio che ti informa se la connessione è stata effettuata correttamente. Seleziona il messaggio per vedere cosa ha modificato Infrastructure Composer per fornire una connessione. Se la connessione non è riuscita, puoi selezionare la visualizzazione Modello per aggiornare manualmente il codice dell'infrastruttura per fornire la connessione.

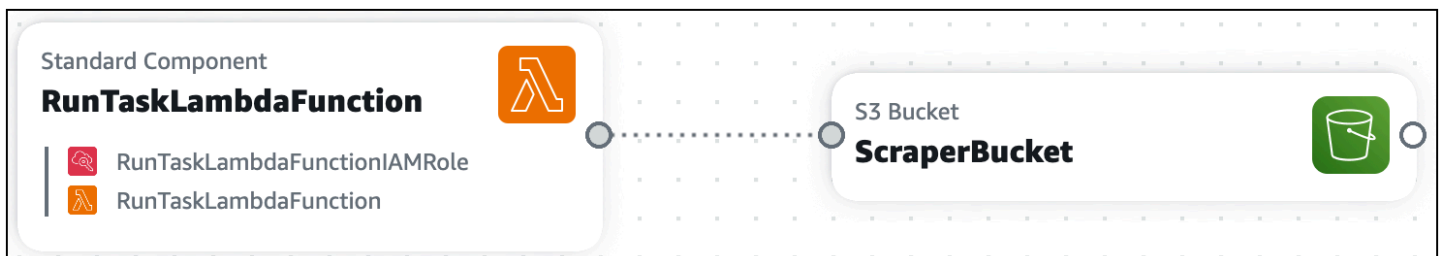
- In caso di successo, fai clic sul messaggio per visualizzare l'ispettore delle modifiche. Qui puoi vedere cosa ha modificato Infrastructure Composer per effettuare il provisioning della tua connessione.
- In caso di esito negativo, verrà visualizzato un messaggio. È possibile selezionare la visualizzazione Modello e aggiornare manualmente il codice dell'infrastruttura per effettuare il provisioning della connessione.



Quando si collegano insieme schede componenti avanzate, Infrastructure Composer crea automaticamente il codice dell'infrastruttura nel modello per fornire la relazione basata sugli eventi tra le risorse.

## Collegamento di schede componenti standard (schede di risorse IAc standard)

Le schede di risorse IAc standard non includono porte per creare connessioni con altre risorse. Durante la [configurazione della scheda](#), specificate le relazioni basate sugli eventi nel modello dell'applicazione, Infrastructure Composer rileverà automaticamente queste connessioni e le visualizzerà con una linea tratteggiata tra le schede. Di seguito è riportato un esempio di connessione tra una scheda componente standard e una scheda componente avanzata:



L'esempio seguente mostra come una funzione Lambda può essere connessa a un'API rest di 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

```

Nell'esempio precedente, il frammento di codice elencato `ApiGatewayMethod:` sotto `Integration:` specifica la relazione basata sugli eventi che collega le due schede.

## Esempi di connessione di schede in Infrastructure Composer

Usa gli esempi in questa sezione per capire come le schede possono essere collegate in Infrastructure Composer.

### Richiama una AWS Lambda funzione quando un articolo viene inserito in un bucket Amazon Simple Storage Service (Amazon S3)

In questo esempio, una bucket card Amazon S3 è collegata a una scheda funzione Lambda. Quando un elemento viene inserito nel bucket Amazon S3, viene richiamata la funzione. La funzione può quindi essere utilizzata per elaborare l'elemento o attivare altri eventi nell'applicazione.



Questa interazione richiede la definizione di un evento per la funzione. Ecco cosa prevede Infrastructure Composer:

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

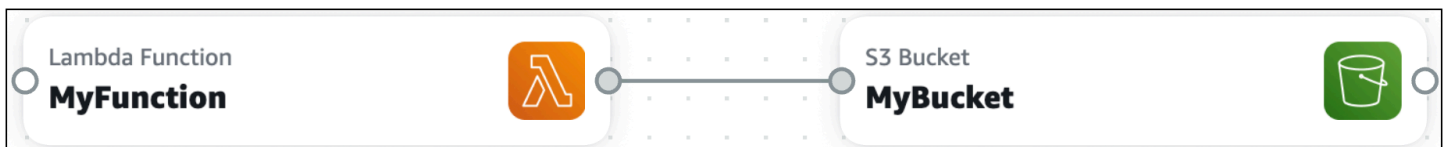
```

...
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

```

## Richiama un bucket Amazon S3 da una funzione Lambda

In questo esempio, una scheda funzione Lambda richiama una bucket card Amazon S3. La funzione Lambda può essere utilizzata per eseguire operazioni CRUD sugli articoli nel bucket Amazon S3.



Questa interazione richiede quanto segue, fornito da Infrastructure Composer:

- Policy IAM che consentono alla funzione Lambda di interagire con il bucket Amazon S3.
- Variabili di ambiente che influenzano il comportamento della funzione 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

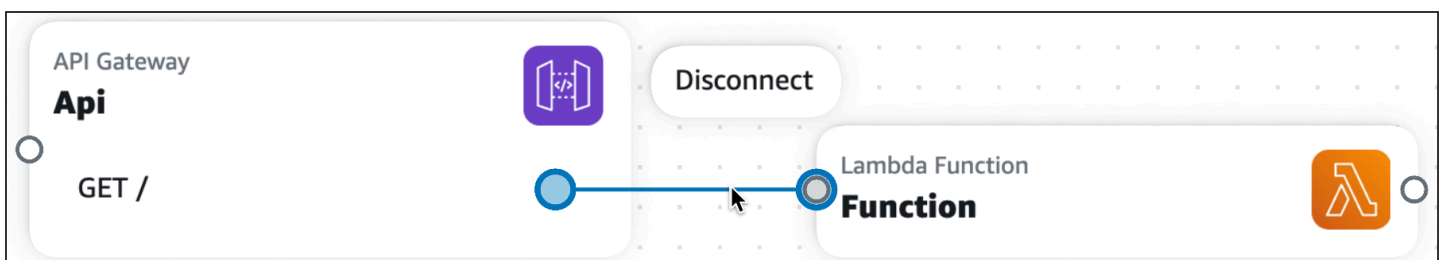
```

## Disconnetti le schede in Infrastructure Composer

In Infrastructure Composer, è possibile connettere e disconnettere AWS le risorse utilizzando schede componenti avanzate e schede componenti standard. Questa sezione descrive come disconnettere entrambi i tipi di schede.

### Schede componenti avanzate

Per disconnettere le schede componenti avanzate, seleziona la linea e scegli Disconnetti.



Infrastructure Composer modificherà automaticamente il modello per rimuovere la relazione basata sugli eventi dall'applicazione.

### Schede componenti standard

Le schede componenti standard non includono porte per creare connessioni con altre risorse. Durante la [configurazione della scheda](#), specificate le relazioni basate sugli eventi nel modello dell'applicazione, Infrastructure Composer rileverà automaticamente queste connessioni e le

visualizzerà con una linea tratteggiata tra le schede. Per disconnettere una scheda componente standard, rimuovete la relazione basata sugli eventi nel modello dell'applicazione.

L'esempio seguente mostra una funzione Lambda connessa a un'API rest di 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:

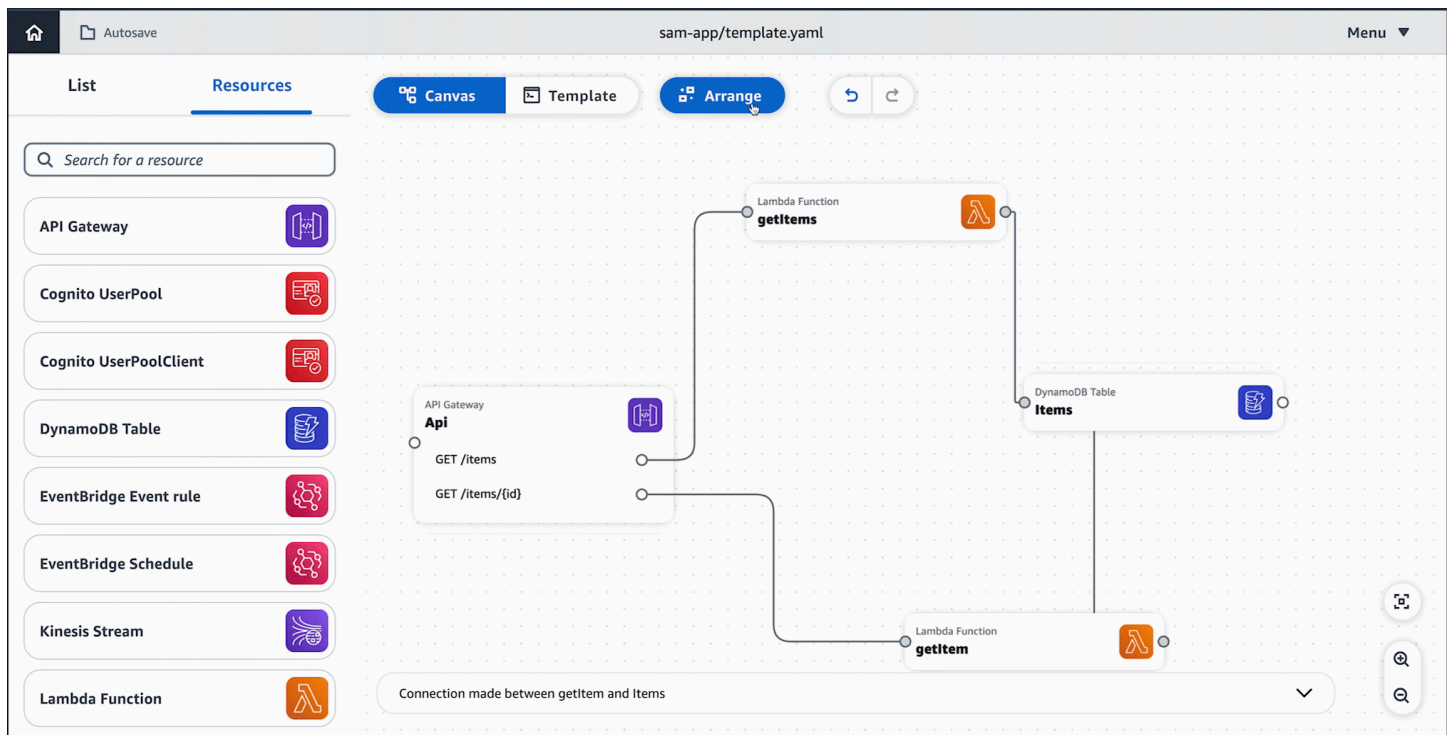
```

```
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
```

Per rimuovere la connessione tra le due schede, rimuovi i riferimenti a `MyLambdaFunction` `ApiGatewayMethod`: quelli elencati di seguito. `Integration`

## Disponi le carte sulla tela visiva di Infrastructure Composer

Seleziona `Disponi` per disporre e organizzare visivamente le carte sulla tela. L'uso del pulsante `Disponi` è particolarmente utile quando ci sono molte schede e connessioni sull'area di disegno.



## Configurazione e modifica delle schede in Infrastructure Composer

In Infrastructure Composer, le schede rappresentano le risorse utilizzate per progettare l'architettura dell'applicazione. Quando si configura una scheda in Infrastructure Composer, si definiscono i dettagli delle risorse dell'applicazione. Ciò include dettagli come l'ID logico e la chiave di partizione di una scheda. Il modo in cui vengono definite queste informazioni varia tra le schede componenti Enhanced e le schede Standard.

Una scheda componente avanzata è una raccolta di CloudFormation risorse che sono state combinate in un'unica scheda curata che migliora la facilità d'uso e la funzionalità e sono progettate per un'ampia varietà di casi d'uso. Una scheda di risorse IAc standard rappresenta una singola risorsa. AWS CloudFormation Ogni scheda di risorse IAc standard, una volta trascinata sulla tela, è etichettata come componente standard.

Questo argomento fornisce dettagli sulla configurazione delle schede componente Enhanced e delle schede componente Standard.

### Note

Questo argomento si applica all'utilizzo delle schede della console Infrastructure Composer, dell' AWS Toolkit for Visual Studio Code estensione e in Infrastructure Composer in

CloudFormation modalità console. Le schede relative a Lambda (Lambda Function e Lambda Layer) richiedono build di codice e soluzioni di packaging che non sono disponibili in Infrastructure Composer in modalità console. CloudFormation Per ulteriori informazioni, consulta [Utilizzo di Infrastructure Composer in modalità CloudFormation console](#).

## Argomenti

- [Schede componenti avanzate in Infrastructure Composer](#)
- [Schede standard in Infrastructure Composer](#)

## Schede componenti avanzate in Infrastructure Composer

Per configurare schede componenti avanzate, Infrastructure Composer fornisce un modulo nel pannello delle proprietà delle risorse. Questo modulo è stato creato in modo univoco per guidare l'utente nella configurazione di ogni scheda componente avanzata. Durante la compilazione del modulo, Infrastructure Composer modifica il codice dell'infrastruttura.

Alcune schede componenti avanzate dispongono di funzionalità aggiuntive. Questa sezione illustra le nozioni di base sull'uso delle schede componenti avanzate e offre dettagli sulle schede con funzionalità aggiuntive.

Per ulteriori informazioni sulle schede componenti avanzate, consulta [Schede componenti avanzate in Infrastructure Composer](#) e [Schede componenti avanzate in Infrastructure Composer](#)

## Procedura

Il pannello delle proprietà delle risorse semplifica la configurazione e aggiunge guide che semplificano la configurazione delle schede. Per utilizzare questo pannello, effettuate le seguenti operazioni:

1. Fate doppio clic su una scheda per visualizzare il pannello delle proprietà delle risorse.
2. Fai clic su una scheda e seleziona Dettagli per visualizzare il pannello delle proprietà della risorsa.
3. Per Infrastructure Composer da Console di gestione AWS, seleziona Template per mostrare il codice dell'applicazione. Configura direttamente da qui.

L'immagine seguente mostra come è possibile eseguire questa operazione:

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' sidebar with a search bar and a list of resource types: API Gateway, Cognito UserPool, Cognito UserPoolClient, DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, and Lambda Function. The main area is a 'Canvas' showing a SAM template in YAML format. The template defines a resource named 'Table' of type 'AWS::DynamoDB::Table'. The properties include 'AttributeDefinitions' with an attribute named 'id' of type 'S', 'BillingMode' set to 'PAY\_PER\_REQUEST', and 'KeySchema' with a primary key named 'id' of type 'HASH'. The 'StreamSpecification' is also defined with 'StreamViewType' set to 'NEW\_AND\_OLD\_IMAGES'. The right sidebar shows the 'Resource properties' for the selected 'Table' resource, including a description 'A key-value database', a 'Logical ID' field with the value 'Table', a 'Partition key' field with the value 'id', a 'Partition key type' dropdown set to 'String', and an 'Expiration key' checkbox which is currently unchecked.

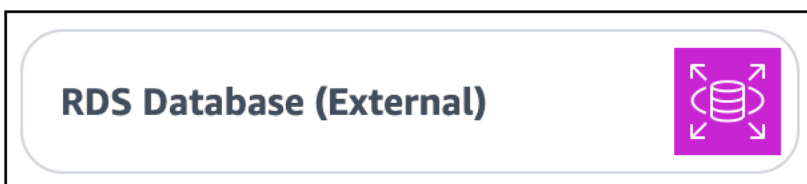
## Utilizzo di Infrastructure Composer con Amazon Relational Database Service (Amazon RDS)

AWS Infrastructure Composer presenta un'integrazione con Amazon Relational Database Service (Amazon RDS). Utilizzando la scheda componente avanzata del database RDS (esterno) in Infrastructure Composer, puoi connettere la tua applicazione a DB cluster, istanze e proxy Amazon RDS definiti su un altro modello o (). CloudFormation AWS Serverless Application Model AWS SAM

La scheda componente avanzata del database RDS (esterno) rappresenta le risorse Amazon RDS definite su un altro modello. Questo include:

- DBCluster o istanza Amazon RDS definito su un altro modello
- Proxy Amazon RDS DB

La scheda componente avanzata del database RDS (esterno) è disponibile nella palette Risorse.



Per utilizzare questa scheda, trascinala sul canvas di Infrastructure Composer, configurala e collegala ad altre risorse.

Puoi connettere la tua applicazione al DB cluster o all'istanza Amazon RDS esterni tramite una funzione Lambda.

## Requisiti

Per utilizzare questa funzionalità, è necessario soddisfare i seguenti requisiti:

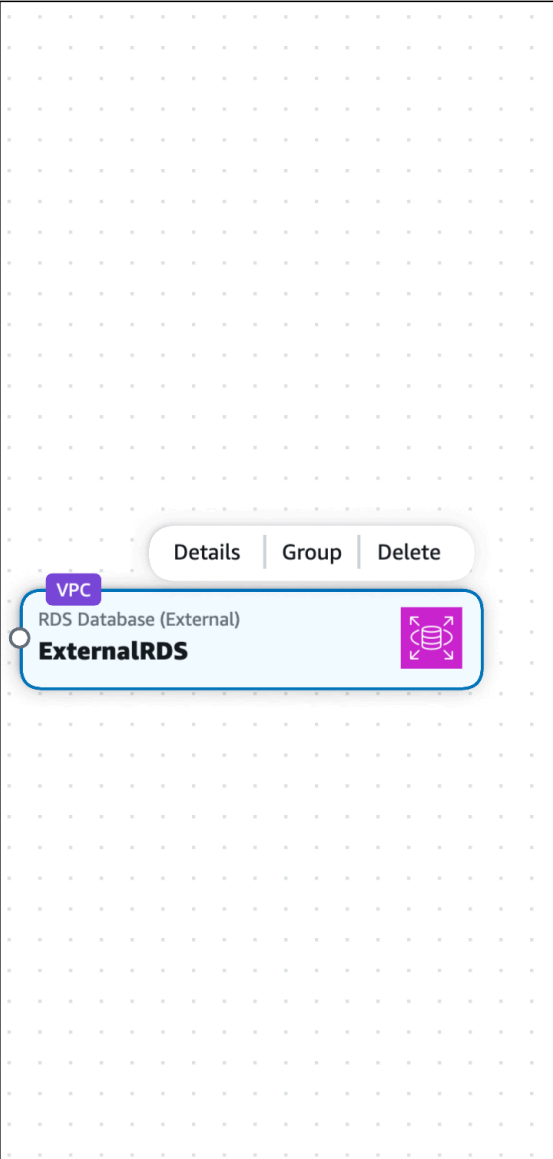
1. È necessario utilizzare il DB cluster, l'istanza o il proxy Amazon RDS esterno Gestione dei segreti AWS per gestire la password dell'utente. Per ulteriori informazioni, consulta [la sezione Gestione delle password con Amazon RDS e Gestione dei segreti AWS](#) la Amazon RDS User Guide.
2. L'applicazione in Infrastructure Composer deve essere un nuovo progetto o deve essere stata originariamente creata in Infrastructure Composer.

## Procedura

### Fase 1: Configurare la scheda RDS Database esterna

Dalla palette Risorse, trascina una scheda componente avanzata del database RDS (esterno) sull'area di lavoro.

Seleziona la scheda e scegli Dettagli o fai doppio clic sulla scheda per visualizzare il pannello delle proprietà delle risorse. Apparirà il pannello delle proprietà delle risorse della scheda:



## 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}.

È possibile configurare quanto segue qui:

- ID logico: un nome univoco per il DB cluster, l'istanza o il proxy Amazon RDS esterno. Questo ID non deve necessariamente corrispondere al valore dell'ID logico della DB risorsa Amazon RDS esterna.
- Segreto del database: un identificatore per il Gestione dei segreti AWS segreto associato al DB cluster, all'istanza o al proxy Amazon RDS. Questo campo accetta i seguenti valori:
  - Valore statico: un identificatore univoco del segreto del database, ad esempio l'ARN segreto. Di seguito è riportato un esempio: `arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-name-1a2b3c`. Per ulteriori

informazioni, consulta [Concetti di Gestione dei segreti AWS](#) nella Guida per l'utente di Gestione dei segreti AWS .

- Valore di output: quando viene distribuito un segreto di Secrets Manager su AWS CloudFormation, viene creato un valore di output. È possibile specificare qui il valore di output utilizzando la funzione [Fn::ImportValue](#) intrinseca. Ad esempio, `!ImportValue MySecret`.
- Valore dall'archivio dei parametri SSM: è possibile memorizzare il segreto nell'archivio dei parametri SSM e specificarne il valore utilizzando un riferimento dinamico. Ad esempio, `{{resolve:ssm:MySecret}}`. Per ulteriori informazioni, consulta i [parametri SSM](#) nella Guida per l'AWS CloudFormation utente.
- Nome host del database: il nome host che può essere utilizzato per connettersi al DB cluster, all'istanza o al proxy Amazon RDS. Questo valore è specificato nel modello esterno che definisce la tua risorsa Amazon RDS. Sono accettati i seguenti valori:
  - Valore statico: un identificatore univoco del nome host del database, ad esempio l'indirizzo dell'endpoint. Di seguito è riportato un esempio: `mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-2.rds.amazonaws.com`.
  - Valore di output: il valore di output di un DB cluster, istanza o proxy Amazon RDS distribuito. È possibile specificare il valore di output utilizzando la funzione [Fn::ImportValue](#) intrinseca. Ad esempio, `!ImportValue myStack-myDatabase-abcd1234`.
  - Valore dall'archivio dei parametri SSM: è possibile memorizzare il nome host del database nell'archivio dei parametri SSM e specificarne il valore utilizzando un riferimento dinamico. Ad esempio, `{{resolve:ssm:MyDatabase}}`.
- Porta del database: il numero di porta che può essere utilizzato per connettersi al DB cluster, all'istanza o al proxy Amazon RDS. Questo valore è specificato nel modello esterno che definisce la tua risorsa Amazon RDS. Sono accettati i seguenti valori:
  - Valore statico: la porta del database. Ad esempio, `3306`.
  - Valore di output: il valore di output di un DB cluster, istanza o proxy Amazon RDS distribuito. Ad esempio, `!ImportValue myStack-MyRDSInstancePort`.
  - Valore da SSM Parameter Store: puoi memorizzare il nome host del database nell'SSM Parameter Store e specificarne il valore utilizzando un riferimento dinamico. Ad esempio, `{{resolve:ssm:MyRDSInstancePort}}`.

**Note**

Qui deve essere configurato solo il valore dell'ID logico. Se preferisci, puoi configurare le altre proprietà al momento della distribuzione.

**Fase 2: Connect una scheda Lambda Function**

Dalla palette Risorse, trascina una scheda componente avanzata della funzione Lambda sull'area di disegno.

Connect la porta sinistra della scheda Function Lambda alla porta destra della scheda RDS Database (esterna).



Infrastructure Composer fornirà il modello per facilitare questa connessione.

Cosa fa Infrastructure Composer per creare la tua connessione

Una volta completata la procedura sopra elencata, Infrastructure Composer esegue azioni specifiche per connettere la funzione Lambda al database.

Quando si specifica il DB cluster, l'istanza o il proxy Amazon RDS esterno

Quando trascini una scheda del database RDS (esterna) sull'area di disegno, Infrastructure Composer aggiorna le Parameters sezioni Metadata e le sezioni del modello in base alle esigenze. Di seguito è riportato un esempio:

```
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

```

I [metadati](#) sono una sezione CloudFormation del modello che viene utilizzata per memorizzare i dettagli sul modello. I metadati specifici di Infrastructure Composer vengono archiviati nella chiave dei `AWS::Composer::ExternalResources` metadati. Qui, Infrastructure Composer memorizza i valori specificati per il DB cluster, l'istanza o il proxy Amazon RDS.

La sezione [Parametri](#) di un CloudFormation modello viene utilizzata per memorizzare valori personalizzati che possono essere inseriti nel modello al momento della distribuzione. A seconda del tipo di valori che fornisci, Infrastructure Composer può memorizzare qui i valori per il tuo DB cluster, istanza o proxy Amazon RDS e specificarli in tutto il modello.

I valori di stringa nella `Parameters` sezione `Metadata` and utilizzano il valore dell'ID logico specificato sulla scheda del database RDS (esterna). Se aggiorni l'ID logico, i valori della stringa cambieranno.

Quando si collega la funzione Lambda al database

Quando si collega una scheda Lambda Function alla scheda RDS Database (esterna), Infrastructure Composer esegue il provisioning delle variabili di ambiente e delle politiche AWS Identity and Access Management (IAM). Di seguito è riportato un esempio:

```

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

```

Le variabili di [ambiente](#) sono variabili che possono essere utilizzate dalla funzione in fase di esecuzione. Per ulteriori informazioni, consulta [Using Lambda Environmental Variables](#) nella AWS Lambda Developer Guide.

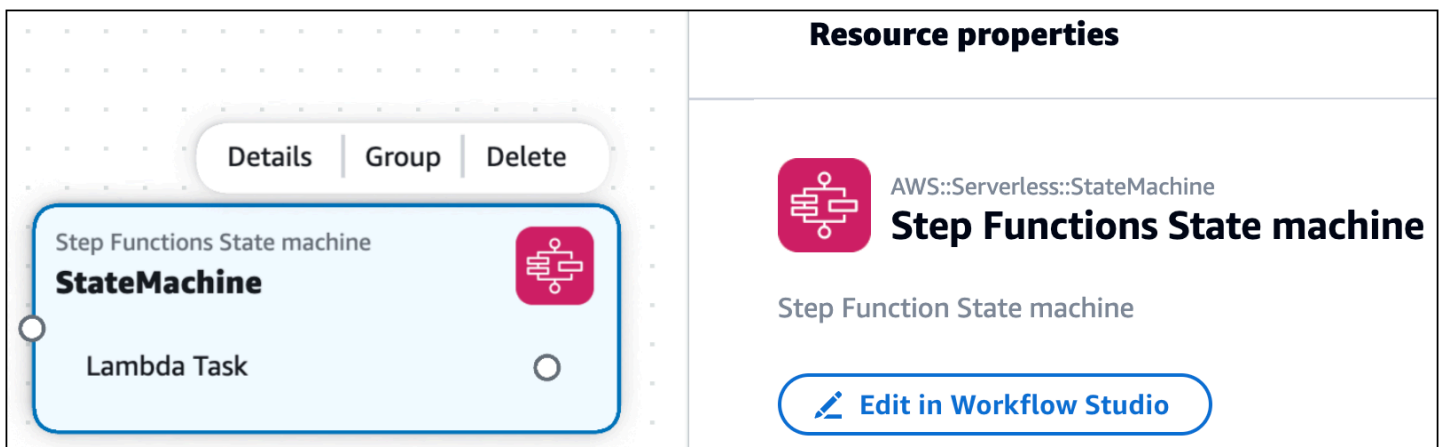
[Le politiche](#) forniscono le autorizzazioni per la tua funzione. Qui, Infrastructure Composer crea una policy per consentire l'accesso in lettura dalla tua funzione a Secrets Manager per ottenere la password per l'accesso al DB cluster, all'istanza o al proxy Amazon RDS.

## Utilizzo AWS Infrastructure Composer con AWS Step Functions

AWS Infrastructure Composer presenta un'integrazione con [AWS Step Functions Workflow Studio](#). Utilizza Infrastructure Composer per effettuare le seguenti operazioni:

- Avvia Step Functions Workflow Studio direttamente all'interno di Infrastructure Composer.
- Crea e gestisci nuovi flussi di lavoro o importa flussi di lavoro esistenti in Infrastructure Composer.
- Integra i tuoi flussi di lavoro con altre AWS risorse utilizzando il canvas Infrastructure Composer.

L'immagine seguente mostra una scheda macchina Step Functions State



Con Step Functions Workflow Studio in Infrastructure Composer, puoi sfruttare i vantaggi di due potenti designer visivi in un unico posto. Mentre progettate il flusso di lavoro e l'applicazione, Infrastructure Composer crea l'infrastruttura come codice (IaC) per guidarvi verso l'implementazione.

### Argomenti

- [Policy IAM](#)
- [Guida introduttiva a Step Functions Workflow Studio in Infrastructure Composer](#)
- [Utilizzo di Step Functions Workflow Studio in Infrastructure Composer](#)

- [Ulteriori informazioni](#)

## Policy IAM

Quando colleghi le attività dal flusso di lavoro alle risorse, Infrastructure Composer crea automaticamente le policy AWS Identity and Access Management (IAM) necessarie per autorizzare l'interazione tra le tue risorse. Di seguito è riportato un esempio:

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

Se necessario, puoi aggiungere altre policy IAM al tuo modello.

Guida introduttiva a Step Functions Workflow Studio in Infrastructure Composer

Per iniziare, puoi creare nuovi flussi di lavoro o importare flussi di lavoro esistenti.

Per creare un nuovo flusso di lavoro

1. Dalla palette Resources, trascina una scheda componente avanzata della macchina Step Functions State sull'area di disegno.



Quando trascini una scheda macchina Step Functions State sull'area di disegno, Infrastructure Composer crea quanto segue:

- Una [AWS::Serverless::StateMachine](#) risorsa che definisce la vostra macchina a stati. Per impostazione predefinita, Infrastructure Composer crea un flusso di lavoro standard. Per creare un flusso di lavoro rapido, modifica il Type valore nel modello da STANDARD aEXPRESS.
  - Una [AWS::Logs::LogGroup](#) risorsa che definisce un gruppo di CloudWatch log Amazon per la tua macchina a stati.
2. Apri il pannello delle proprietà delle risorse della scheda e seleziona Modifica in Workflow Studio per aprirlo Workflow Studio in Infrastructure Composer.

Step Functions Workflow Studio si apre in modalità Progettazione. Per ulteriori informazioni, consulta la [modalità Progettazione](#) nella Guida per AWS Step Functions gli sviluppatori.

#### Note

È possibile modificare Infrastructure Composer per salvare la definizione della macchina a stati in un file esterno. Per ulteriori informazioni, consulta [Lavorare con file esterni](#).

3. Crea il tuo flusso di lavoro e scegli Salva. Per uscireWorkflow Studio, scegli Return to Infrastructure Composer.

Infrastructure Composer definisce il flusso di lavoro utilizzando la Definition proprietà della AWS::Serverless::StateMachine risorsa.

4. È possibile modificare il flusso di lavoro effettuando una delle seguenti operazioni:
  - Workflow StudioRiapri e modifica il flusso di lavoro.
  - Per Infrastructure Composer dalla console, puoi aprire la visualizzazione Modello dell'applicazione e modificare il modello. Se si utilizza la sincronizzazione locale, è possibile modificare il flusso di lavoro nell'IDE locale. Infrastructure Composer rileverà le modifiche e aggiornerà il flusso di lavoro in Infrastructure Composer.
  - Per Infrastructure Composer del Toolkit for VS Code, puoi modificare direttamente il tuo modello. Infrastructure Composer rileverà le modifiche e aggiornerà il flusso di lavoro in Infrastructure Composer.

## Per importare flussi di lavoro esistenti

È possibile importare flussi di lavoro da applicazioni definite utilizzando modelli AWS Serverless Application Model (AWS SAM). Usa qualsiasi macchina a stati definita con il tipo di `AWS::Serverless::StateMachine` risorsa e verrà visualizzata come una scheda componente avanzata della macchina a stati Step Functions che puoi utilizzare per l'avvioWorkflow Studio.

La `AWS::Serverless::StateMachine` risorsa può definire i flussi di lavoro utilizzando una delle seguenti proprietà:

- [Definition](#)— Il flusso di lavoro è definito all'interno del AWS SAM modello come oggetto.
- [DefinitionUri](#)— Il flusso di lavoro è definito su un file esterno utilizzando [Amazon States Language](#). Il percorso locale del file viene quindi specificato con questa proprietà.

### Proprietà di definizione

#### Infrastructure Composer dalla console

Per i flussi di lavoro definiti utilizzando la `Definition` proprietà, è possibile importare un singolo modello o l'intero progetto.

- Modello: per istruzioni sull'importazione di un modello, vedere [Importa un modello di progetto esistente nella console Infrastructure Composer](#). Per salvare le modifiche apportate in Infrastructure Composer, è necessario esportare il modello.
- Progetto: quando importate un progetto, dovete attivare la sincronizzazione locale. Le modifiche apportate vengono salvate automaticamente sul computer locale. Per istruzioni sull'importazione di un progetto, vedere [Importa una cartella di progetto esistente nella console Infrastructure Composer](#).

#### Infrastructure Composer del Toolkit for VS Code

Per i flussi di lavoro definiti utilizzando la `Definition` proprietà, puoi aprire Infrastructure Composer dal tuo modello. Per istruzioni, consulta [Accedi a Infrastructure Composer dal AWS Toolkit for Visual Studio Code](#).

## DefinitionUri proprietà

### Infrastructure Composer dalla console

Per i flussi di lavoro definiti utilizzando la `DefinitionUri` proprietà, è necessario importare il progetto e attivare la sincronizzazione locale. Per istruzioni sull'importazione di un progetto, vedere. [Importa una cartella di progetto esistente nella console Infrastructure Composer](#)

### Infrastructure Composer del Toolkit for VS Code

Per i flussi di lavoro definiti utilizzando la `DefinitionUri` proprietà, puoi aprire Infrastructure Composer dal tuo modello. Per istruzioni, consulta [Accedi a Infrastructure Composer dal AWS Toolkit for Visual Studio Code](#).

## Utilizzo di Step Functions Workflow Studio in Infrastructure Composer

### Crea flussi di lavoro

Infrastructure Composer utilizza le sostituzioni delle definizioni per mappare le attività del flusso di lavoro alle risorse dell'applicazione. Per ulteriori informazioni sulle sostituzioni delle definizioni, consulta [DefinitionSubstitutions](#) la Guida per gli sviluppatori.AWS Serverless Application Model

Quando crei attività inWorkflow Studio, specifica una definizione sostitutiva per ogni attività. È quindi possibile connettere le attività alle risorse nell'area di disegno di Infrastructure Composer.

Per specificare una sostituzione della definizione in Workflow Studio

1. Apri la scheda Configurazione dell'attività e individua il campo Parametri API.

The screenshot displays the AWS Infrastructure Composer interface. On the left, a state machine workflow is visualized on a grid background. It starts with a 'Start' node, followed by a 'Lambda: Invoke Check Stock Value' activity. This leads to a 'Choice state Choice' activity with a decision condition '\$.stock\_price <= 50'. If the condition is true, it proceeds to 'Lambda: Invoke Buy Stock'; otherwise, it goes to 'Lambda: Invoke Sell Stock'. Both paths converge into a 'DynamoDB: PutItem Record Transaction' activity, which then ends at an 'End' node.

On the right, the configuration panel for the state machine is shown. The title is 'Check Stock Value' with a 'Definition' toggle and a right arrow. Below the title are four tabs: 'Configuration' (selected), 'Input', 'Output', and 'Error handling'. The configuration section includes:
 

- State name:** A text input field containing 'Check Stock Value'.
- API:** A dropdown menu set to 'Lambda: Invoke'.
- Integration type:** A dropdown menu set to 'Optimized'.
- API Parameters:** A section with an 'Edit as JSON' toggle. It contains a 'Function name' field with a placeholder 'Enter a Cloudformation substitution' and a dropdown arrow. Below it, a text input field contains the substitution string '\$\${LambdaFunction1}'.

 A note at the bottom of the configuration panel states: 'Substitutions must be specified in \${dollar\_sign\_brace} notation. They will be mapped via the DefinitionSubstitution property inside your StateMachine resource in the Application Composer Canvas.'

- Se il campo Parametri API ha un'opzione a discesa, scegli Inserisci una CloudFormation sostituzione. Quindi, fornisci un nome univoco.

Per le attività che si connettono alla stessa risorsa, specifica la stessa definizione sostitutiva per ogni attività. Per utilizzare una sostituzione di definizione esistente, scegli Seleziona una sostituzione e seleziona la CloudFormation sostituzione da utilizzare.

- Se il campo Parametri API contiene un oggetto JSON, modifica la voce che specifica il nome della risorsa per utilizzare una definizione sostitutiva. Nell'esempio seguente, passiamo a "MyDynamoDBTable" "\${RecordTransaction}"

The diagram shows a state machine flow starting with a 'Start' node, followed by a 'Lambda: Invoke Check Stock Value' task. A 'Choice state Choice' follows, with a transition for '\$.stock\_price <= 50' leading to 'Lambda: Invoke Buy Stock' and a 'Default' path leading to 'Lambda: Invoke Sell Stock'. Both paths converge into a 'DynamoDB: PutItem Record Transaction' task, which ends at an 'End' node.

**Record Transaction** Definition >

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

**State name**  
Record Transaction

**API**  
DynamoDB: PutItem

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

Optimized

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

```

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

```

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

4. Seleziona Salva e torna a Infrastructure Composer.

Le attività del flusso di lavoro verranno visualizzate sulla scheda macchina Step Functions State.

Step Functions State machine

## StateMachine

Check Stock Value

Buy Stock

Sell Stock

Record Transaction

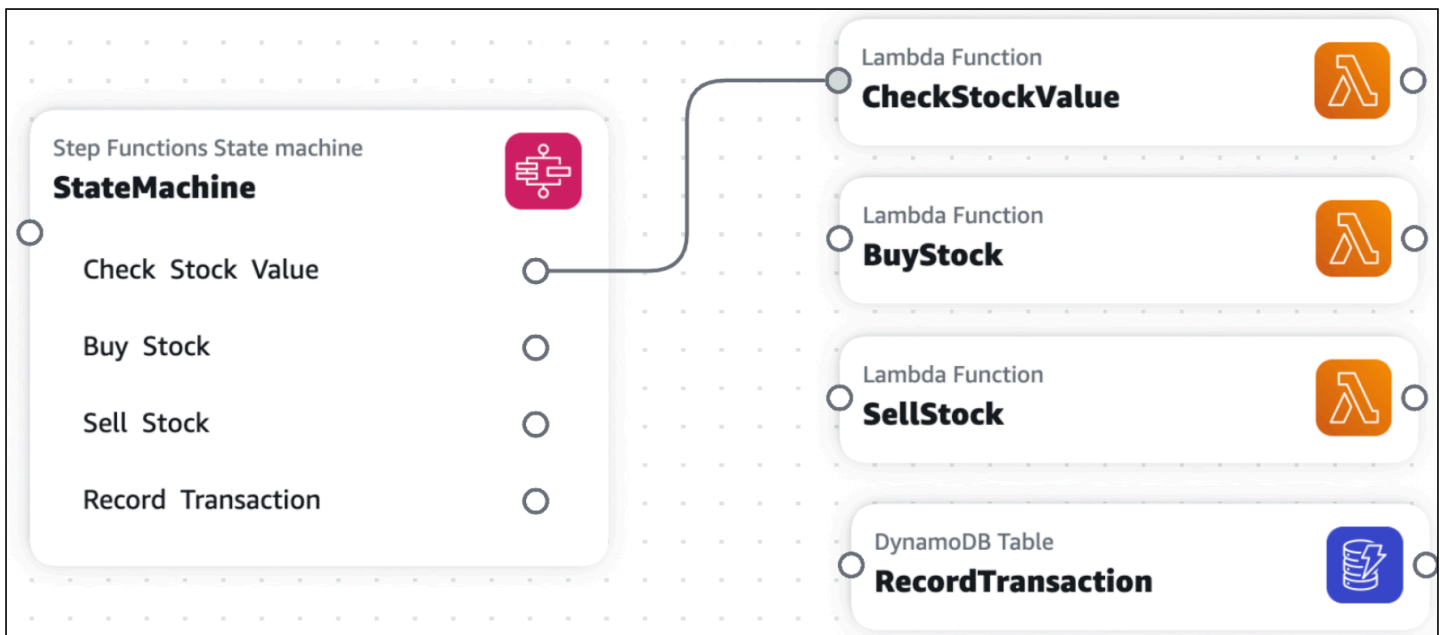
## Connect le risorse alle attività del flusso di lavoro

È possibile creare connessioni in Infrastructure Composer tra le attività di workflow supportate e le schede Infrastructure Composer supportate.

- Attività supportate del flusso di lavoro: attività per Servizi AWS le quali sono ottimizzate per Step Functions. Per ulteriori informazioni, consulta [Optimized integrations for Step Functions](#) nella AWS Step Functions Developer Guide.
- Schede Infrastructure Composer supportate: sono supportate le schede componenti avanzate. Per ulteriori informazioni sulle schede in Infrastructure Composer, consulta. [Configurazione e modifica delle schede in Infrastructure Composer](#)

Quando si crea una connessione, Servizio AWS l'attività e la scheda devono corrispondere. Ad esempio, è possibile collegare un'attività del flusso di lavoro che richiama una funzione Lambda a una scheda componente avanzata Lambda Function.

Per creare una connessione, fai clic e trascina la porta di un'attività sulla porta sinistra di una scheda componente avanzata.



Infrastructure Composer aggiornerà automaticamente il `DefinitionSubstitution` valore per definire la connessione. Di seguito è riportato un esempio:

```
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:
        ...
```

## Lavorare con file esterni

Quando si crea un flusso di lavoro dalla scheda macchina Step Functions State, Infrastructure Composer salva la definizione della macchina a stati all'interno del modello utilizzando la `Definition` proprietà. È possibile configurare Infrastructure Composer per salvare la definizione della macchina a stati su un file esterno.

### Note

Per utilizzare questa funzionalità con Infrastructure Composer di Console di gestione AWS, è necessario che la sincronizzazione locale sia attivata. Per ulteriori informazioni, consulta [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#).

Per salvare la definizione della macchina a stati in un file esterno

1. Apri il pannello delle proprietà delle risorse della scheda macchina Step Functions State.
2. Seleziona l'opzione Usa file esterno per la definizione della macchina a stati.
3. Fornite un percorso e un nome relativi per il file di definizione della macchina a stati.

#### 4. Scegli Save (Salva).

Infrastructure Composer eseguirà le seguenti operazioni:

1. Sposta la definizione della macchina a stati dal `Definition` campo al file esterno.
2. Salva la definizione della tua macchina a stati in un file esterno utilizzando Amazon States Language.
3. Modifica il modello per fare riferimento al file esterno utilizzando il `DefinitionUri` campo.

#### Ulteriori informazioni

Per ulteriori informazioni su Step Functions in Infrastructure Composer, consulta quanto segue:

- [Utilizzo Workflow Studio in Infrastructure Composer](#) nella Guida per gli AWS Step Functions sviluppatori.
- [DefinitionSubstitutions nei AWS SAM modelli](#) della AWS Step Functions Developer Guide.

## Schede standard in Infrastructure Composer

Tutte le CloudFormation risorse possono essere utilizzate come schede di risorse IaC standard dalla palette Risorse. Dopo essere stata trascinata sulla tela visiva, una scheda di risorse IaC standard diventa una scheda componente standard. Ciò significa semplicemente che la scheda è una o più risorse IaC standard. Per ulteriori esempi e dettagli, consulta gli argomenti di questa sezione.

È possibile modificare il codice dell'infrastruttura tramite la visualizzazione Modello e la finestra delle proprietà della risorsa. Ad esempio, quello che segue è un esempio di modello iniziale di una risorsa IaC `Alexa::ASK::Skill` standard:

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

```
S3Bucket: <String>  
S3Key: <String>
```

Un modello iniziale di scheda di risorse IAc standard è costituito da quanto segue:

- Il tipo di CloudFormation risorsa.
- Proprietà obbligatorie o di uso comune.
- Il tipo richiesto di valore da fornire per ogni proprietà.

#### Note

È possibile Amazon Q utilizzarlo per generare suggerimenti sul codice di infrastruttura per schede di risorse standard. Per ulteriori informazioni, consulta [Utilizzo AWS Infrastructure Composer con Amazon Q Developer](#).

## Procedura

È possibile modificare il codice di infrastruttura per ogni risorsa in una scheda componente standard tramite il pannello delle proprietà della risorsa.

Per modificare una scheda componente standard

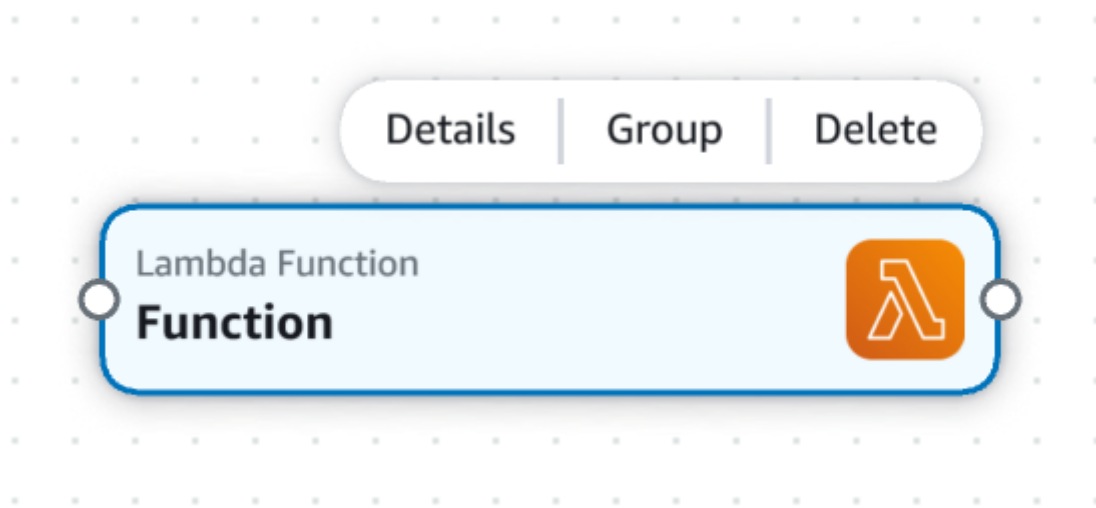
1. Aprire il pannello delle proprietà delle risorse della scheda componente IAc standard.
2. Nel campo Modifica, selezionate la risorsa IAc standard da modificare dall'elenco a discesa.
3. Modifica il codice dell'infrastruttura e salva.

## Eliminare schede in Infrastructure Composer

Questa sezione fornisce istruzioni per eliminare le schede in AWS Infrastructure Composer.

### Schede componenti avanzate

Per eliminare una scheda componente avanzata, seleziona una scheda che hai posizionato nell'area di visualizzazione. Dal menu Azioni sulla scheda, seleziona Elimina.



## Schede componenti standard

Per eliminare le schede componente standard, è necessario rimuovere manualmente il codice di infrastruttura per ogni CloudFormation risorsa dal modello. Di seguito è riportato un modo semplice per eseguire questa operazione:

1. Prendi nota dell'ID logico della risorsa da eliminare.
2. Sul modello, individua la risorsa in base al relativo ID logico `Outputs` nella sezione `Resources` o.
3. Elimina la risorsa dal modello. Ciò include l'ID logico della risorsa e i relativi valori annidati, come `Type` e `Properties`.
4. Controlla la vista Canvas per verificare che la risorsa sia stata rimossa dal tuo canvas.

## Visualizza gli aggiornamenti del codice con Change Inspector in Infrastructure Composer

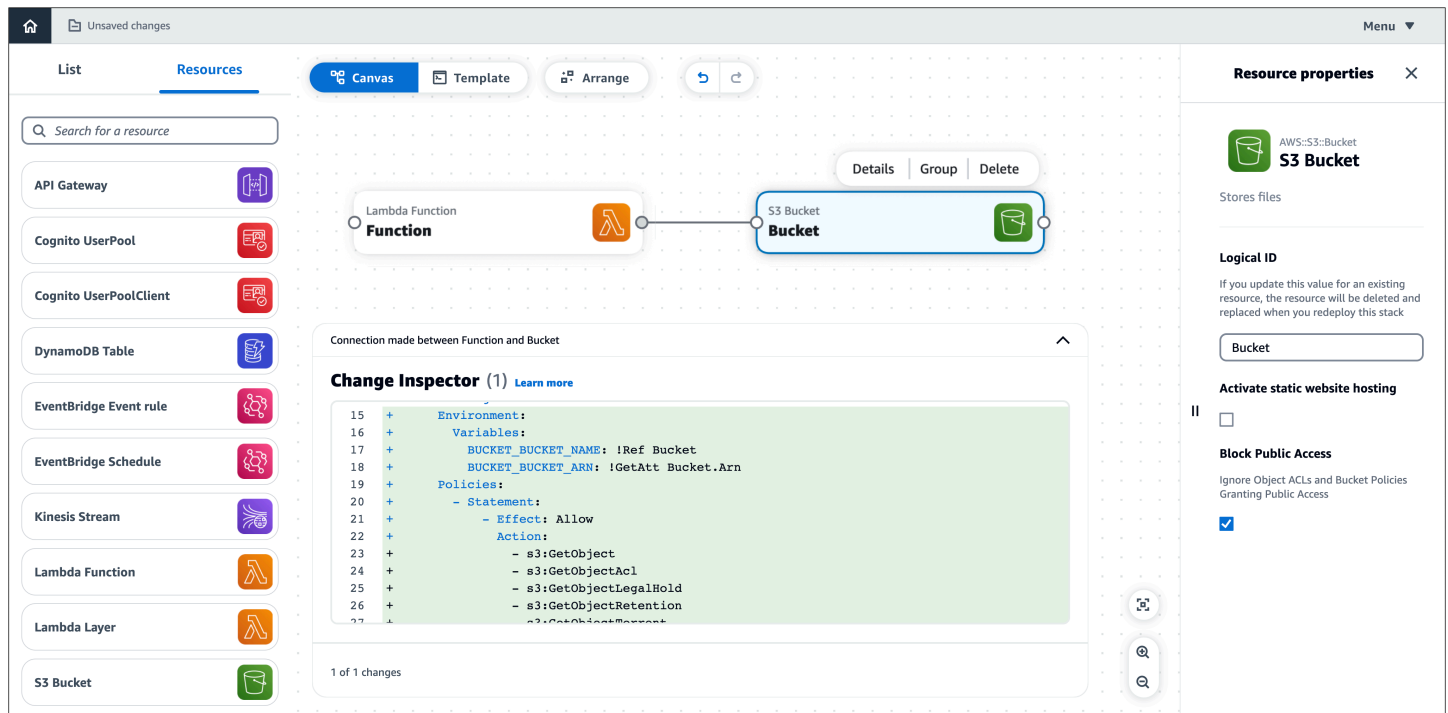
Durante la progettazione nella console Infrastructure Composer, il codice dell'infrastruttura viene creato automaticamente. Usa Change Inspector per visualizzare gli aggiornamenti del codice del modello e scoprire cosa Infrastructure Composer sta creando per te.

In questo argomento viene illustrato l'utilizzo di Infrastructure Composer dall'estensione Console di gestione AWS o dall'estensione AWS Toolkit for Visual Studio Code

Change Inspector è uno strumento visivo all'interno di Infrastructure Composer che mostra gli aggiornamenti recenti del codice.

- Durante la progettazione dell'applicazione, i messaggi vengono visualizzati nella parte inferiore dell'area di visualizzazione. Questi messaggi forniscono commenti sulle azioni che state eseguendo.
- Se supportato, è possibile espandere un messaggio per visualizzare Change Inspector.
- Il Change Inspector mostra le modifiche al codice apportate all'interazione più recente.

L'esempio seguente mostra come funziona Change Inspector:



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

```

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

```

On the right side of the interface, the 'Resource properties' panel for the 'S3 Bucket' is visible, showing options like 'Block Public Access' which is checked.

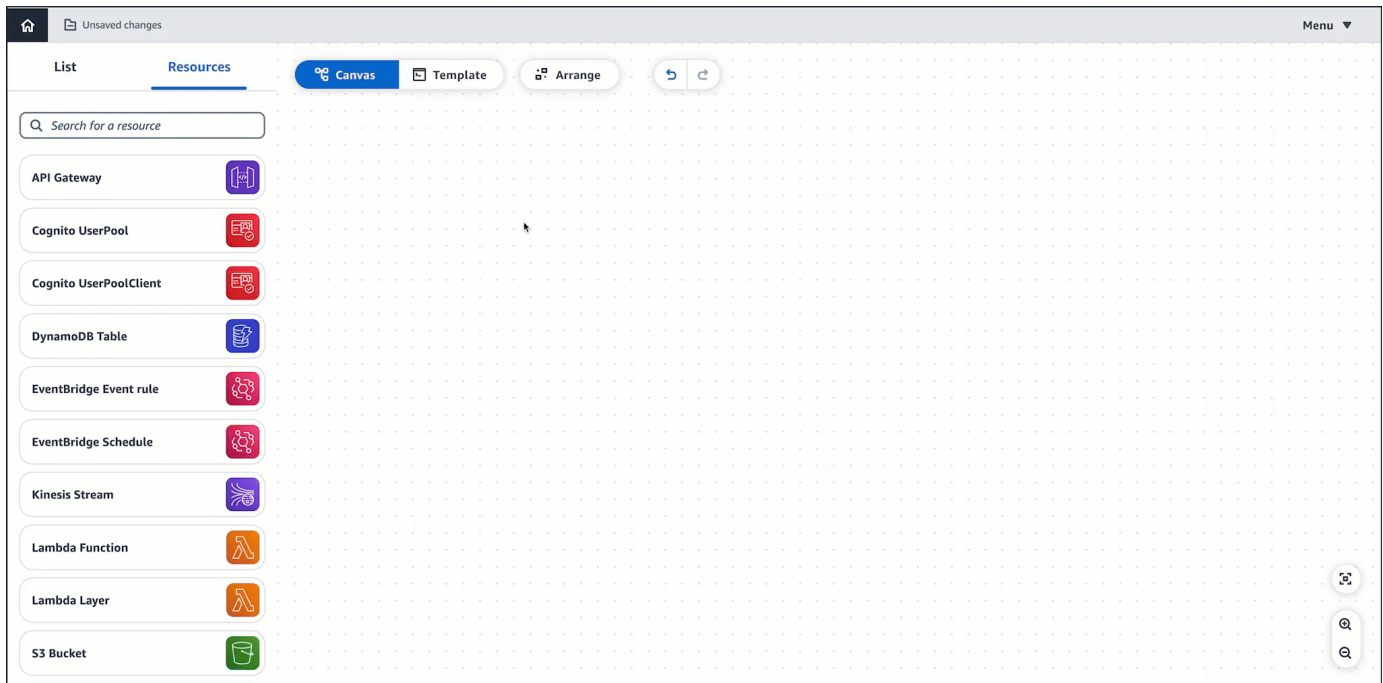
## Vantaggi del Change Inspector

Change Inspector è un ottimo modo per visualizzare il codice modello che Infrastructure Composer crea per te. È anche un ottimo modo per imparare a scrivere codice di infrastruttura. Durante la progettazione di applicazioni in Infrastructure Composer, visualizza gli aggiornamenti del codice in Change Inspector per conoscere il codice necessario per eseguire il provisioning del progetto.

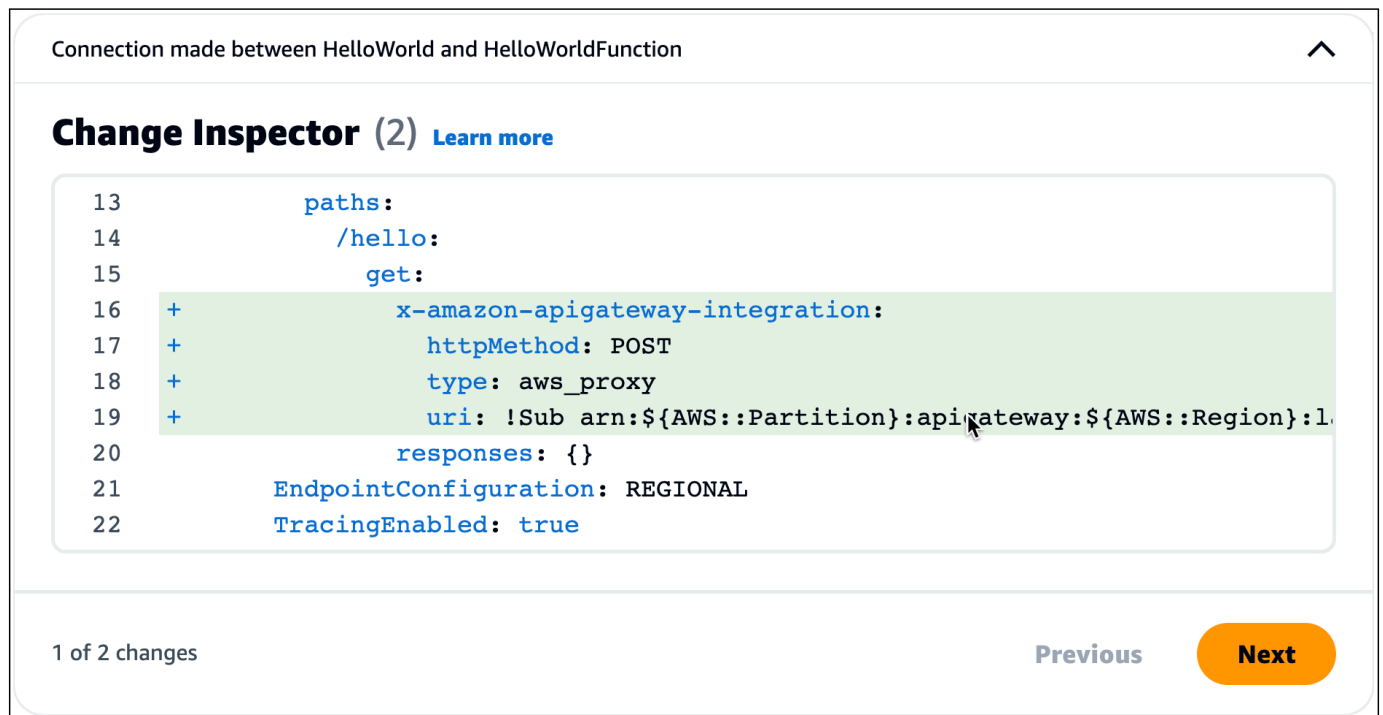
## Procedura

Per utilizzare Change Inspector

1. Espandi un messaggio per visualizzare Change Inspector.



2. Visualizza il codice che è stato composto automaticamente per te.



- Il codice evidenziato in verde indica il codice appena aggiunto.
- Il codice evidenziato in rosso indica il codice appena rimosso.
- I numeri di riga indicano la posizione all'interno del modello.

- Quando più sezioni del modello sono state aggiornate, Change Inspector le organizza. Seleziona i pulsanti Precedente e Avanti per visualizzare tutte le modifiche.



### Note

Per Infrastructure Composer dalla console, è possibile visualizzare le modifiche al codice nel contesto dell'intero modello, utilizzando la visualizzazione modello. Puoi anche sincronizzare Infrastructure Composer con un IDE locale e visualizzare l'intero modello sul tuo computer locale. Per ulteriori informazioni, consulta [Connect la console Infrastructure Composer all'IDE locale](#).

## Ulteriori informazioni

Per ulteriori informazioni sul codice creato da Infrastructure Composer, consulta quanto segue:

- [Connessioni a schede in Infrastructure Composer](#).

## Fai riferimento ai file esterni in Infrastructure Composer

Puoi usare file esterni con i tuoi AWS Serverless Application Model (AWS SAM) modelli per riutilizzare codice ripetuto e organizzare i tuoi progetti. Ad esempio, potresti avere più risorse API REST di Amazon API Gateway descritte da una OpenAPI specifica. Invece di replicare il codice delle OpenAPI specifiche nel modello, puoi creare un file esterno e farvi riferimento per ciascuna delle tue risorse.

AWS Infrastructure Composer supporta i seguenti casi d'uso di file esterni:

- REST API Risorse API Gateway definite da file di OpenAPI specifiche esterne.
- AWS Step Functions risorse delle macchine a stati definite da file di definizione delle macchine a stati esterni.

Per ulteriori informazioni sulla configurazione di file esterni per le risorse supportate, consulta quanto segue:

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

### Note

Per fare riferimento a file esterni con Infrastructure Composer dalla console Infrastructure Composer, è necessario utilizzare Infrastructure Composer in modalità di sincronizzazione locale. Per ulteriori informazioni, consulta [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#).

### Argomenti

- [Procedure consigliate per i file di riferimento esterni di Infrastructure Composer](#)
- [Crea un riferimento a un file esterno in Infrastructure Composer](#)
- [Carica un progetto con un riferimento a un file esterno in Infrastructure Composer](#)
- [Crea un'applicazione che faccia riferimento a un file esterno in Infrastructure Composer](#)
- [Fai riferimento a un file esterno di OpenAPI specifiche con Infrastructure Composer](#)

## Procedure consigliate per i file di riferimento esterni di Infrastructure Composer

### Usa Infrastructure Composer con un IDE locale

Quando si utilizza Infrastructure Composer con un IDE locale in modalità di sincronizzazione locale, è possibile utilizzare l'IDE locale per visualizzare e modificare file esterni. Il contenuto dei file esterni supportati a cui si fa riferimento nel modello verrà aggiornato automaticamente nell'area di disegno di Infrastructure Composer. Per ulteriori informazioni, consulta [Connect la console Infrastructure Composer all'IDE locale](#).

### Conserva i file esterni nella directory principale del progetto

È possibile creare sottodirectory all'interno della directory principale del progetto per organizzare i file esterni. Infrastructure Composer non può accedere ai file esterni archiviati in una directory esterna alla directory principale del progetto.

### Distribuite l'applicazione utilizzando il AWS SAM CLI

Quando si distribuisce l'applicazione su Cloud AWS, i file esterni locali devono prima essere caricati in una posizione accessibile, come Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3). È possibile utilizzare la AWS SAM CLI per facilitare automaticamente questo processo. Per ulteriori informazioni, consulta [Caricare file locali al momento della distribuzione](#) nella Guida per gli AWS Serverless Application Model sviluppatori.

## Crea un riferimento a un file esterno in Infrastructure Composer

È possibile creare un riferimento a un file esterno dal pannello delle proprietà delle risorse supportate.

Per creare un riferimento a un file esterno

1. Da una scheda componente avanzata API Gateway o Step Functions, seleziona Dettagli per visualizzare il pannello delle proprietà delle risorse.
2. Individua e seleziona l'opzione Usa file esterno.
3. Specificate il percorso relativo al file esterno. Questo è il percorso dal `template.yaml` file al file esterno.

Ad esempio, per fare riferimento al file `api-spec.yaml` esterno dalla seguente struttura del progetto, specificate `./api-spec.yaml` come percorso relativo.

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

#### Note

Se il file esterno e il percorso specificato non esistono, Infrastructure Composer lo creerà.

4. Salva le modifiche.

## Carica un progetto con un riferimento a un file esterno in Infrastructure Composer

Segui i passaggi elencati in questa pagina per caricare un progetto Infrastructure Composer con un riferimento a un file esterno.

Dalla console Infrastructure Composer

1. Completa le fasi descritte in [Importa un modello di progetto esistente nella console Infrastructure Composer](#).
2. Conferma che Infrastructure Composer ti richieda di connetterti alla cartella principale del progetto

Se il browser supporta l'API File System Access, Infrastructure Composer richiederà di connettersi alla cartella principale del progetto. Infrastructure Composer aprirà il progetto in modalità di sincronizzazione locale per supportare il file esterno. Se il file esterno di riferimento non è supportato, riceverai un messaggio di errore. Per ulteriori informazioni sui messaggi di errore, vedere [Risoluzione dei problemi](#).

## Dal Toolkit for VS Code

1. Completa le fasi descritte in [Accedi a Infrastructure Composer dal AWS Toolkit for Visual Studio Code](#).
2. Apri il modello che desideri visualizzare in Infrastructure Composer.

Quando accedi a Infrastructure Composer da un modello, Infrastructure Composer rileva automaticamente il tuo file esterno. Se il file esterno di riferimento non è supportato, riceverai un messaggio di errore. Per ulteriori informazioni sui messaggi di errore, vedere [Risoluzione dei problemi](#).

## Crea un'applicazione che faccia riferimento a un file esterno in Infrastructure Composer

Questo esempio utilizza il AWS SAMCLI per creare un'applicazione che fa riferimento a un file esterno per la definizione della macchina a stati. Si carica quindi il progetto in Infrastructure Composer con il file esterno correttamente referenziato.

### Esempio

1. Innanzitutto, utilizzate il `sam init` comando AWS SAM CLI per inizializzare una nuova applicazione denominata. `demo` Durante il flusso interattivo, selezionate il modello di avvio rapido del flusso di lavoro in più fasi.

```
$ 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
```

```
...
```

Questa applicazione fa riferimento a un file esterno per la definizione della macchina a stati.

```
...
```

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

Il file esterno si trova nella `statemachine` sottodirectory della nostra applicazione.

```
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. Quindi, carica l'applicazione in Infrastructure Composer dalla console. Dalla home page di Infrastructure Composer, seleziona Carica un CloudFormation modello.
3. Seleziona la cartella `demo` del nostro progetto e attendi che venga richiesto di visualizzare i file. Seleziona il nostro `template.yaml` file e seleziona Crea. Quando richiesto, seleziona Salva modifiche.

## Open project folder ✕

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

📁 Select folder

✔️ demo

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

template.yaml ▾

Cancel
Create

Infrastructure Composer rileva automaticamente il file di definizione della macchina a stati esterna e lo carica. Seleziona la nostra `StockTradingStateMachinerisorsa` e scegli `Dettagli` per mostrare il pannello delle proprietà della risorsa. Qui puoi vedere che Infrastructure Composer si è connesso automaticamente al nostro file di definizione della macchina a stati esterna.

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' list with various AWS services. The main canvas displays a diagram of resources: Lambda Functions (StockCheckerFunction, StockSellerFunction, StockBuyerFunction), a DynamoDB Table (TransactionTable), and an EventBridge Timer (ImplicitTimer). A 'Details' panel is open for the 'StockTradingStateMachine' resource, showing its state machine definition. The right sidebar displays the 'Resource properties' for the 'Step Functions State machine'.

**Resource properties**

**AWS::Serverless::StateMachine**  
**Step Functions State machine**

Step Function State machine

**Logical ID**

If you update this value for an existing resource, the resource will be deleted and replaced when you redeploy this stack

StockTradingStateMachine

**State machine definition**

```

Comment: A state machine
StartAt: Check Stock Value
States:
  Check Stock Value:
    Type: Task
    Resource: ${StockChec
  Retry:
    - ErrorEquals:
      - States.TaskFa
      IntervalSeconds:
      MaxAttempts: 5
      BackoffRate: 1.5
    Next: Buy or Sell
  
```

Use external file for state machine definition

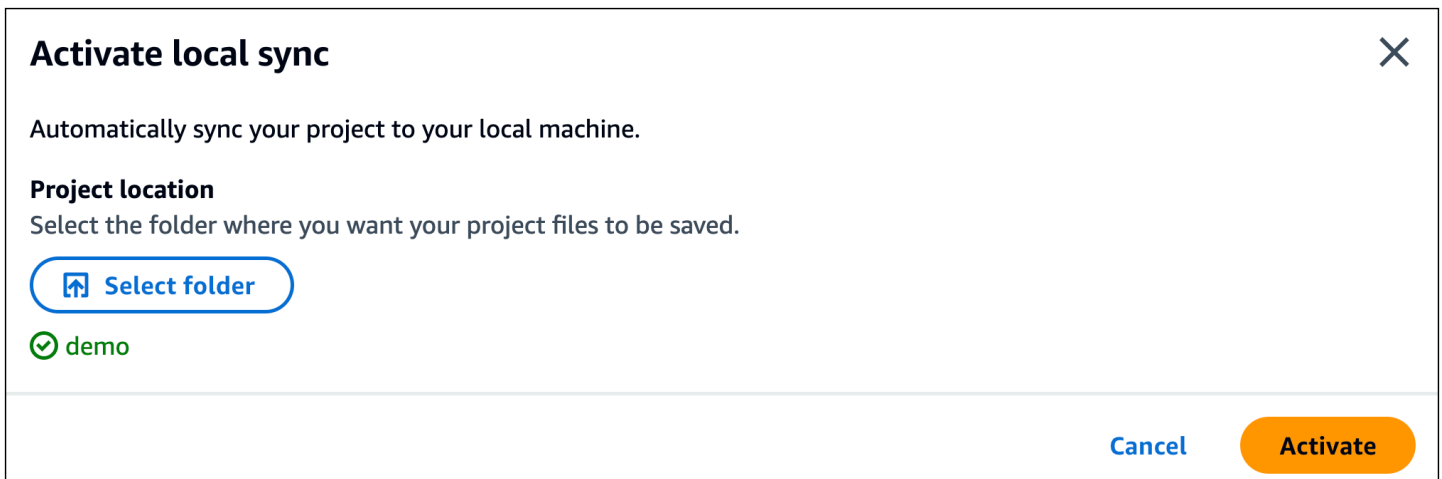
Qualsiasi modifica apportata al file di definizione della macchina a stati si rifletterà automaticamente in Infrastructure Composer.

## Fai riferimento a un file esterno di OpenAPI specifiche con Infrastructure Composer

Questo esempio utilizza Infrastructure Composer dalla console per fare riferimento a un file di OpenAPI specifiche esterno che definisce un API Gateway REST API.

Innanzitutto, crea un nuovo progetto dalla home page di Infrastructure Composer.

Quindi, attiva la sincronizzazione locale selezionando Attiva sincronizzazione locale dal menu. Crea una nuova cartella denominata demo, consenti alla richiesta di visualizzare i file e seleziona Attiva. Quando richiesto, seleziona Salva modifiche.



Successivamente, trascina una scheda Amazon API Gateway sull'area di disegno. Seleziona Dettagli per visualizzare il pannello delle proprietà delle risorse.

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' panel with a search bar and a list of AWS services including API Gateway, Cognito UserPool, Cognito UserPoolClient, DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, Lambda Function, Lambda Layer, and S3 Bucket. The main canvas displays an API Gateway resource named 'Api' with a 'GET /' endpoint. The right-hand side shows the 'Resource properties' panel for the 'Api Gateway' resource. The 'Logical ID' is 'Api'. The 'Default authorizer' is set to 'None'. The 'Method' is 'GET'. The 'Path' is currently empty.

Dal pannello delle proprietà delle risorse, configura quanto segue e salva.

- Seleziona l'opzione Usa file esterno per la definizione dell'API.
- Immissione `./api-spec.yaml` come percorso relativo al file esterno

## Use external file for api definition



## Relative path to external file

`./api-spec.yaml`

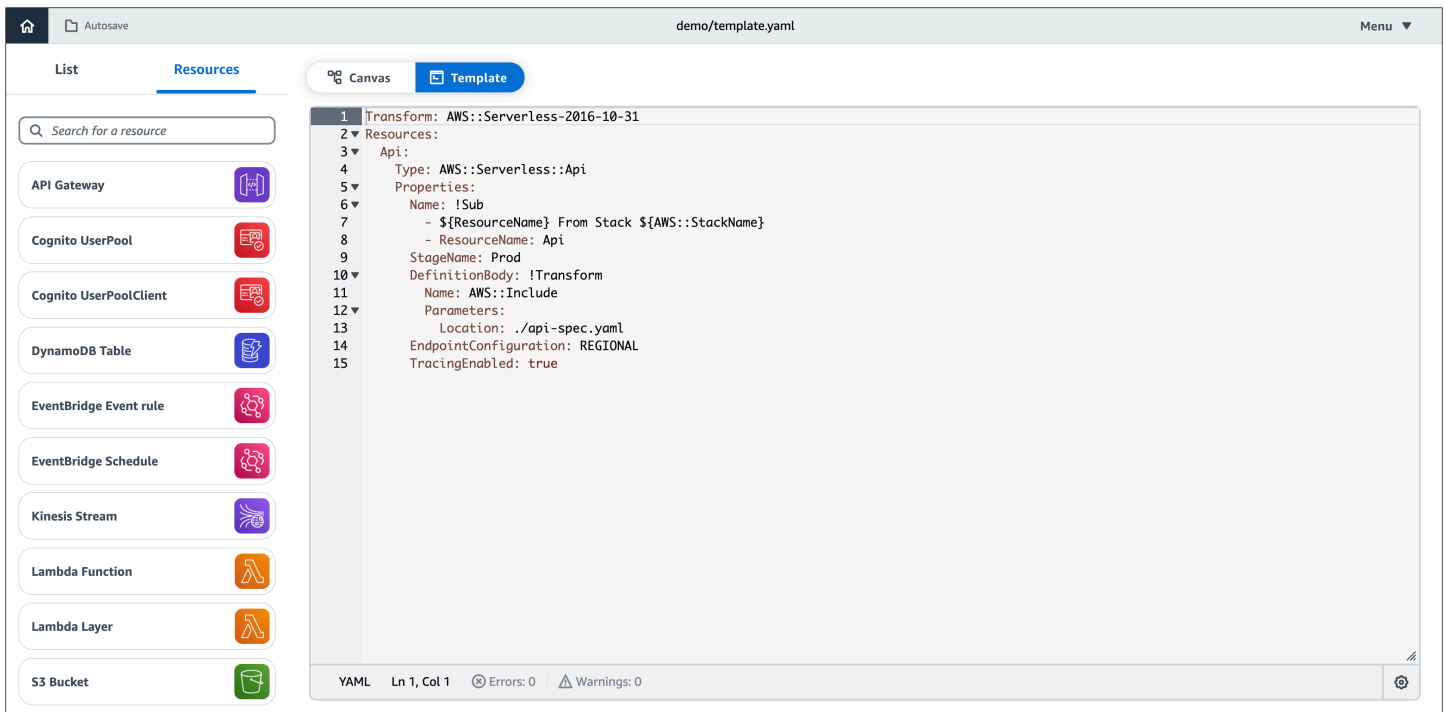
Questo crea la seguente directory sul nostro computer locale:

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

Ora puoi configurare il file esterno sul nostro computer locale. Usando il nostro IDE, apri il file che `api-spec.yaml` si trova nella cartella del tuo progetto. Sostituisci il suo contenuto con quanto segue:

```
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
```

Nella vista Modello di Infrastructure Composer, puoi vedere che Infrastructure Composer ha aggiornato automaticamente il modello in modo che faccia riferimento al file esterno.

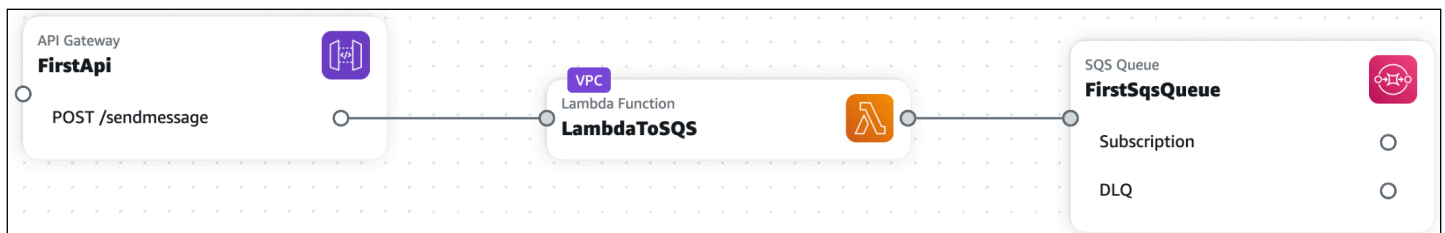


## Integra Infrastructure Composer con Amazon Virtual Private Cloud (Amazon VPC)

AWS Infrastructure Composer presenta un'integrazione con il servizio Amazon Virtual Private Cloud (Amazon VPC). Utilizzando Infrastructure Composer, puoi fare quanto segue:

- Identifica le risorse sulla tela che si trovano in un VPC tramite un tag VPC visivo.
- Configura AWS Lambda le funzioni con VPCs un modello esterno.

L'immagine seguente mostra un esempio di applicazione con una funzione Lambda configurata con un VPC.



Per ulteriori informazioni su Amazon VPC, consulta [Cos'è Amazon VPC?](#) nella Guida per l'utente di Amazon VPC.

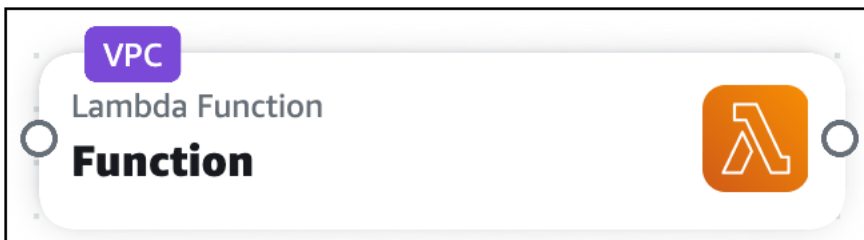
## Argomenti

- [Identifica le risorse di Infrastructure Composer e le informazioni correlate in un VPC](#)
- [Configura le funzioni Lambda con funzioni esterne VPCs in Infrastructure Composer](#)
- [Parametri nei modelli importati per un VPC esterno con Infrastructure Composer](#)
- [Aggiungere nuovi parametri ai modelli importati con Infrastructure Composer](#)
- [Configura una funzione Lambda e un VPC definiti in un altro modello con Infrastructure Composer](#)

## Identifica le risorse di Infrastructure Composer e le informazioni correlate in un VPC

Per integrare Infrastructure Composer con Amazon VPC, devi prima identificare le risorse in un VPC e le informazioni necessarie per completare un'integrazione. Ciò include anche informazioni di configurazione relative a gruppi di sicurezza, identificatori di sottorete, tipi di parametri, tipi SSM, tipi di valori statici.

Infrastructure Composer visualizza le risorse in un VPC utilizzando un tag VPC. Questo tag viene applicato alle carte sulla tela. Di seguito è riportato un esempio di funzione Lambda con un tag VPC:



I tag VPC vengono applicati alle schede sull'area di disegno quando si eseguono le seguenti operazioni:

- Configura una funzione Lambda con un VPC in Infrastructure Composer.
- Importa un modello che contiene risorse configurate con un VPC.

## Identificatori di gruppi di sicurezza e sottoreti

Una funzione Lambda può essere configurata con più gruppi di sicurezza e sottoreti. Per configurare un gruppo di sicurezza o una sottorete per una funzione Lambda, fornite un valore e un tipo.

- Valore: un identificatore per il gruppo di sicurezza o la sottorete. I valori accettati variano in base al tipo.

- Tipo: sono consentiti i seguenti tipi di valori:
  - Nome del parametro
  - AWS Systems Manager (SSM) Archivio parametri
  - Valore statico

## Tipo parametro

La `Parameters` sezione di un AWS CloudFormation modello può essere utilizzata per archiviare informazioni sulle risorse su più modelli. Per ulteriori informazioni sui parametri, consultate [Parametri](#) nella Guida AWS CloudFormation per l'utente.

Per il tipo di parametro, è possibile fornire un nome di parametro. Nell'esempio seguente, forniamo un valore per il nome del `PrivateSubnet1` parametro:

**Subnet IDs**

List of VPC subnet identifiers

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

Quando fornite un nome di parametro, Infrastructure Composer lo definisce nella `Parameters` sezione del modello. Quindi, Infrastructure Composer fa riferimento al parametro nella risorsa della funzione Lambda. Di seguito è riportato un esempio:

```

...
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 SSM

SSM Parameter Store fornisce uno storage sicuro e gerarchico per la gestione dei dati di configurazione e la gestione dei segreti. Per ulteriori informazioni, consulta [Archivio dei parametri AWS Systems Manager](#) nella Guida per l'utente di AWS Systems Manager .

Per il tipo SSM, puoi fornire i seguenti valori:

- Riferimento dinamico a un valore dall'archivio dei parametri SSM.
- ID logico di una `AWS::SSM::Parameter` risorsa definita nel modello.

### Riferimento dinamico

È possibile fare riferimento a un valore da SSM Parameter Store utilizzando un riferimento dinamico nel seguente formato: `{{resolve:ssm:reference-key}}` Per ulteriori informazioni, consulta i [parametri SSM nella Guida](#) per l'AWS CloudFormation utente.

Infrastructure Composer crea il codice dell'infrastruttura per configurare la funzione Lambda con il valore di SSM Parameter Store. Di seguito è riportato un esempio:

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

### ID logico

È possibile fare riferimento a una `AWS::SSM::Parameter` risorsa nello stesso modello tramite ID logico.

Di seguito è riportato un esempio di `AWS::SSM::Parameter` risorsa denominata `PrivateSubnet1Parameter` che memorizza l'ID di sottorete per `PrivateSubnet1`:

```
...
Resources:
```

```

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

```

Di seguito è riportato un esempio di questo valore di risorsa fornito dall'ID logico per la funzione Lambda:

### Subnet IDs

List of VPC subnet identifiers

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

Infrastructure Composer crea il codice dell'infrastruttura per configurare la funzione Lambda con il parametro SSM:

```

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

```

## Tipo di valore statico

Quando viene distribuito un gruppo di sicurezza o una sottorete CloudFormation, viene creato un valore ID. È possibile fornire questo ID come valore statico.

Per il tipo di valore statico, sono validi i seguenti valori:

- Per i gruppi di sicurezza, fornire `GroupId`. Per ulteriori informazioni, consulta [Restituisci valori](#) nella Guida AWS CloudFormation per l'utente. Di seguito è riportato un esempio: `sg-0b61d5c742dc2c773`.
- Per le sottoreti, fornisci il `SubnetId`. Per ulteriori informazioni, consulta [Restituisci valori nella Guida](#) per l'AWS CloudFormation utente. Di seguito è riportato un esempio: `subnet-01234567890abcdef`.

Infrastructure Composer crea il codice dell'infrastruttura per configurare la funzione Lambda con il valore statico. Di seguito è riportato un esempio:

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

## Utilizzo di più tipi

Per i gruppi di sicurezza e le sottoreti, puoi utilizzare più tipi insieme. Di seguito è riportato un esempio che configura tre gruppi di sicurezza per una funzione Lambda fornendo valori di tipo diverso:

### Security group IDs

List of VPC security group identifiers

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

Infrastructure Composer fa riferimento a tutti e tre i valori all'interno della proprietà: `SecurityGroupIds`

```

...
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

```

## Configura le funzioni Lambda con funzioni esterne VPCs in Infrastructure Composer

Per iniziare a configurare una funzione Lambda con un VPC definito su un altro modello, usa la scheda componente avanzata Lambda Function. Questa scheda rappresenta una funzione Lambda che utilizza il tipo di `AWS::Serverless::Function` risorsa AWS Serverless Application Model (AWS SAM).

Per configurare una funzione Lambda con un VPC da un modello esterno

1. Dal pannello delle proprietà delle risorse della funzione Lambda, espandi la sezione a discesa delle impostazioni VPC (avanzate).
2. Seleziona Assegna a VPC esterno.
3. Fornisci valori per i gruppi di sicurezza e le sottoreti da configurare per la funzione Lambda. Per informazioni dettagliate, vedi [Identificatori di gruppi di sicurezza e sottoreti](#).
4. Salva le modifiche.

## Parametri nei modelli importati per un VPC esterno con Infrastructure Composer

Quando importi un modello esistente con parametri definiti per i gruppi di sicurezza e le sottoreti di un VPC esterno, Infrastructure Composer fornisce un elenco a discesa da cui selezionare i parametri.

Di seguito è riportato un esempio della `Parameters` sezione di un modello importato:

```

...
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
...

```

Quando si configura un VPC esterno per una nuova funzione Lambda sull'area di disegno, questi parametri saranno disponibili da un elenco a discesa. Di seguito è riportato un esempio:

### Subnet IDs

List of VPC subnet identifiers

Value	Type
<input style="width: 90%; border: none;" type="text" value="🔍  "/>	Parameter <span style="float: right;">▼</span>
VPCSubnets	
VPCSubnet	

## Limitazioni all'importazione dei tipi di parametri dell'elenco

Normalmente, è possibile specificare più identificatori di gruppi di sicurezza e sottorete per ogni funzione Lambda. Se il modello esistente contiene tipi di parametri di elenco, ad esempio `List<AWS::EC2::SecurityGroup::Id>` o `List<AWS::EC2::Subnet::Id>`, è possibile specificare un solo identificatore.

Per ulteriori informazioni sul tipo di elenchi di parametri, [consultate Tipi di parametri AWS specifici supportati nella Guida](#) per l'AWS CloudFormation utente.

Di seguito è riportato un esempio di modello che `VPCSecurityGroups` definisce un tipo di parametro di elenco:

```

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

```

In Infrastructure Composer, se si seleziona il `VPCSecurityGroups` valore come identificatore del gruppo di sicurezza per una funzione Lambda, verrà visualizzato il seguente messaggio:

### Security group IDs

List of VPC security group identifiers

Value	Type
<input type="text" value="VPCSecurityGroups"/>	<input type="text" value="Parameter"/>

[Add new item](#)

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

Questa limitazione si verifica perché le `SubnetIds` proprietà `SecurityGroupIds` e di un `AWS::Lambda::Function VpcConfig` oggetto accettano entrambe solo un elenco di valori di stringa. Poiché un singolo tipo di parametro di elenco contiene un elenco di stringhe, può essere l'unico oggetto fornito quando specificato.

Per i tipi di parametri di elenco, il seguente è un esempio di come vengono definiti nel modello quando configurati con una funzione 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

```

## Aggiungere nuovi parametri ai modelli importati con Infrastructure Composer

Quando importate un modello esistente con parametri definiti, potete anche creare nuovi parametri. Invece di selezionare un parametro esistente dall'elenco a discesa, fornite un nuovo tipo e valore. Di seguito è riportato un esempio che crea un nuovo parametro denominato `MySecurityGroup`:

### Security group IDs

List of VPC security group identifiers

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

Per tutti i nuovi valori forniti nel pannello delle proprietà delle risorse per la funzione Lambda, Infrastructure Composer li definisce in un elenco sotto le `SubnetIds` proprietà `SecurityGroupIds` o di una funzione Lambda. Di seguito è riportato un esempio:

```

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

```

Se desideri fare riferimento all'ID logico di un tipo di parametro di elenco da un modello esterno, ti consigliamo di utilizzare la vista `Modello` e modificare direttamente il modello. L'ID logico di un tipo di parametro di elenco deve essere sempre fornito come valore singolo e come unico valore.

```

...
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

```

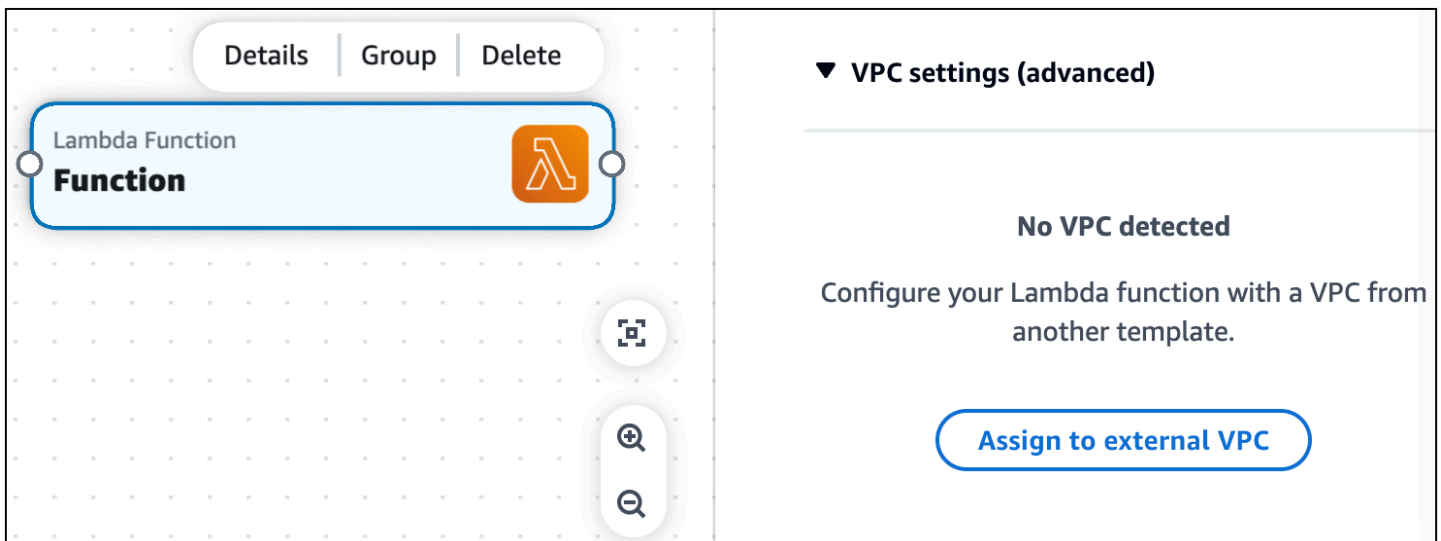
## Configura una funzione Lambda e un VPC definiti in un altro modello con Infrastructure Composer

In questo esempio, configuriamo una funzione Lambda in Infrastructure Composer con un VPC definito su un altro modello.

Iniziamo trascinando una scheda componente avanzata della funzione Lambda sull'area di lavoro.

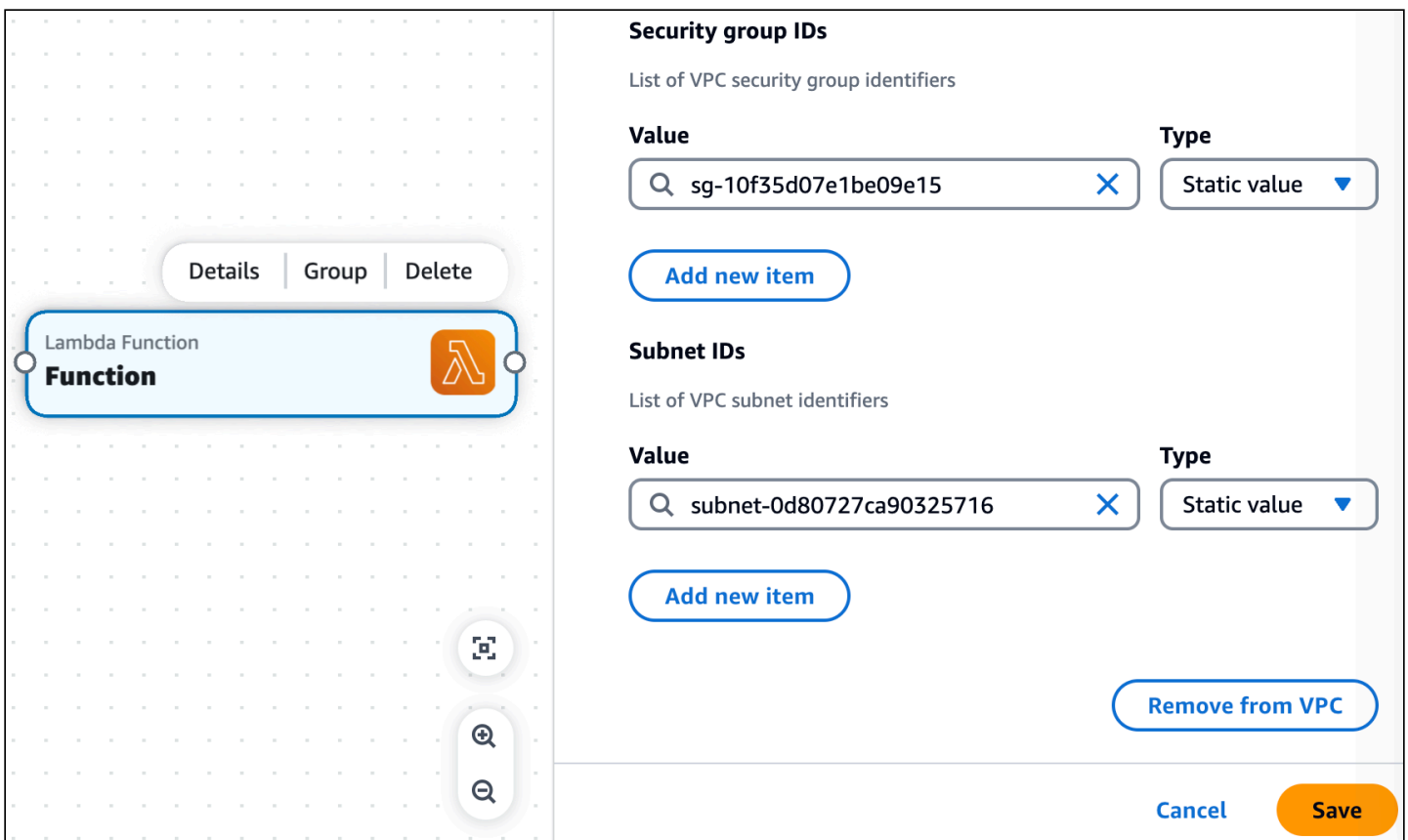


Successivamente, apriamo il pannello delle proprietà delle risorse della scheda ed espandiamo la sezione a discesa delle impostazioni VPC (avanzate).

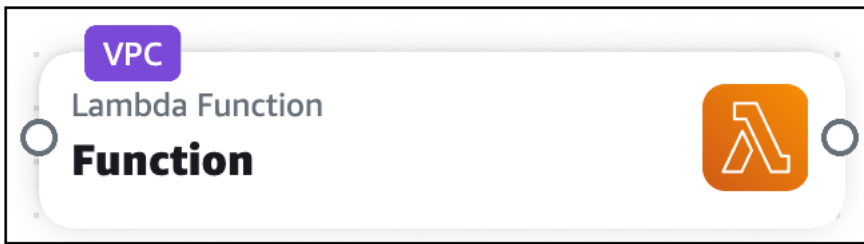


Successivamente, selezioniamo Assegna a VPC esterno per iniziare a configurare un VPC da un modello esterno.

In questo esempio, facciamo riferimento a un ID di gruppo di sicurezza e a un ID di sottorete. Questi valori vengono creati quando viene distribuito il modello che definisce il VPC. Scegliamo il tipo di valore statico e inseriamo il valore del nostro. IDs Selezioniamo Salva quando hai finito.



Ora che la nostra funzione Lambda è configurata con il nostro VPC, il tag VPC viene visualizzato sulla nostra scheda.



Infrastructure Composer ha creato il codice dell'infrastruttura per configurare la nostra funzione Lambda con il gruppo di sicurezza e la sottorete del VPC esterno.

```
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}
```

# Distribuisci la tua applicazione serverless Infrastructure Composer nel cloud AWS

AWS Infrastructure Composer Da utilizzare per progettare applicazioni serverless pronte per l'implementazione. Per la distribuzione, utilizza qualsiasi servizio compatibile. AWS CloudFormation Ti consigliamo di utilizzare [AWS Serverless Application Model \(AWS SAM\)](#).

AWS SAM è un framework open source che fornisce strumenti di sviluppo per la creazione e l'esecuzione di applicazioni serverless su. AWS Con AWS SAM la sua sintassi abbreviata, gli sviluppatori dichiarano CloudFormation risorse e risorse serverless specializzate che vengono trasformate in infrastruttura durante la distribuzione.

## Concetti importanti AWS SAM

Prima di utilizzarlo AWS SAM, è importante acquisire familiarità con alcuni dei suoi concetti fondamentali.

- [Come AWS SAM funziona](#): questo argomento, contenuto nella Guida per gli AWS Serverless Application Model sviluppatori, fornisce informazioni importanti sui componenti principali utilizzati per creare un'applicazione serverless: The AWS SAMCLI, il AWS SAM progetto e il template. AWS SAM
- [Come usare AWS Serverless Application Model \(AWS SAM\)](#): questo argomento, contenuto nella Guida per gli AWS Serverless Application Model sviluppatori, fornisce una panoramica di alto livello dei passaggi da completare AWS SAM per distribuire l'applicazione nel cloud. AWS

Durante la progettazione dell'applicazione in Infrastructure Composer, è possibile utilizzare il `sam sync` comando per fare in modo che rilevi AWS SAMCLI automaticamente le modifiche locali e le distribuisca. CloudFormation Per ulteriori informazioni, consulta [Using sam sync](#) nella AWS Serverless Application Model Developer Guide.

## Fasi successive

Consulta [Configurato per la distribuzione con AWS SAMCLI e Infrastructure Composer](#) per prepararsi alla distribuzione dell'applicazione.

# Configurato per la distribuzione con AWS SAMCLI e Infrastructure Composer

Per distribuire la tua applicazione con AWS SAM, devi prima installare e accedere a. AWSCLI AWS SAMCLI Gli argomenti di questa sezione forniscono dettagli su questa operazione.

## Installa il AWSCLI

Si consiglia di installare e configurare AWSCLI prima di installare il AWS SAMCLI. Per istruzioni, consulta [Installazione o aggiornamento alla versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente](#).

### Note

Dopo aver installato AWSCLI, è necessario configurare AWS le credenziali. Per ulteriori informazioni, consulta [Configurazione rapida](#) nella Guida per l'AWS Command Line Interface utente.

## Installa il AWS SAMCLI

Per installare AWS SAMCLI, consulta [Installazione di AWS SAMCLI nella Guida per gli AWS Serverless Application Model sviluppatori](#).

## Accedi a AWS SAMCLI

Se si utilizza Infrastructure Composer di Console di gestione AWS, sono disponibili le seguenti opzioni per utilizzare il. AWS SAMCLI

### Attiva la modalità di sincronizzazione locale

Con la modalità di sincronizzazione locale, la cartella del progetto, incluso il AWS SAM modello, viene salvata automaticamente sul computer locale. Infrastructure Composer struttura la directory del progetto in modo tale da AWS SAM riconoscere. Puoi eseguirla AWS SAMCLI dalla directory principale del tuo progetto.

Per ulteriori informazioni sulla modalità di sincronizzazione locale, vedere [Sincronizza e salva localmente il tuo progetto nella console Infrastructure Composer](#).

## Esporta il tuo modello

È possibile esportare il modello sul computer locale. Quindi, esegui il file AWS SAMCLI dalla cartella principale che contiene il modello. Puoi anche utilizzare l'opzione `--template-file` con qualsiasi AWS SAMCLI comando e fornire il percorso del modello.

## Usa Infrastructure Composer dal AWS Toolkit for Visual Studio Code

È possibile utilizzare Infrastructure Composer dal Toolkit for VS Code per portare Infrastructure Composer sul computer locale. Quindi, usa Infrastructure Composer e VS Code. AWS SAMCLI

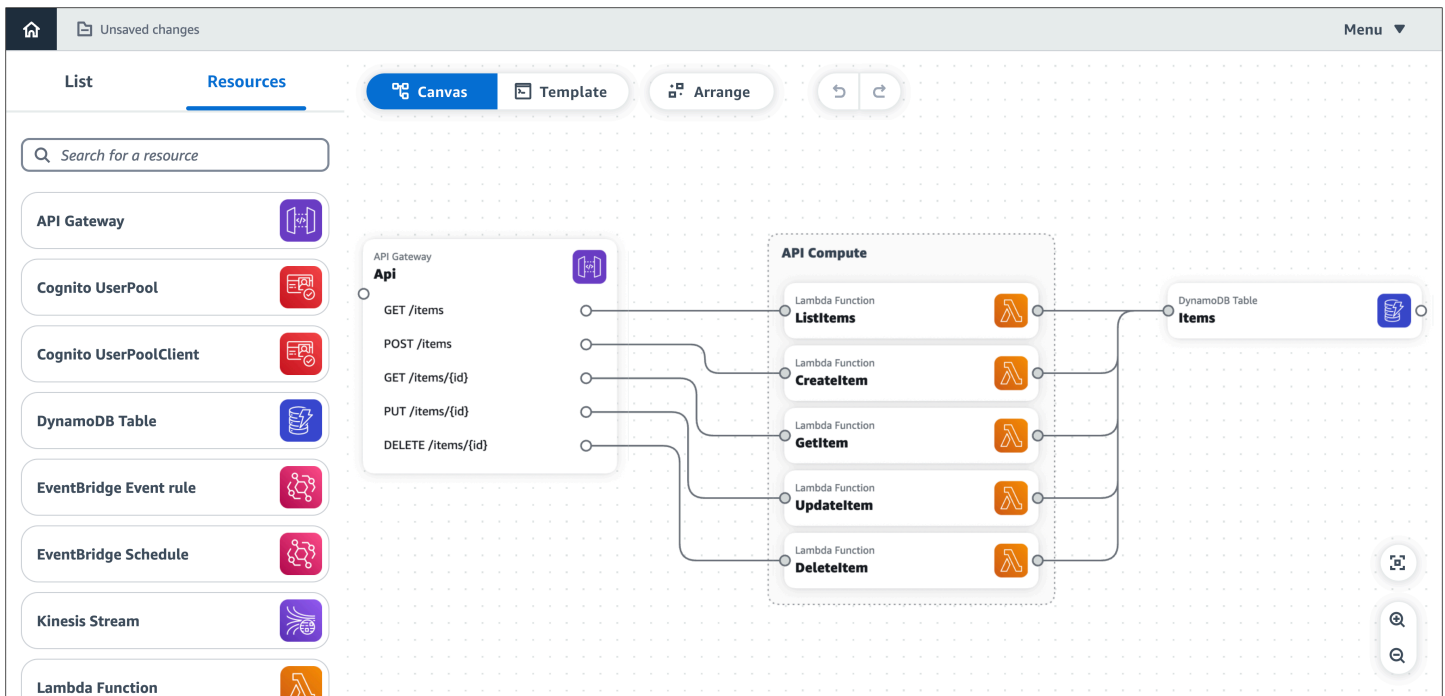
## Fasi successive

Per distribuire l'applicazione, fare riferimento a [Usa Infrastructure Composer con AWS SAM per creare e distribuire](#)

## Usa Infrastructure Composer con AWS SAM per creare e distribuire

Ora che hai completato [Configurato per la distribuzione con AWS SAMCLI e Infrastructure Composer](#), puoi distribuire la tua applicazione con Infrastructure AWS SAM Composer. Questa sezione fornisce un esempio che descrive in dettaglio come eseguire questa operazione. Puoi anche fare riferimento alla sezione [Distribuisci l'applicazione e le risorse AWS SAM nella Guida per gli AWS Serverless Application Model sviluppatori per istruzioni su come distribuire l'applicazione con. AWS SAM](#)

Questo esempio mostra come creare e distribuire l'applicazione demo Infrastructure Composer. L'applicazione demo dispone delle seguenti risorse:



### Note

- Per ulteriori informazioni sull'applicazione demo, consulta. [Caricare e modificare il progetto demo di Infrastructure Composer](#)
- Per questo esempio, utilizziamo Infrastructure Composer con la sincronizzazione locale attivata.

1. Usa il `sam build` comando per creare l'applicazione.

```
$ 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
```

Quindi AWS SAMCLI crea la `./aws-sam` directory nella cartella del progetto. Questa directory contiene elementi di compilazione per le funzioni Lambda dell'applicazione. Ecco un output della directory del progetto:

```
.
### 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. Ora l'applicazione è pronta per essere distribuita. Useremo `sam deploy --guided`. Questo prepara l'applicazione per la distribuzione tramite una serie di istruzioni.

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

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

Setting default arguments for 'sam deploy'
=====
Stack Name [aws-app-composer-basic-api]: AWS Region [us-west-2]:
```

```

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

```

AWS SAMCLIVisualizza un riepilogo di ciò che verrà distribuito:

```

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 SAMCLIdistribuisce l'applicazione, innanzitutto creando un CloudFormation changeset:

```

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

Waiting for changeset to be created..
CloudFormation stack changeset
-----

```

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

+ Add	UpdateItem
AWS::Lambda::Function	N/A

-----

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

Quindi, AWS SAMCLI distribuisce l'applicazione:

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
CREATE_COMPLETE	AWS::DynamoDB::Table		Items
CREATE_IN_PROGRESS	AWS::IAM::Role		
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
CREATE_IN_PROGRESS	AWS::IAM::Role	UpdateItemRole	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::IAM::Role	CreateItemRole	Resource creation Initiated
CREATE_COMPLETE	AWS::IAM::Role	DeleteItemRole	

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

CREATE_IN_PROGRESS	AWS::Lambda::Permission	
GetItemApiGETItemsidPermissionPro	-	d
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
ListItemsApiGETItemsPermissionPro	-	d
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
DeleteItemApiDELETEItemsidPermiss	-	ionProd
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	
ApiDeploymentccc153d135b	-	
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
UpdateItemApiPUTItemsidPermission	-	Prod
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
CreateItemApiPOSTItemsPermissionP	-	rod
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
GetItemApiGETItemsidPermissionPro	Resource creation Initiated	d
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
UpdateItemApiPUTItemsidPermission	Resource creation Initiated	Prod
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
CreateItemApiPOSTItemsPermissionP	Resource creation Initiated	rod
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
ListItemsApiGETItemsPermissionPro	Resource creation Initiated	d
CREATE_IN_PROGRESS	AWS::Lambda::Permission	
DeleteItemApiDELETEItemsidPermiss	Resource creation Initiated	ionProd
CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	
ApiDeploymentccc153d135b	Resource creation Initiated	
CREATE_COMPLETE	AWS::ApiGateway::Deployment	
ApiDeploymentccc153d135b	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::Stage	
ApiProdStage	-	
CREATE_IN_PROGRESS	AWS::ApiGateway::Stage	
ApiProdStage	Resource creation Initiated	
CREATE_COMPLETE	AWS::ApiGateway::Stage	
ApiProdStage	-	
CREATE_COMPLETE	AWS::Lambda::Permission	
CreateItemApiPOSTItemsPermissionP	-	

```

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

```

Infine, viene visualizzato un messaggio che informa che la distribuzione è avvenuta con successo:

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

## Usa Infrastructure Composer con AWS SAM per eliminare uno stack

Questo esempio mostra come eliminare uno CloudFormation stack utilizzando il `sam delete` comando.

Immettete il comando `sam delete` AWS SAMCLI e confermate se desiderate eliminare lo stack e il modello:

```

$ 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 risoluzione dei problemi

Gli argomenti di questa sezione forniscono indicazioni sulla risoluzione dei messaggi di errore durante l'utilizzo AWS Infrastructure Composer.

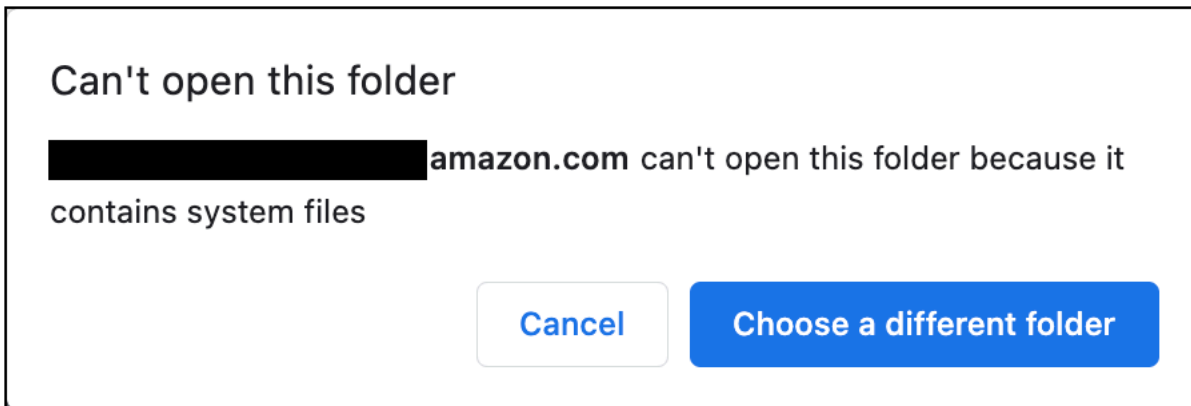
Argomenti

- [Messaggi di errore](#)

## Messaggi di errore

«Impossibile aprire questa cartella»

Esempio di errore:



Possibile causa: Infrastructure Composer non è in grado di accedere a una directory sensibile utilizzando la modalità di sincronizzazione locale.

Per ulteriori informazioni su questo errore, consulta [Data Infrastructure Composer ottiene l'accesso a.](#)

Prova a connetterti a una directory locale diversa o a utilizzare Infrastructure Composer con la sincronizzazione locale disattivata.

«Modello incompatibile»

Esempio di errore: quando si carica un nuovo progetto in Infrastructure Composer, viene visualizzato quanto segue:

Possibile causa: il progetto contiene un file referenziato esternamente che non è supportato in Infrastructure Composer.

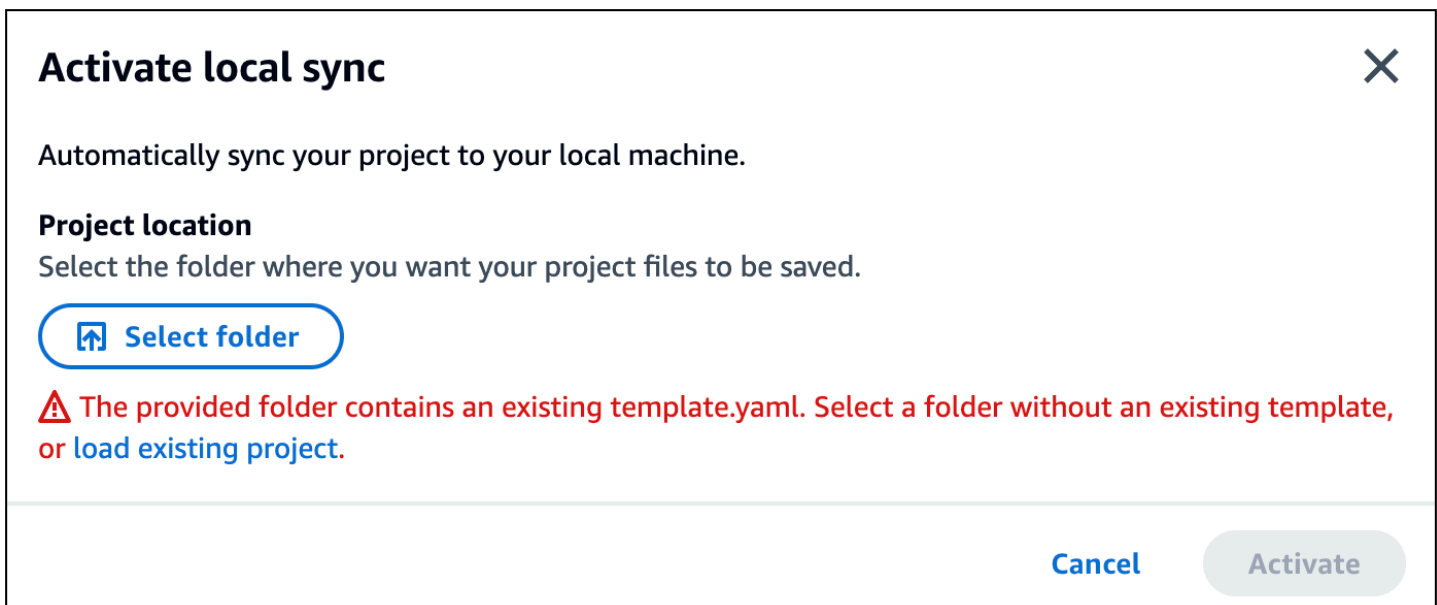
Per informazioni sui file esterni supportati in Infrastructure Composer, consulta. [File esterni di riferimento](#)

Possibile causa: il progetto si collega a un file esterno in una directory locale diversa.

Spostate il file referenziato esternamente in una sottodirectory della directory selezionata per l'utilizzo con la modalità di sincronizzazione locale di Infrastructure Composer.

## «La cartella fornita contiene un template.yaml esistente»

Quando si tenta di attivare la sincronizzazione locale, viene visualizzato il seguente errore:



Possibile causa: la cartella selezionata contiene già un file template.yaml.

Seleziona un'altra directory che non contenga un modello di applicazione o crea una nuova directory.

«Il tuo browser non dispone delle autorizzazioni per salvare il progetto in quella cartella...»

Possibile causa: Infrastructure Composer non è in grado di accedere a una directory sensibile utilizzando la modalità di sincronizzazione locale.

Per ulteriori informazioni su questo errore, consulta [Data Infrastructure Composer ottiene l'accesso a.](#)

Prova a connetterti a una directory locale diversa o usa Infrastructure Composer con la sincronizzazione locale disattivata.

# Sicurezza in AWS Infrastructure Composer

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS Infrastructure Composer, consulta [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza Infrastructure Composer. I seguenti argomenti mostrano come configurare Infrastructure Composer per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse di Infrastructure Composer.

## Argomenti

- [Protezione dei dati in AWS Infrastructure Composer](#)
- [AWS Identity and Access Management per AWS Infrastructure Composer](#)
- [Convalida della conformità per AWS Infrastructure Composer](#)
- [Resilienza in AWS Infrastructure Composer](#)

## Protezione dei dati in AWS Infrastructure Composer

Il modello di [responsabilità AWS condivisa modello](#) di di si applica alla protezione dei dati in AWS Infrastructure Composer. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo

dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per maggiori informazioni sulla privacy dei dati, consulta le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [AWS Modello di responsabilità condivisa e GDPR](#) nel AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando si lavora con Infrastructure Composer o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

**Note**

Tutti i dati immessi in Infrastructure Composer vengono utilizzati al solo scopo di fornire funzionalità all'interno di Infrastructure Composer e generare file e directory di progetto che vengono salvati localmente sul computer. Infrastructure Composer non salva, archivia o trasmette nessuno di questi dati.

## Crittografia dei dati

Infrastructure Composer non crittografa i contenuti dei clienti poiché i dati non vengono salvati, archiviati o trasmessi.

### Crittografia dei dati a riposo

Infrastructure Composer non crittografa i contenuti dei clienti poiché i dati non vengono salvati, archiviati o trasmessi.

### Crittografia dei dati in transito

Infrastructure Composer non crittografa i contenuti dei clienti poiché i dati non vengono salvati, archiviati o trasmessi.

## Gestione delle chiavi

Infrastructure Composer non supporta la gestione delle chiavi poiché i contenuti dei clienti non vengono salvati, archiviati o trasmessi.

## Riservatezza del traffico inter-rete

Infrastructure Composer non genera traffico con client e applicazioni locali.

## AWS Identity and Access Management per AWS Infrastructure Composer

AWS Identity and Access Management (IAM) è un programma Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori

IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse di Infrastructure Composer. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

## Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso tramite policy](#)
- [Come AWS Infrastructure Composer funziona con IAM](#)

## Destinatari

Infrastructure Composer richiede almeno l'accesso in sola lettura a. Console di gestione AWS. Qualsiasi utente con questa autorizzazione può utilizzare tutte le funzionalità di Infrastructure Composer. L'accesso granulare a funzionalità specifiche di Infrastructure Composer non è supportato.

## Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/Facebook. Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente di IAM.

## Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

## Identità federata

Come best practice, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Che cos'è il Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

## Utenti e gruppi IAM

Un [utente IAM](#) è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente di IAM.

## Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

## Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste

politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

## Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scegliere tra policy gestite e policy in linea](#) nella Guida per l'utente di IAM.

## Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

## Elenchi di controllo degli accessi (ACLs)

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica della lista di controllo degli accessi \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

## Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

## Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

## Come AWS Infrastructure Composer funziona con IAM

AWS Infrastructure Composer richiede almeno l'accesso in sola lettura a. Console di gestione AWS Qualsiasi utente con questa autorizzazione può utilizzare tutte le funzionalità di Infrastructure Composer. L'accesso granulare a funzionalità specifiche di Infrastructure Composer non è supportato.

Quando distribuisce il modello di progetto e i file su AWS CloudFormation, avrai bisogno delle autorizzazioni necessarie. Per ulteriori informazioni, consulta [Controllare l'accesso con AWS Identity and Access Management nella Guida](#) per l'AWS CloudFormation utente.

La tabella seguente mostra con quali funzionalità IAM possono essere utilizzate AWS Infrastructure Composer.

Funzionalità IAM	Supporto per Infrastructure Composer
<a href="#">Policy basate sull'identità</a>	No
<a href="#">Policy basate su risorse</a>	No
<a href="#">Operazioni di policy</a>	No
<a href="#">Risorse relative alle policy</a>	No
<a href="#">Chiavi di condizione delle policy</a>	No
<a href="#">ACLs</a>	No
<a href="#">ABAC (tag nelle policy)</a>	No
<a href="#">Credenziali temporanee</a>	Si
<a href="#">Autorizzazioni del principale</a>	No
☹️ <a href="#">Ruoli di servizio</a>	No
<a href="#">Ruoli collegati al servizio</a>	No

Per avere una visione di alto livello di come Infrastructure Composer e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

## Politiche basate sull'identità per Infrastructure Composer

Supporta politiche basate sull'identità: No

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare

una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

## Politiche basate sulle risorse all'interno di Infrastructure Composer

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#). I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

## Azioni politiche per Infrastructure Composer

Supporta azioni politiche: No

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni di Infrastructure Composer, consulta [Actions Defined by AWS Infrastructure Composer](#) nel Service Authorization Reference.

## Risorse politiche per Infrastructure Composer

Supporta le risorse di policy: No

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (\*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di risorse di Infrastructure Composer e relativi ARNs, consulta [Resources Defined by AWS Infrastructure Composer](#) nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, consulta [Azioni definite da AWS Infrastructure Composer](#).

## Chiavi relative alle condizioni delle politiche per Infrastructure Composer

Supporta le chiavi di condizione delle policy specifiche del servizio: No

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di Infrastructure Composer, consulta [Condition Keys for AWS Infrastructure Composer](#) nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, vedi [Azioni definite da AWS Infrastructure Composer](#).

## ACLs in Infrastructure Composer

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

## ABAC con Infrastructure Composer

Supporta ABAC (tag nelle policy): No

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

## Utilizzo di credenziali temporanee con Infrastructure Composer

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza temporanee in IAM](#) e [Servizi AWS compatibili con IAM](#) nella Guida per l'utente IAM.

È possibile utilizzare credenziali temporanee per accedere a Infrastructure Composer tramite Console di gestione AWS. Per un esempio, consulta [Abilitare l'accesso di un broker di identità personalizzato alla AWS console nella Guida](#) per l'utente IAM.

## Autorizzazioni principali multiservizio per Infrastructure Composer

Supporta l'inoltro delle sessioni di accesso (FAS): no

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

## Ruoli di servizio per Infrastructure Composer

Supporta i ruoli di servizio: no

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.

### Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di Infrastructure Composer. Modifica i ruoli di servizio solo quando Infrastructure Composer fornisce indicazioni in tal senso.

## Ruoli collegati ai servizi per Infrastructure Composer

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

## Convalida della conformità per AWS Infrastructure Composer

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione [Scope by Compliance Program Servizi AWS](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la [Documentazione AWS sulla sicurezza](#).

## Resilienza in AWS Infrastructure Composer

L'infrastruttura AWS globale è costruita attorno a zone Regioni AWS di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità è possibile progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

[Per ulteriori informazioni sulle zone di disponibilità, vedere Global Regioni AWS Infrastructure.AWS](#)

Tutti i dati immessi in Infrastructure Composer vengono utilizzati al solo scopo di fornire funzionalità all'interno di Infrastructure Composer e generare file e directory di progetto che vengono salvati localmente sul computer. Infrastructure Composer non salva né archivia nessuno di questi dati.

# Cronologia dei documenti per Infrastructure Composer

La tabella seguente descrive importanti versioni della documentazione per Infrastructure Composer. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

- Ultimo aggiornamento della documentazione: 30 novembre 2023

Modifica	Descrizione	Data
<a href="#">Contenuti ristrutturati e aggiornati in tutta la guida per gli sviluppatori</a>	La guida è stata riorganizzata e ristrutturata per migliorarne la reperibilità e l'usabilità. Titoli aggiornati e migliorati. Sono stati forniti dettagli aggiuntivi durante l'introduzione di argomenti e concetti.	1° agosto 2024
<a href="#">È stata aggiunta documentazione per l'utilizzo di Infrastructure Composer in modalità CloudFormation console e ha ristrutturato la Infrastructure Composer Developer Guide.</a>	AWS Infrastructure Composer ora può essere utilizzato in CloudFormation modalità console. Per ulteriori informazioni, consulta <a href="#">Utilizzo di Infrastructure Composer in modalità CloudFormation console</a> . Inoltre, gran parte del contenuto della guida per l'utente è stato riorganizzato per creare un'esperienza semplificata.	28 marzo 2024
<a href="#">È stata aggiunta documentazione per l'integrazione di Infrastructure Composer con CodeWhisperer</a>	AWS Infrastructure Composer dal Toolkit for VS Code fornisce un'integrazione con Amazon CodeWhisperer. Per ulteriori informazioni, consulta <a href="#">Usare AWS Infrastru</a>	30 novembre 2023

<a href="#">È stata aggiunta la documentazione per la distribuzione dell'applicazione con Infrastructure Composer dal AWS Toolkit for Visual Studio Code</a>	<a href="#">cture Composer con Amazon CodeWhisperer.</a> Utilizzate il pulsante di sincronizzazione dal canvas di Infrastructure Composer per distribuire l'applicazione su. Cloud AWS Per ulteriori informazioni, consulta <a href="#">Distribuire l'applicazione con sam sync.</a>	30 novembre 2023
<a href="#">È stata aggiunta la documentazione per Infrastructure Composer fornita da AWS Toolkit for Visual Studio Code</a>	È ora possibile utilizzare Infrastructure Composer di VS Code con. AWS Toolkit for Visual Studio Code Per ulteriori informazioni, consulta <a href="#">Using AWS Infrastructure Composer from the AWS Toolkit for Visual Studio Code.</a>	30 novembre 2023
<a href="#">Aggiunta l'integrazione di Step Functions Workflow Studio</a>	Avvia Step Functions Workflow Studio dal canvas di Infrastructure Composer. Per ulteriori informazioni, consulta <a href="#">Usare AWS Infrastructure Composer con AWS Step Functions.</a>	27 novembre 2023
<a href="#">Aggiunta la console Lambda e l'integrazione di Infrastructure Composer</a>	Avvia il canvas Infrastructure Composer dalla console Lambda. Per ulteriori informazioni, consulta <a href="#">Utilizzo AWS Infrastructure Composer con la AWS Lambda console.</a>	14 novembre 2023

[Aggiunto Amazon VPC come servizio in primo piano con Infrastructure Composer](#)

Infrastructure Composer introduce un tag VPC per visualizzare le risorse configurate con un VPC. Puoi anche configurare le funzioni Lambda VPCs definendole su un modello esterno. Per ulteriori informazioni, consulta [Using Infrastructure Composer with Amazon VPC](#).

17 ottobre 2023

[Aggiunto Amazon RDS come servizio in primo piano con Infrastructure Composer](#)

Connetti la tua applicazione Infrastructure Composer a un cluster o istanza Amazon RDS DB definito su un modello esterno. Per ulteriori informazioni, consulta [Using Infrastructure Composer with Amazon RDS](#).

17 ottobre 2023

[Aggiunto il supporto di Infrastructure Composer per la progettazione con tutte le risorse CloudFormation](#)

Seleziona CloudFormation una risorsa dalla palette Risorse con cui progettare e le tue applicazioni. Per ulteriori informazioni, consulta [Lavorare con qualsiasi CloudFormation risorsa](#).

26 settembre 2023

[È stata aggiunta la documentazione per le schede in Infrastructure Composer](#)

Infrastructure Composer supporta diversi tipi di schede che è possibile utilizzare per progettare e creare l'applicazione. Per ulteriori informazioni, consulta [Progettazione con schede in Infrastructure Composer](#).

20 settembre 2023

<a href="#">È stata aggiunta la documentazione per le funzionalità di annullamento e ripristino</a>	Utilizzate i pulsanti Annulla e Ripristina nella tela di Infrastruttura Composer. Per ulteriori informazioni, consulta <a href="#">Annulla e ripristina</a> .	1° agosto 2023
<a href="#">È stata aggiunta documentazione per la modalità di sincronizzazione locale</a>	Utilizza la modalità di sincronizzazione locale per sincronizzare e salvare automaticamente il progetto sul computer locale. Per saperne di più, consulta <a href="#">Modalità di sincronizzazione locale</a> .	1° agosto 2023
<a href="#">Documentazione aggiunta per la funzionalità di esportazione della tela</a>	Usa la funzione di esportazione della tela per esportare la tela dell'applicazione come immagine sul tuo computer locale. Per ulteriori informazioni, <a href="#">consulta Esportazione della tela</a> .	1° agosto 2023
<a href="#">Supporto di Infrastructure Composer per riferimenti a file esterni</a>	Fai riferimento ai file esterni per le risorse supportate in Infrastructure Composer. Per ulteriori informazioni, consulta <a href="#">Lavorare con modelli che fanno riferimento a file esterni</a> .	17 maggio 2023
<a href="#">Nuova documentazione sulla connessione delle risorse</a>	Connect le risorse tra loro per definire relazioni basate sugli eventi tra le risorse dell'applicazione. Per ulteriori informazioni, consulta <a href="#">Connettere le risorse utilizzando l'area di disegno visiva di Infrastructure Composer</a> .	7 marzo 2023

<a href="#">Nuova funzionalità Change Inspector</a>	Usa Change Inspector per visualizzare gli aggiornamenti del codice del modello e scoprire cosa Infrastructure Composer sta creando per te. Per ulteriori informazioni, consulta <a href="#">Visualizza gli aggiornamenti del codice con Change Inspector</a> .	7 marzo 2023
<a href="#">Infrastructure Composer è ora disponibile a livello generale</a>	AWS Infrastructure Composer è ora disponibile a livello generale. Per ulteriori informazioni, vedi <a href="#">AWS Infrastructure Composer Ora disponibile al pubblico: crea rapidamente applicazioni serverless visivamente</a> .	7 marzo 2023
<a href="#">Ampliamento dei vantaggi dell'utilizzo della modalità connessa</a>	Usa Infrastructure Composer in modalità connessa con il tuo IDE locale per accelerare e lo sviluppo. Per ulteriori informazioni, consulta <a href="#">Utilizzo di Infrastructure Composer con l'IDE locale</a> .	7 marzo 2023
<a href="#">Argomento aggiornato sull'utilizzo di altri AWS servizi per distribuire l'applicazione</a>	Usa Infrastructure Composer per progettare applicazioni serverless pronte per l'implementazione. Usalo per distribuire la tua applicazione serverless AWS SAM . Per ulteriori informazioni, consulta <a href="#">Utilizzo di Infrastructure Composer con e. CloudFormation AWS SAM</a>	3 marzo 2023

[Aggiunta la sezione dedicata ai concetti serverless](#)

Scopri i concetti di base del serverless prima di utilizzare Infrastructure Composer. Per ulteriori informazioni, consulta Concetti [serverless](#).

2 marzo 2023

[Rilascio pubblico](#)

Versione pubblica iniziale di Infrastructure Composer.

1 dicembre 2022

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.