



Guide du développeur

AWS Infrastructure Composer



AWS Infrastructure Composer: Guide du développeur

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

Les marques commerciales et la présentation commerciale d'Amazon ne peuvent pas être utilisées en relation avec un produit ou un service extérieur à Amazon, d'une manière susceptible d'entraîner une confusion chez les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce qu'Infrastructure Composer ?	1
Composez votre architecture	2
Définissez vos modèles	4
Intégrez à vos flux de travail	5
Moyens d'accéder à Infrastructure Composer	6
En savoir plus	8
Étapes suivantes	8
Concepts sans serveur	8
Concepts sans serveur	9
Cartes	10
Cartes de composants améliorées	11
Exemple	12
Cartes à composants standard	13
Connexions par carte	17
Connexions entre les cartes	17
Connexions entre des cartes de composants améliorées	18
Connexions vers et depuis des cartes de ressources laC standard	20
Premiers pas	21
Visite guidée de la console	21
Étapes suivantes	22
Charger et modifier	22
Étape 1 : Ouvrez la démo	23
Étape 2 : Explorez le canevas visuel	23
Étape 3 : étendez votre architecture	27
Étape 4 : Enregistrez votre candidature	28
Étapes suivantes	29
Génération	29
Propriétés de ressource	30
Étape 1 : Créez votre projet	30
Ajouter des cartes	33
Étape 3 : Configurez votre REST API	34
Étape 4 : Configurez vos fonctions	35
Étape 5 : Connectez vos cartes	36
Étape 6 : Organiser le canevas	37

Ajouter une table DynamoDB	38
Étape 8 : passez en revue votre modèle	39
Étape 9 : Intégrez à vos flux de travail	40
Étapes suivantes	40
Où utiliser Infrastructure Composer	41
Console Infrastructure Composer	41
Vue d'ensemble visuelle	42
Gérez votre projet	45
Connectez-vous à votre réseau local IDE	49
Autoriser l'accès aux pages Web	52
Synchronisation et enregistrement locaux	53
Importer depuis la console Lambda	57
Toile d'exportation	57
CloudFormation mode console	59
Pourquoi utiliser ce mode ?	59
Accédez à ce mode	60
Visualisez un déploiement	60
Création d'un nouveau modèle	61
Mettre à jour une pile existante	62
AWS Toolkit for Visual Studio Code	64
Vue d'ensemble visuelle	65
Accès depuis VS Code	66
Synchroniser avec AWS Cloud	67
Compositeur d'infrastructure avec Amazon Q	69
Comment composer	72
Placez les cartes sur la toile	72
Regroupez les cartes	73
Regroupement de cartes de composants améliorées	73
Regroupement d'une carte de composant standard dans une autre	74
Cartes Connect	76
Connexion de cartes de composants améliorées	76
Connexion de cartes standard	77
Exemples	79
Déconnecter les cartes	81
Cartes de composants améliorées	81
Cartes à composants standard	82

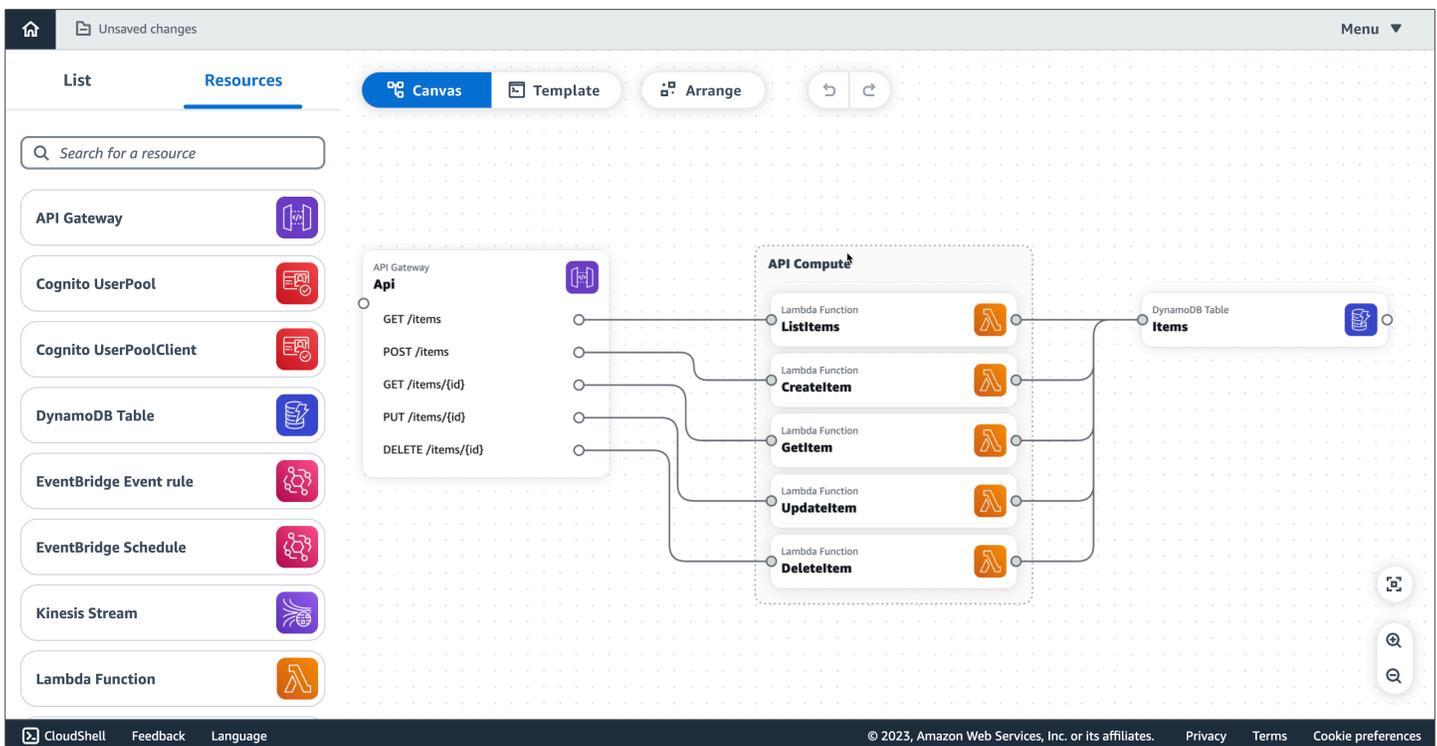
Organiser les cartes	83
Configuration et modification des cartes	84
Cartes améliorées	85
Cartes standard	101
Supprimer des cartes	102
Cartes de composants améliorées	103
Cartes à composants standard	103
Afficher les mises à jour du code	103
Avantages du Change Inspector	104
Procédure	105
En savoir plus	106
Fichiers externes de référence	107
Bonnes pratiques	108
Création d'une référence de fichier externe	108
Charger un projet	109
Créer une application à l'aide du AWS SAM CLI	110
Référencer un OpenAPI Spécification de	114
Intégrer à Amazon VPC	117
Identifier les ressources et les informations	118
Configuration des fonctions	124
Paramètres dans les modèles importés	124
Ajouter de nouveaux paramètres aux modèles importés	127
Configurer une fonction Lambda avec un VPC dans un autre modèle	128
Déployez dans le AWS cloud	132
AWS SAM Concepts importants	132
Étapes suivantes	132
Configurez le AWS SAM CLI	133
Installez le AWS CLI	133
Installez le AWS SAM CLI	133
Accédez au AWS SAM CLI	133
Étapes suivantes	134
Créer et déployer	134
Supprimer une pile	142
Résolution des problèmes	144
Messages d'erreur	144
« Impossible d'ouvrir ce dossier »	144

« Modèle incompatible »	144
« Le dossier fourni contient un modèle .yaml existant »	145
« Votre navigateur n'est pas autorisé à enregistrer votre projet dans ce dossier... »	146
Sécurité	147
Protection des données	147
Chiffrement des données	149
Chiffrement en transit	149
Gestion des clés	149
Confidentialité du trafic inter-réseaux	149
AWS Identity and Access Management	149
Public ciblé	150
Authentification par des identités	150
Gestion des accès à l'aide de politiques	154
Comment AWS Infrastructure Composer fonctionne avec IAM	157
Validation de conformité	164
Résilience	165
Historique de la documentation	167
.....	clxxiv

Qu'est-ce que c'est AWS Infrastructure Composer ?

AWS Infrastructure Composer vous permet de composer visuellement des applications modernes sur AWS. Plus précisément, vous pouvez utiliser Infrastructure Composer pour visualiser, créer et déployer des applications modernes à partir de tous les AWS services pris en charge par, AWS CloudFormation sans avoir besoin d'être un expert en la matière AWS CloudFormation.

Au fur et à mesure que vous composez votre AWS CloudFormation infrastructure, Infrastructure Composer crée des modèles d'infrastructure sous forme de code (IaC) via une drag-and-drop interface agréable, tout en suivant les AWS meilleures pratiques. L'image suivante montre à quel point il est facile de glisser-déposer, de configurer et de connecter des ressources sur le canevas visuel d'Infrastructure Composer.



Infrastructure Composer peut être utilisé depuis la console Infrastructure Composer, le AWS Toolkit for Visual Studio Code, et en mode CloudFormation console.

Rubriques

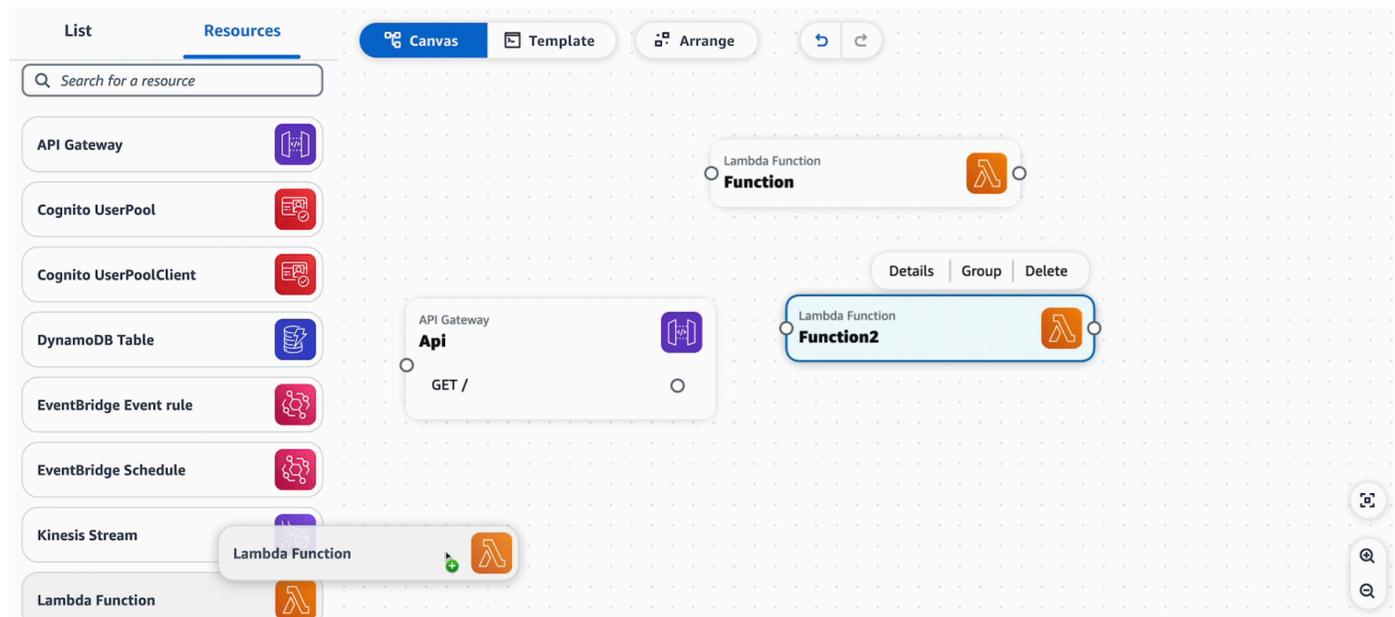
- [Composez l'architecture de votre application](#)
- [Définissez votre infrastructure sous forme de modèles de code \(IaC\)](#)
- [Intégrez vos flux de travail existants](#)

- [Moyens d'accéder à Infrastructure Composer](#)
- [En savoir plus](#)
- [Étapes suivantes](#)
- [Concepts sans serveur pour AWS Infrastructure Composer](#)

Composez l'architecture de votre application

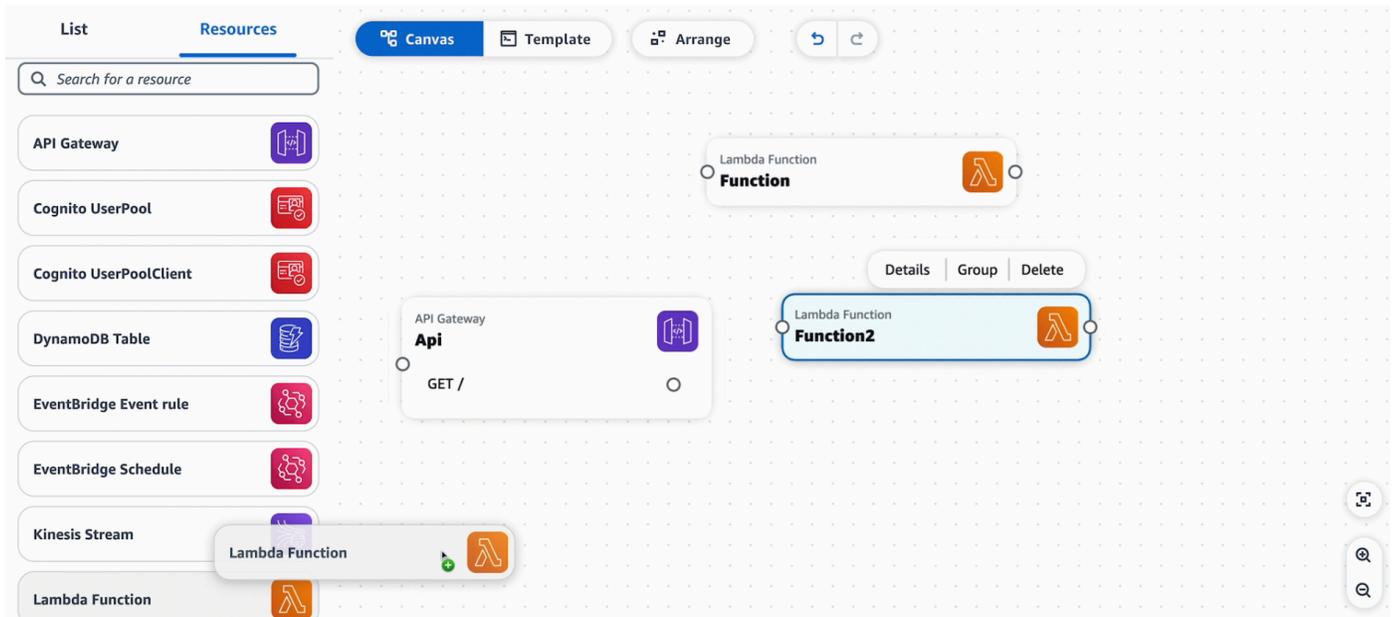
Construisez avec des cartes

Placez des cartes sur le canevas Infrastructure Composer pour visualiser et créer l'architecture de votre application.



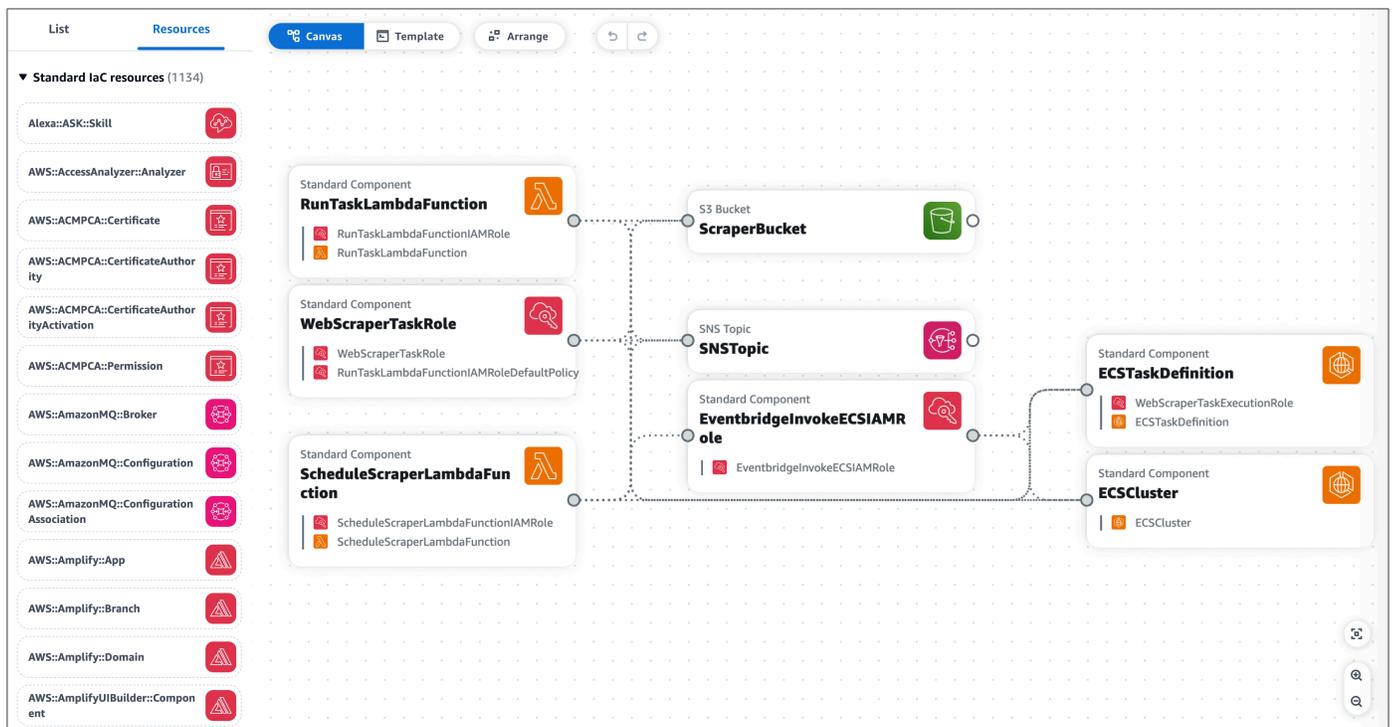
Connect les cartes entre elles

Configurez la manière dont vos ressources interagissent les unes avec les autres en les connectant visuellement entre elles. Spécifiez leurs propriétés plus en détail via un panneau de propriétés organisé.



Travaillez avec n'importe quelle AWS CloudFormation ressource

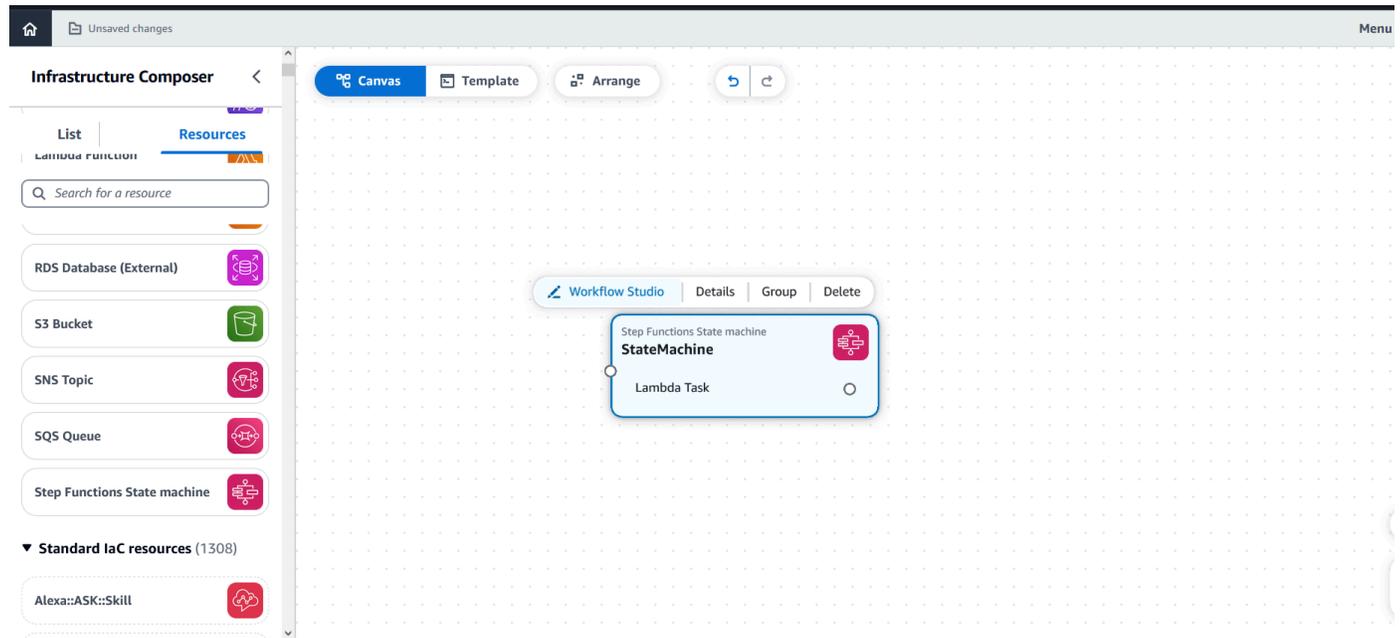
Faites glisser n'importe quelle AWS CloudFormation ressource sur le canevas pour composer l'architecture de votre application. Infrastructure Composer fournit un modèle IaC de départ que vous pouvez utiliser pour spécifier les propriétés de votre ressource. Pour en savoir plus, consultez [Configurer et modifier des cartes dans Infrastructure Composer](#).



Accédez à des fonctionnalités supplémentaires grâce aux fonctionnalités Services AWS

Fonctionnalités d' Services AWS Infrastructure Composer couramment utilisées ou configurées conjointement lors de la création d'applications. Pour en savoir plus, consultez [Intégrer à Amazon VPC](#).

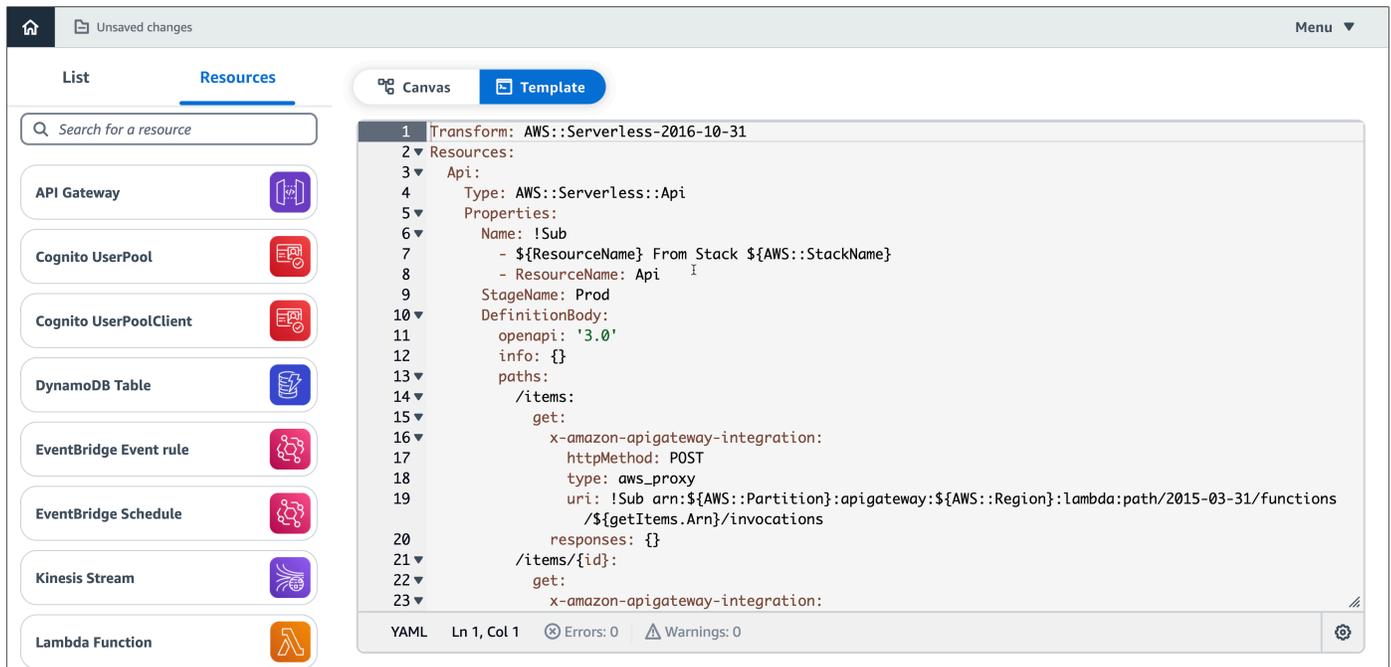
Voici un exemple de la AWS Step Functions fonctionnalité, qui fournit une intégration pour lancer Step Functions Workflow Studio directement dans le canevas Infrastructure Composer.



Définissez votre infrastructure sous forme de modèles de code (IaC)

Infrastructure Composer crée votre code d'infrastructure

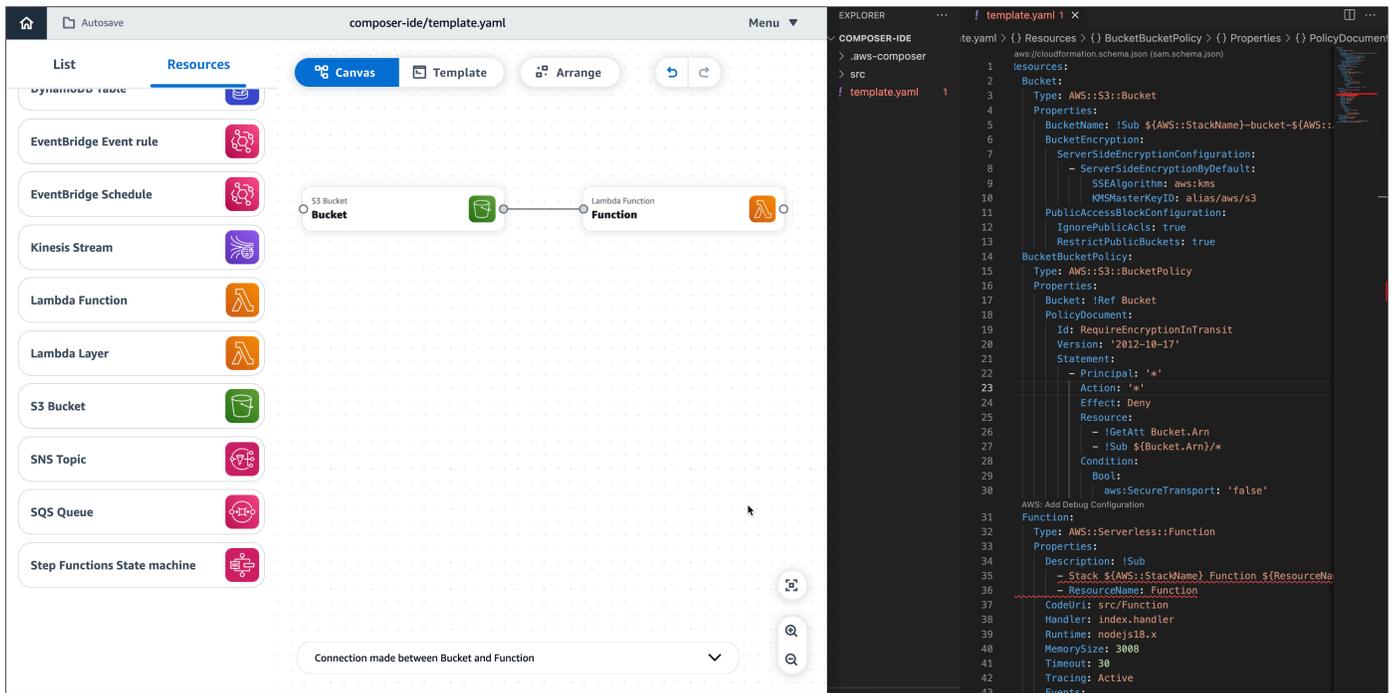
Au fur et à mesure que vous composez, Infrastructure Composer crée automatiquement vos modèles AWS CloudFormation and AWS Serverless Application Model (AWS SAM), conformément aux AWS meilleures pratiques. Vous pouvez consulter et modifier vos modèles directement depuis Infrastructure Composer. Infrastructure Composer synchronise automatiquement les modifications entre le canevas visuel et le code de votre modèle.



Intégrez vos flux de travail existants

Importer des modèles et des projets existants

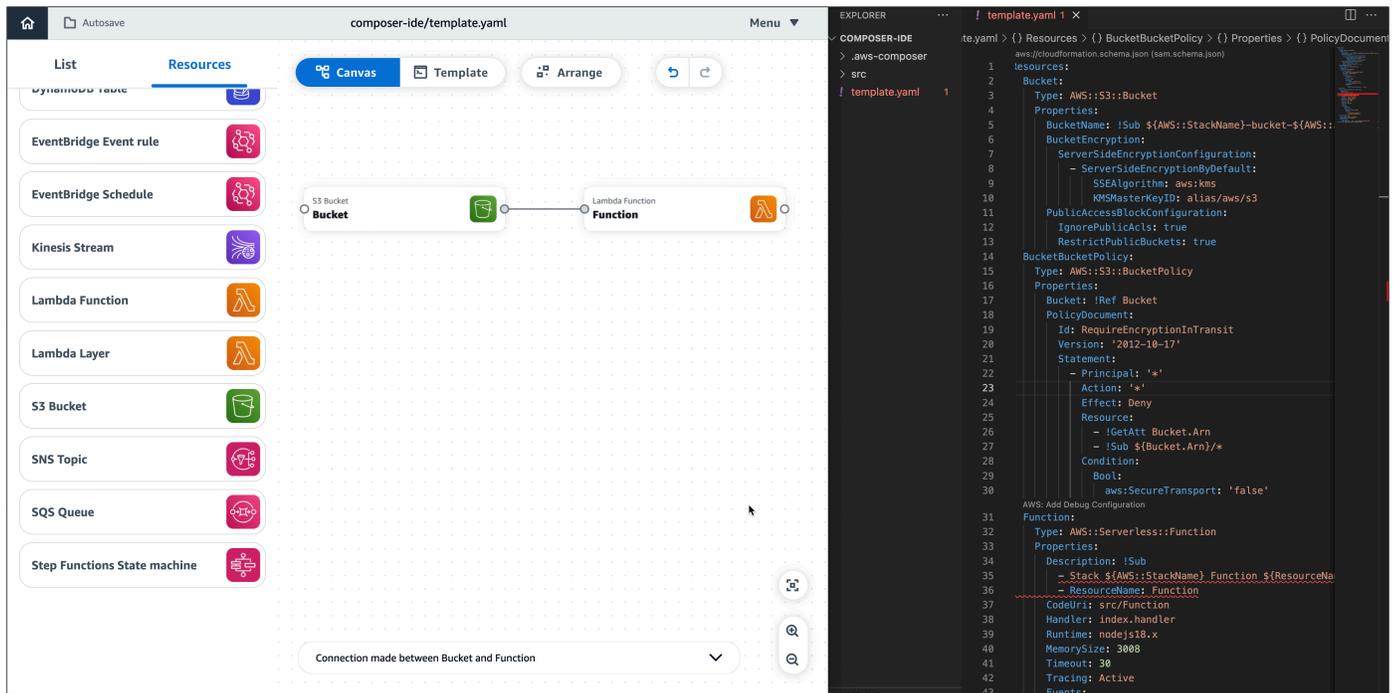
Importez des AWS SAM modèles AWS CloudFormation et des modèles existants pour les visualiser afin de mieux comprendre et de modifier leur conception. Exportez les modèles que vous créez dans Infrastructure Composer et intégrez-les dans vos flux de travail existants en vue du déploiement.



Moyens d'accéder à Infrastructure Composer

Depuis la console Infrastructure Composer

Accédez à Infrastructure Composer via la console Infrastructure Composer pour démarrer rapidement. En outre, vous pouvez utiliser le mode de synchronisation locale pour synchroniser et enregistrer automatiquement Infrastructure Composer avec votre machine locale.



Depuis la AWS CloudFormation console

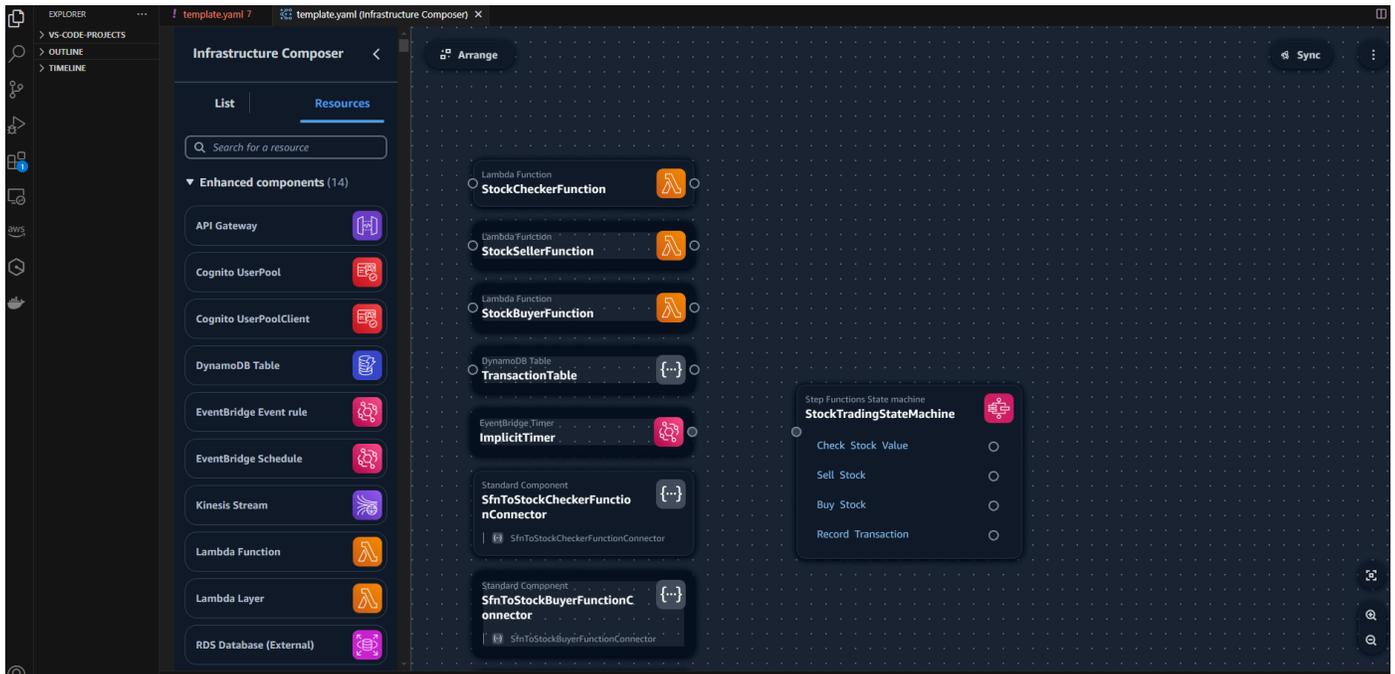
La console Infrastructure Composer prend également en charge le [mode CloudFormation console](#), une amélioration par rapport à CloudFormation Designer qui est intégré au flux de travail de AWS CloudFormation stack. Ce nouvel outil est désormais l'outil recommandé pour visualiser vos CloudFormation modèles.

Depuis la console Lambda

Avec Infrastructure Composer, vous pouvez également importer des fonctions Lambda depuis la console Lambda. Pour en savoir plus, consultez [Importer des fonctions dans Infrastructure Composer depuis la console Lambda](#).

À partir du AWS Toolkit for Visual Studio Code

Accédez à Infrastructure Composer via l'extension Toolkit for VS Code pour intégrer Infrastructure Composer dans votre environnement de développement local.



En savoir plus

Pour en savoir plus sur Infrastructure Composer, consultez les ressources suivantes :

- [Cartes Infrastructure Composer](#)
- [Composez et créez visuellement des applications sans serveur | Serverless Office Hours](#) — Présentation et démonstration d'Infrastructure Composer.

Étapes suivantes

Pour configurer Infrastructure Composer, voir [Commencer à utiliser la console Infrastructure Composer](#).

Concepts sans serveur pour AWS Infrastructure Composer

Découvrez les concepts de base du mode sans serveur avant de l'utiliser AWS Infrastructure Composer.

Concepts sans serveur

Architecture basée sur les événements

Une application sans serveur comprend des AWS services individuels, tels que AWS Lambda pour le calcul et Amazon DynamoDB pour la gestion des bases de données, qui jouent chacun un rôle spécialisé. Ces services sont ensuite librement intégrés les uns aux autres par le biais d'une architecture basée sur les événements. Pour en savoir plus sur l'architecture basée sur les événements, consultez [Qu'est-ce qu'une architecture basée sur les événements ?](#).

Infrastructure en tant que code (IaC)

L'infrastructure en tant que code (IaC) permet de traiter l'infrastructure de la même manière que les développeurs traitent le code, en appliquant la même rigueur que le développement du code d'application à l'approvisionnement de l'infrastructure. Vous définissez votre infrastructure dans un fichier modèle, vous la déployez et vous AWS créez les ressources pour vous. AWS AvecIaC, vous définissez dans le code ce que vous AWS souhaitez provisionner. Pour plus d'informations, consultez la section [L'infrastructure en tant que code](#) dans le AWS AWS livre blanc Introduction à DevOps on.

Technologies sans serveur

Grâce aux technologies AWS sans serveur, vous pouvez créer et exécuter des applications sans avoir à gérer vos propres serveurs. L'ensemble de la gestion des serveurs est assuré par AWS ce qui offre de nombreux avantages tels que le dimensionnement automatique et la haute disponibilité intégrée, ce qui vous permet de mettre rapidement votre idée en production. Grâce aux technologies sans serveur, vous pouvez vous concentrer sur l'essentiel de votre produit sans avoir à vous soucier de la gestion et de l'exploitation des serveurs. Pour en savoir plus sur le mode sans serveur, consultez [Serverless activé](#). AWS

Pour une introduction de base aux principaux services AWS sans serveur, voir [Serverless 101 : Understanding the serverless services at Serverless Land](#).

Cartes Infrastructure Composer

Infrastructure Composer simplifie le processus d'écriture de l'infrastructure sous forme de code (IaC) pour les AWS CloudFormation ressources. Pour utiliser efficacement Infrastructure Composer, vous devez d'abord comprendre deux concepts de base : les [cartes](#) Infrastructure Composer et [les connexions par carte](#).

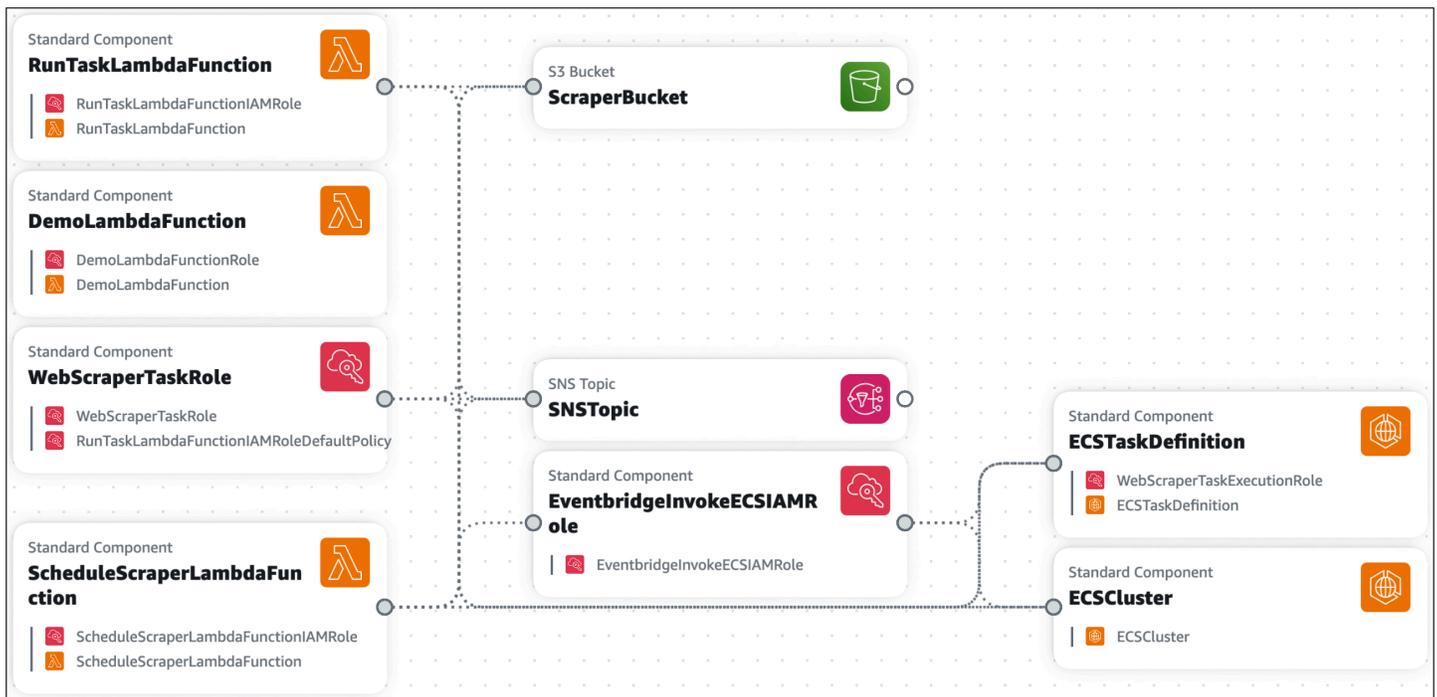
Dans Infrastructure Composer, les cartes représentent AWS CloudFormation des ressources. Il existe deux catégories générales de cartes :

- [Carte de composants améliorée](#) : collection de AWS CloudFormation ressources qui ont été combinées dans une seule carte organisée qui améliore la facilité d'utilisation et les fonctionnalités, et qui est conçue pour une grande variété de cas d'utilisation. Les cartes de composants améliorées sont les premières cartes répertoriées dans la palette Ressources d'Infrastructure Composer.
- [Carte de ressources IaC standard](#) — Une AWS CloudFormation ressource unique. Chaque carte de ressources IaC standard, une fois glissée sur le canevas, est étiquetée Composant standard et peut être combinée en plusieurs ressources.

Note

En fonction de la carte, une carte de ressources IaC standard peut être qualifiée de carte de composant standard après avoir été glissée sur le canevas visuel. Cela signifie simplement que la carte est un ensemble d'une ou plusieurs cartes de ressources IaC standard.

Bien que certains types de cartes soient disponibles dans la palette Ressources, des cartes peuvent également apparaître sur le canevas lorsque vous importez un modèle existant AWS CloudFormation ou AWS Serverless Application Model (AWS SAM) dans Infrastructure Composer. L'image suivante est un exemple d'application importée contenant différents types de cartes :



Rubriques

- [Cartes de composants améliorées dans Infrastructure Composer](#)
- [Cartes de composants standard dans Infrastructure Composer](#)
- [Connexions par carte dans Infrastructure Composer](#)

Cartes de composants améliorées dans Infrastructure Composer

Les cartes de composants améliorées sont créées et gérées par Infrastructure Composer. Chaque carte contient AWS CloudFormation des ressources couramment utilisées conjointement lors de la création d'applications AWS. Leur code d'infrastructure est créé par Infrastructure Composer conformément aux AWS meilleures pratiques. Les cartes de composants améliorées constituent un excellent moyen de commencer à concevoir votre application.

Les fiches de composants améliorées sont disponibles dans la palette Ressources, dans la section Composants améliorés.

Les cartes de composants améliorées peuvent être entièrement configurées et utilisées dans Infrastructure Composer pour concevoir et créer vos applications sans serveur. Nous vous recommandons d'utiliser des cartes de composants améliorées lors de la conception de vos applications sans code existant.

Ce tableau présente nos composants améliorés avec des liens vers la spécification du modèle AWS CloudFormation or AWS Serverless Application Model (AWS SAM) de la ressource vedette de la carte :

Carte	Référence
API Passerelle Amazon	AWS: :Sans serveur : : API
Amazon Cognito UserPool	AWS: :Cognito : : UserPool
Amazon Cognito UserPoolClient	AWS: :Cognito : : UserPoolClient
Tableau Amazon DynamoDB	AWS: :DynamoDB : : Tableau
Règle Amazon EventBridge Event	AWS: :Events : : Rule
EventBridge Horaire	AWS: :Scheduler : : Schedule
Amazon Kinesis Stream	AWS: :Kinesis : : Stream
AWS Lambda Fonction	AWS: :Serverless : : Function
Couche Lambda	AWS: :Sans serveur : : LayerVersion
Compartiment Amazon Simple Storage Service (Amazon S3)	AWS: :S3 : : Bucket
Rubrique Amazon Simple Notification Service (Amazon SNS)	AWS: : SNS : : Sujet
File d'attente Amazon Simple Queue Service (Amazon SQS)	AWS: : SQS : : File d'attente
AWS Step Functions Machine d'État	AWS: :Sans serveur : : StateMachine

Exemple

Voici un exemple de composant amélioré du compartiment S3 :



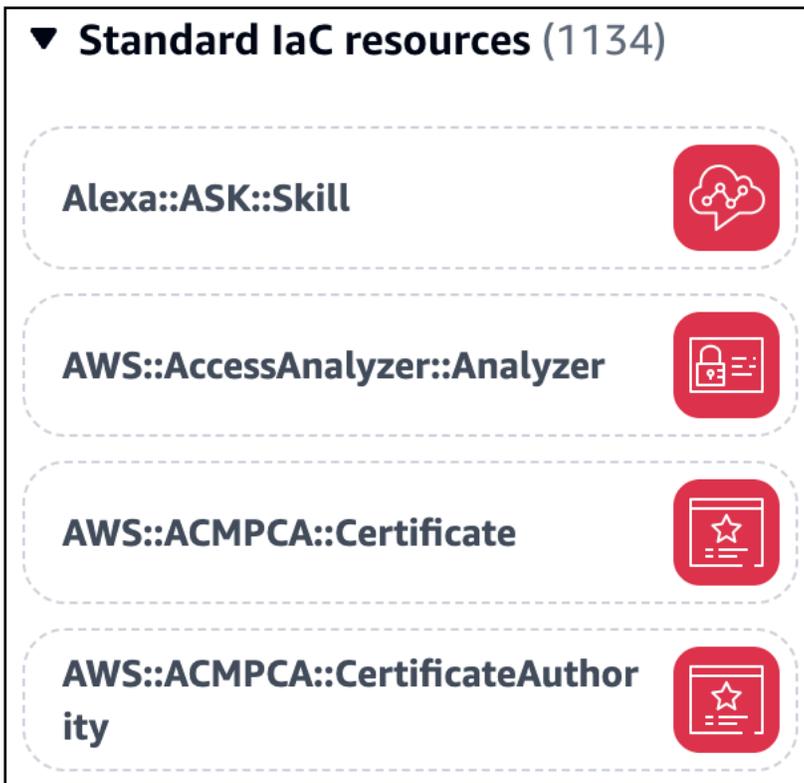
Lorsque vous faites glisser une carte de composant du compartiment S3 sur le canevas et que vous visualisez votre modèle, les deux AWS CloudFormation ressources suivantes sont ajoutées à votre modèle :

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

La carte de composants améliorée du compartiment S3 représente deux AWS CloudFormation ressources qui sont toutes deux requises pour qu'un compartiment Amazon Simple Storage Service (Amazon S3) interagisse avec les autres services de votre application.

Cartes de composants standard dans Infrastructure Composer

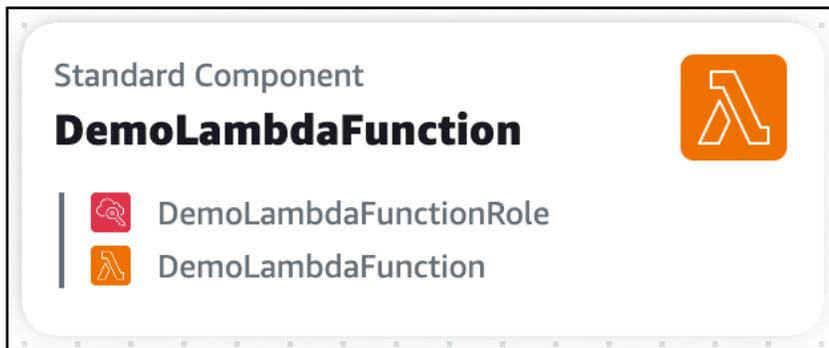
Avant qu'une carte de composant standard ne soit placée sur le canevas visuel d'Infrastructure Composer, elle est répertoriée en tant que carte de ressources standard (iAc) dans la palette Ressources d'Infrastructure Composer. Une carte de ressources standard (laC) représente une AWS CloudFormation ressource unique. Chaque carte de ressources laC standard, une fois placée sur le canevas visuel, devient une carte intitulée Composant standard et peut être combinée pour représenter plusieurs AWS CloudFormation ressources.



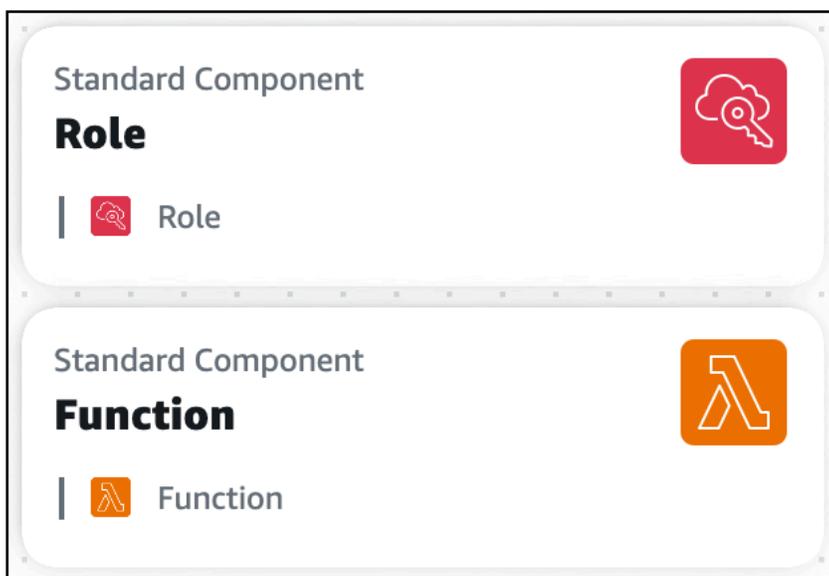
Chaque carte de ressources IaC standard peut être identifiée par son type de AWS CloudFormation ressource. Voici un exemple de carte de ressources IaC standard qui représente un type de `AWS::ECS::Cluster` AWS CloudFormation ressource :



Chaque carte de composant standard permet de visualiser les AWS CloudFormation ressources qu'elle contient. Voici un exemple de carte de composants standard qui inclut deux ressources IaC standard :



Lorsque vous configurez les propriétés de vos cartes de composants standard, Infrastructure Composer peut combiner des cartes associées. Par exemple, voici deux cartes de composants standard :



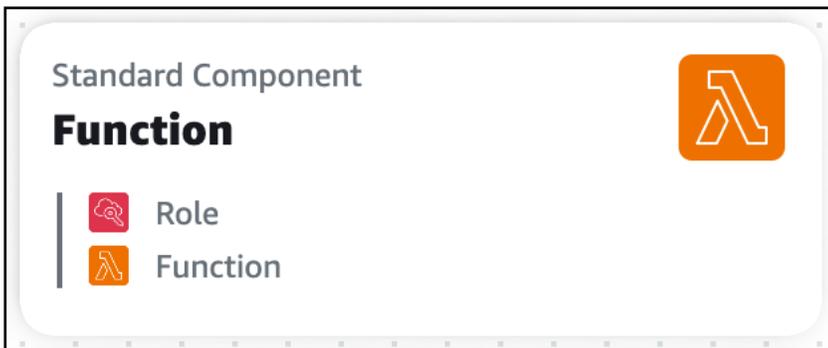
Dans le panneau des propriétés des ressources de la carte de composant standard représentant une `AWS::Lambda::Function` ressource, nous référençons le rôle AWS Identity and Access Management (IAM) par son identifiant logique :

The image shows the AWS Infrastructure Composer interface. On the left is a component palette with a grid background. It contains two standard components: a 'Role' component (red icon) and a 'Function' component (orange icon). The 'Function' component is highlighted with a blue border. On the right is a 'Resource properties' panel for an '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 'Resource properties' panel is a 'Resource reference' button with an external link icon.

Après avoir enregistré notre modèle, les deux cartes de composants standard sont combinées en une seule carte de composant standard.



Connexions par carte dans Infrastructure Composer

Dans AWS Infrastructure Composer, une connexion entre deux cartes est affichée visuellement par une ligne. Ces lignes représentent les relations basées sur les événements au sein de votre application.

Rubriques

- [Connexions entre les cartes](#)
- [Connexions entre des cartes de composants améliorées](#)
- [Connexions vers et depuis des cartes de ressources laC standard](#)

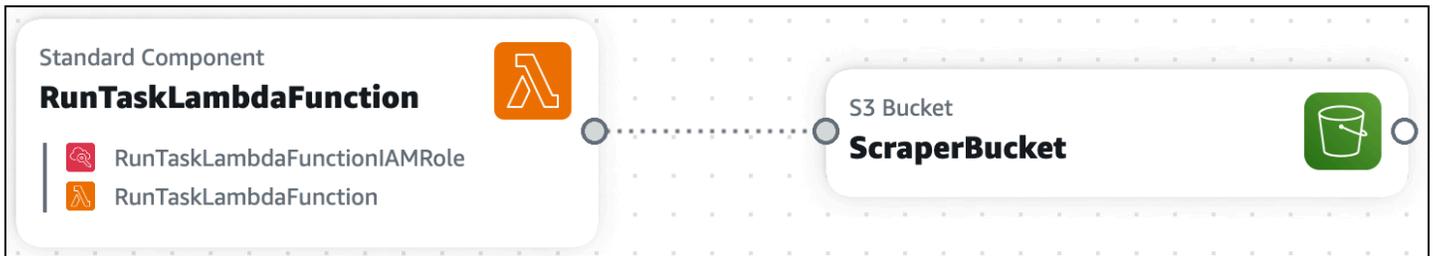
Connexions entre les cartes

La façon dont vous connectez les cartes entre elles varie en fonction du type de carte. Chaque carte améliorée possède au moins un port de connexion. Pour les connecter, il vous suffit de sélectionner un port de connecteur et de le faire glisser vers le port d'une autre carte, et Infrastructure Composer connectera les deux ressources ou affichera un message indiquant que cette configuration n'est pas prise en charge.



Comme indiqué ci-dessus, les lignes entre les cartes de composants améliorées sont continues. À l'inverse, les cartes de ressources laC standard (également appelées cartes à composants standard)

ne disposent pas de ports de connexion. Pour ces cartes, vous devez spécifier ces relations basées sur les événements dans le modèle de votre application, et Infrastructure Composer détectera automatiquement leurs connexions et les visualisera avec une ligne en pointillés entre vos cartes.

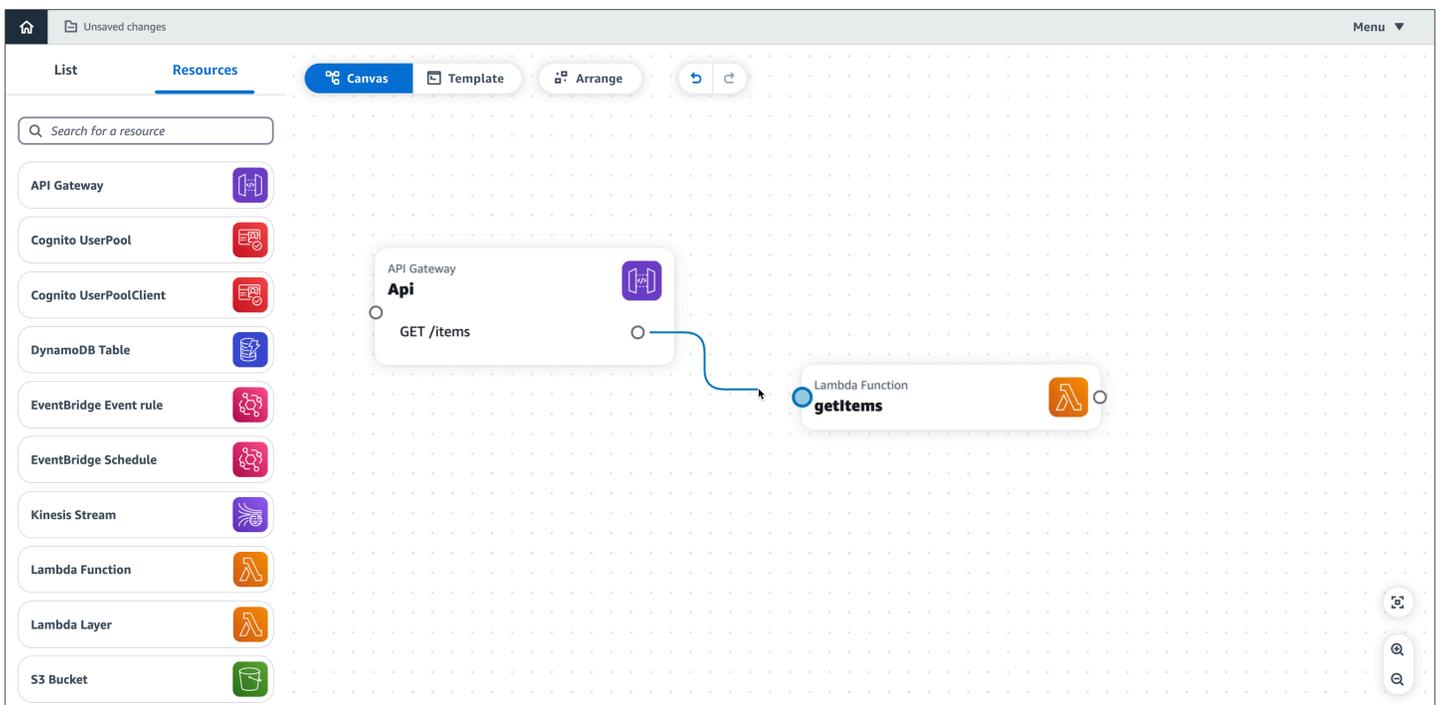


Pour en savoir plus, consultez les sections ci-dessous.

Connexions entre des cartes de composants améliorées

Dans Infrastructure Composer, une connexion entre deux cartes de composants améliorées est affichée visuellement par une ligne continue. Ces lignes représentent les relations basées sur les événements au sein de votre application.

Pour connecter deux cartes, cliquez sur un port d'une carte et faites-le glisser sur le port d'une autre carte.



Note

Les cartes de ressources IaC standard ne disposent pas de ports de connexion. Pour ces cartes, vous devez spécifier leurs relations basées sur les événements dans le modèle de votre application, et Infrastructure Composer détectera automatiquement leurs connexions et les visualisera avec une ligne en pointillés entre vos cartes.

Pour de plus amples informations, veuillez consulter [Connectez les cartes sur le canevas visuel d'Infrastructure Composer](#).

Quelle offre améliorée de cartes de composants ?

Les connexions entre deux cartes, indiquées visuellement par une ligne, fournissent les éléments suivants si nécessaire :

- AWS Identity and Access Management (IAM) politiques
- Variables d'environnement
- Événements

Stratégies IAM

Lorsqu'une ressource a besoin d'une autorisation pour appeler une autre ressource, Infrastructure Composer fournit des politiques basées sur les ressources à l'aide de modèles de politiques AWS Serverless Application Model (AWS SAM).

- Pour en savoir plus sur IAM les autorisations et les politiques, voir [Présentation de la gestion des accès : autorisations et politiques](#) dans le guide de IAM l'utilisateur.
- Pour en savoir plus sur les AWS SAM modèles de règles, consultez les [modèles de AWS SAM politiques](#) dans le Guide du AWS Serverless Application Model développeur.

Variables d'environnement

Les variables d'environnement sont des valeurs temporaires qui peuvent être modifiées pour affecter le comportement de vos ressources. Si nécessaire, Infrastructure Composer définit le code d'infrastructure pour utiliser les variables d'environnement entre les ressources.

Événements

Les ressources peuvent invoquer une autre ressource par le biais de différents types d'événements. Le cas échéant, Infrastructure Composer définit le code d'infrastructure nécessaire pour que les ressources interagissent par le biais de types d'événements.

Connexions vers et depuis des cartes de ressources IaC standard

Toutes les AWS CloudFormation ressources peuvent être utilisées sous forme de cartes de ressources IaC standard dans la palette Ressources. Lorsque vous faites glisser une carte de ressource iAC standard sur le canevas, une carte de ressources iAC standard devient une carte de composant standard, ce qui invite Infrastructure Composer à créer un modèle de départ pour votre ressource dans votre application.

Pour de plus amples informations, veuillez consulter [Cartes standard dans Infrastructure Composer](#).

Commencer à utiliser la console Infrastructure Composer

Utilisez les rubriques de cette section pour configurer AWS Infrastructure Composer et apprendre à concevoir une application à l'aide de son canevas visuel. La visite guidée et les didacticiels de cette section sont présentés dans la console Infrastructure Composer, qui constitue l'expérience utilisateur par défaut. Les rubriques de cette section vous montrent comment remplir les conditions préalables à l'utilisation d'Infrastructure Composer, utiliser la console Infrastructure Composer, charger et modifier un projet, et créer votre première application.

Infrastructure Composer est également disponible depuis AWS Toolkit for Visual Studio Code et en mode CloudFormation console. Les expériences entre les outils sont généralement les mêmes, mais il existe des différences entre les deux. Pour plus de détails sur l'utilisation d'Infrastructure Composer dans chacun de ces outils, consultez [Où vous pouvez utiliser Infrastructure Composer](#).

Rubriques

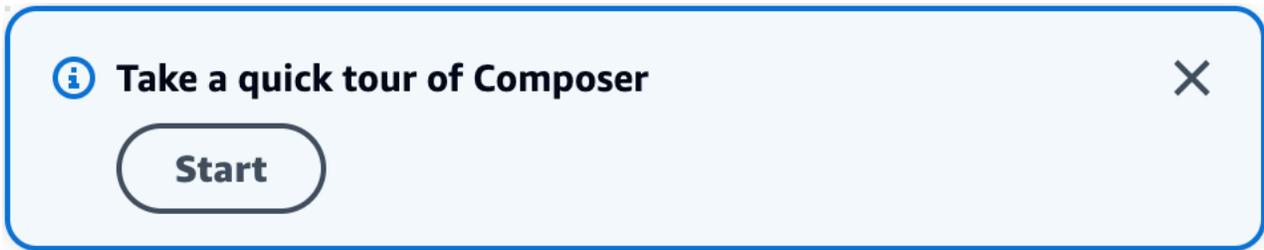
- [Visite guidée de la console Infrastructure Composer](#)
- [Chargez et modifiez le projet de démonstration d'Infrastructure Composer](#)
- [Créez votre première application avec Infrastructure Composer](#)

Visite guidée de la console Infrastructure Composer

Pour avoir une idée générale du AWS Infrastructure Composer fonctionnement, suivez la visite guidée intégrée à la console Infrastructure Composer. Pour une présentation de la console Infrastructure Composer, consultez [Visite guidée de la console Infrastructure Composer](#). Pour obtenir des conseils détaillés sur l'utilisation d'Infrastructure Composer, reportez-vous à [Comment composer AWS Infrastructure Composer](#).

Pour visiter Infrastructure Composer

1. Connectez-vous à la [console Infrastructure Composer](#).
2. Sur la page d'accueil, choisissez Ouvrir une démo.
3. Dans le coin supérieur droit, dans la fenêtre Visite rapide de Composer, sélectionnez Démarrer.



4. Dans la fenêtre de visite de Composer, procédez comme suit :

- Pour passer à l'étape suivante, choisissez Next.
- Pour revenir à l'étape précédente, choisissez Previous.
- À la dernière étape, pour terminer la visite, choisissez Fin.

La visite fournit un bref aperçu des fonctionnalités de base d'Infrastructure Composer, telles que l'utilisation, la configuration et la connexion de cartes. Pour plus d'informations, consultez [Comment composer AWS Infrastructure Composer](#).

Étapes suivantes

Pour charger et modifier un projet dans Infrastructure Composer, consultez [Chargez et modifiez le projet de démonstration d'Infrastructure Composer](#).

Chargez et modifiez le projet de démonstration d'Infrastructure Composer

Utilisez ce didacticiel pour vous familiariser avec l'interface utilisateur d'Infrastructure Composer et apprendre à charger, modifier et enregistrer le projet de démonstration d'Infrastructure Composer.

Ce didacticiel est réalisé dans la console Infrastructure Composer. Une fois terminé, vous serez prêt à commencer [Créez votre première application avec Infrastructure Composer](#).

Rubriques

- [Étape 1 : Ouvrez la démo](#)
- [Étape 2 : Explorez le canevas visuel d'Infrastructure Composer](#)
- [Étape 3 : étendez l'architecture de votre application](#)
- [Étape 4 : Enregistrez votre candidature](#)
- [Étapes suivantes](#)

Étape 1 : Ouvrez la démo

Commencez à utiliser Infrastructure Composer en créant un projet de démonstration.

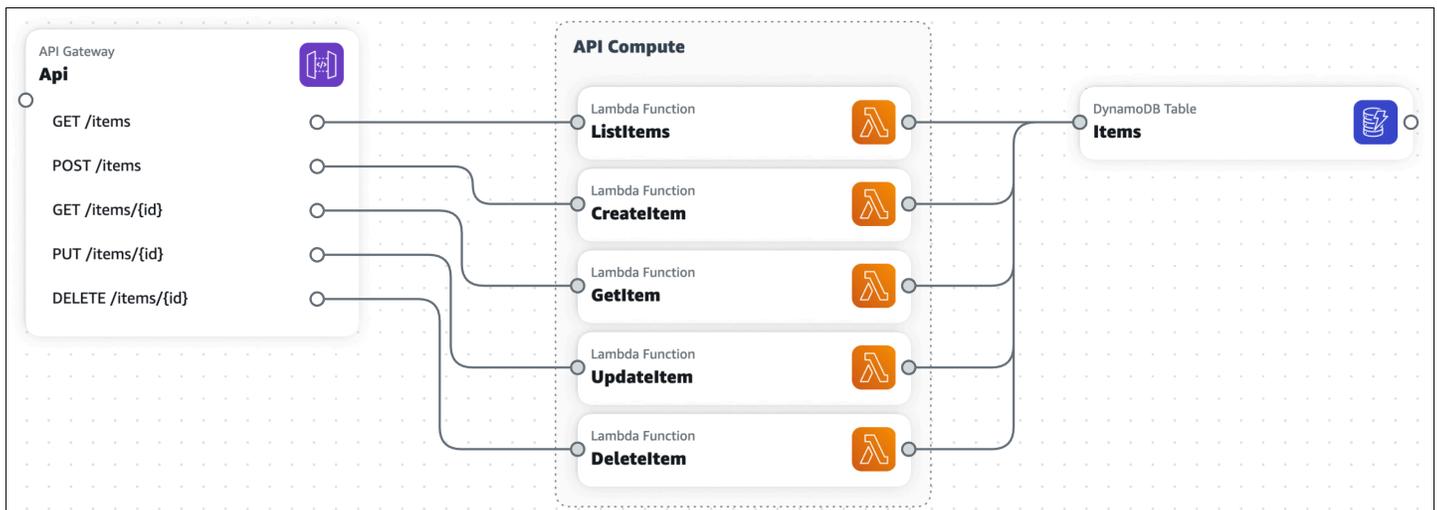
Pour créer un projet de démonstration

1. Connectez-vous à la [console Infrastructure Composer](#).
2. Sur la page d'accueil, choisissez Ouvrir une démo.

L'application de démonstration est une application sans serveur de base de création, de lecture, de suppression et de mise à jour (CRUD) qui inclut :

- Une ressource Amazon API Gateway avec cinq itinéraires.
- Cinq AWS Lambda fonctions.
- Une table Amazon DynamoDB.

L'image suivante représente la démo :

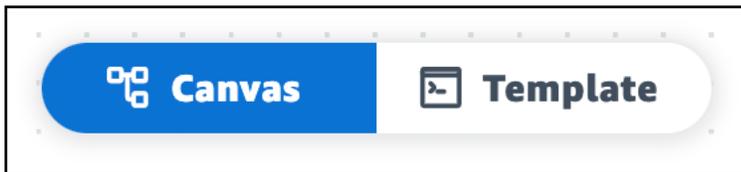


Étape 2 : Explorez le canevas visuel d'Infrastructure Composer

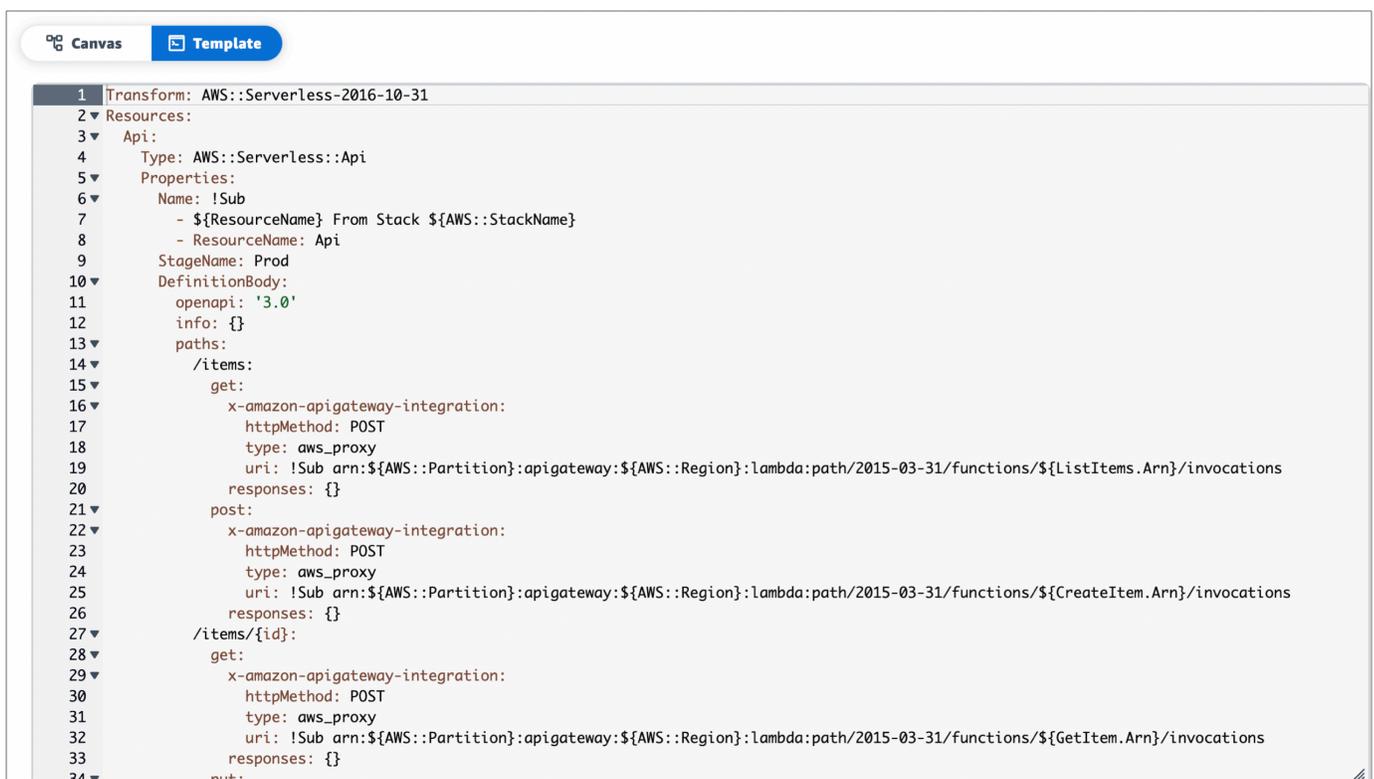
Découvrez les fonctionnalités du canevas visuel pour créer votre projet de démonstration d'Infrastructure Composer. Pour un aperçu de la disposition visuelle du canevas, voir [Vue d'ensemble visuelle](#).

Pour explorer les fonctionnalités du canevas visuel

1. Lorsque vous ouvrez un projet d'application nouveau ou existant, Infrastructure Composer charge la vue du canevas, comme indiqué au-dessus de la zone d'affichage principale.

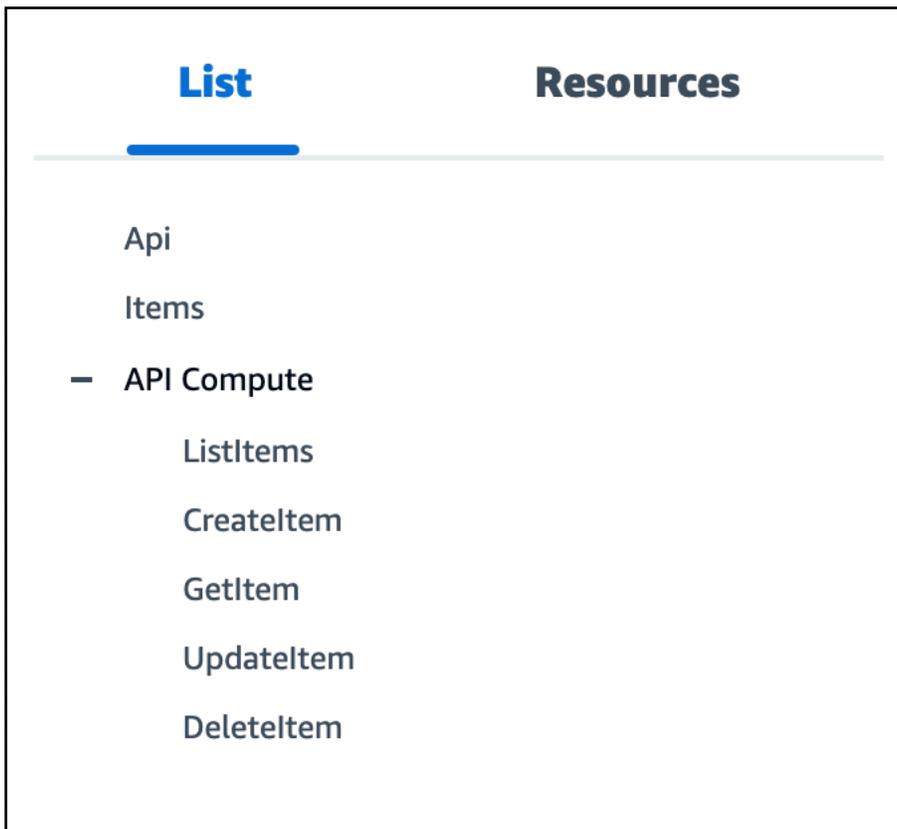


Pour afficher le code d'infrastructure de votre application dans la zone d'affichage principale, choisissez Modèle. Par exemple, voici la vue du modèle AWS Serverless Application Model (AWS SAM) du projet de démonstration d'Infrastructure Composer.

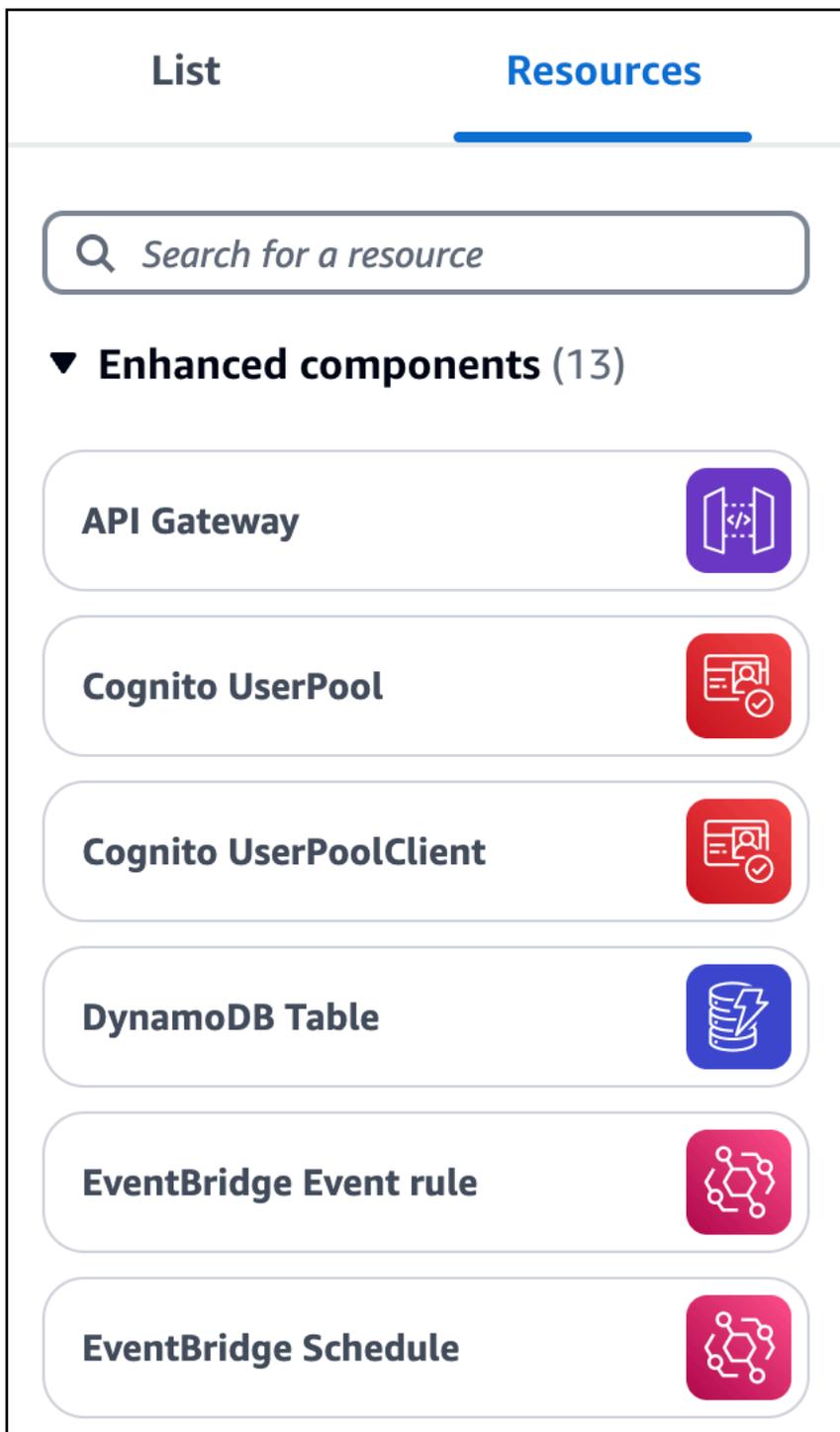


```
1 Transform: AWS::Serverless-2016-10-31
2 Resources:
3   Api:
4     Type: AWS::Serverless::Api
5     Properties:
6       Name: !Sub
7         - ${ResourceName} From Stack ${AWS::StackName}
8         - ResourceName: Api
9       StageName: Prod
10    DefinitionBody:
11      openapi: '3.0'
12      info: {}
13      paths:
14        /items:
15          get:
16            x-amazon-apigateway-integration:
17              httpMethod: POST
18              type: aws_proxy
19              uri: !Sub arn:${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. Pour afficher à nouveau la vue du canevas de votre application, choisissez Canvas.
3. Pour afficher les ressources de votre application organisées sous forme d'arborescence, choisissez Liste.



4. Pour afficher la palette de ressources, sélectionnez Ressources. Cette palette contient des cartes que vous pouvez utiliser pour étendre l'architecture de votre application. Vous pouvez rechercher des cartes ou parcourir la liste.



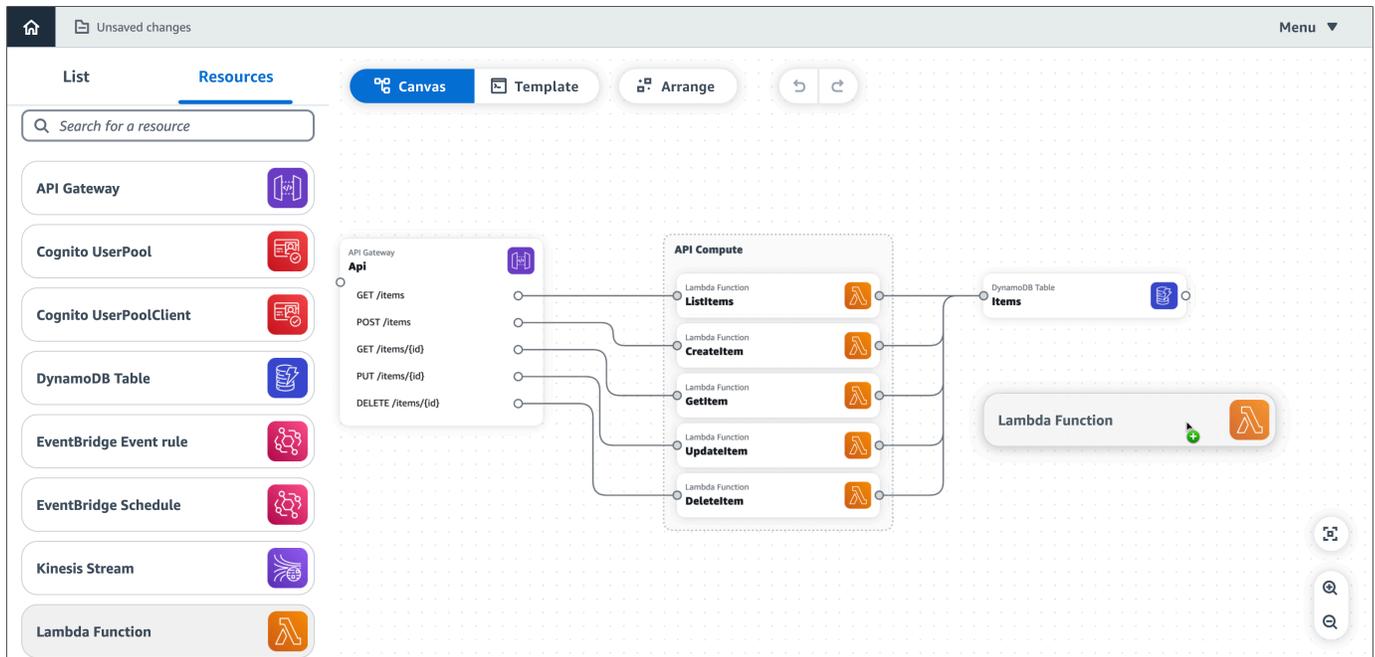
5. Pour vous déplacer dans le canevas visuel, utilisez des gestes de base. Pour de plus amples informations, veuillez consulter [Placez les cartes sur la toile](#).

Étape 3 : étendez l'architecture de votre application

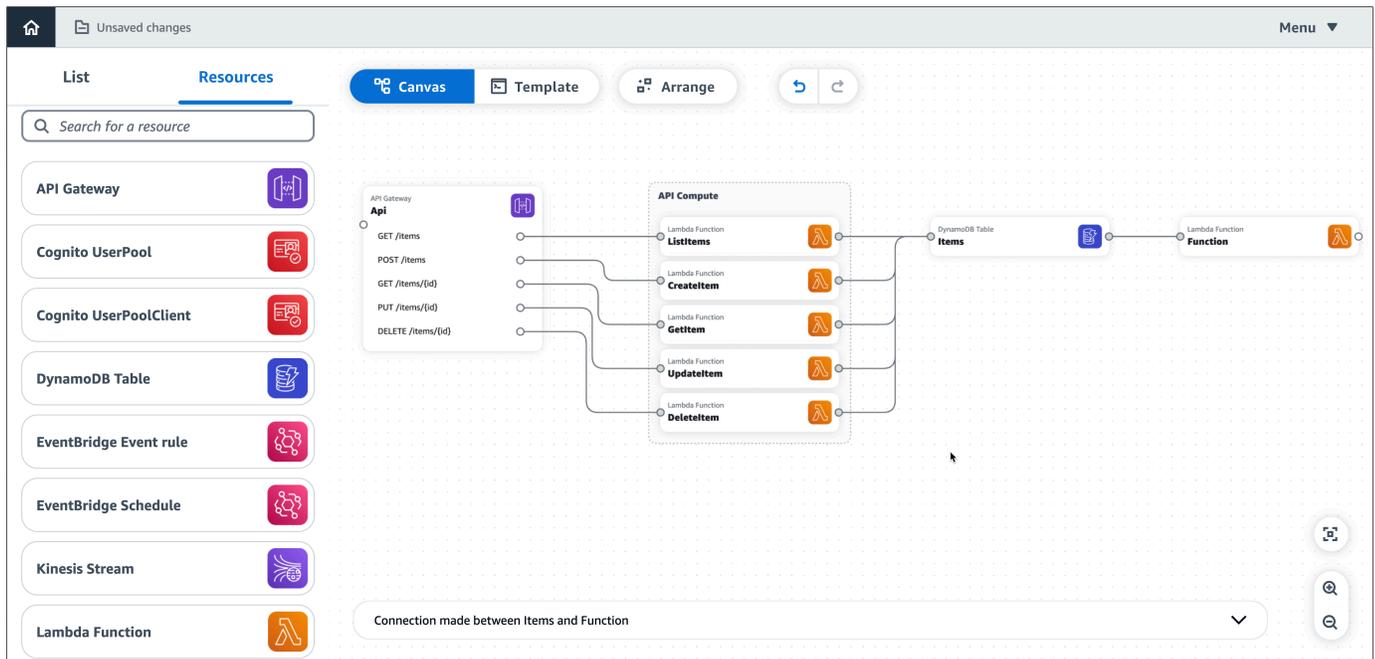
Au cours de cette étape, vous allez étendre l'architecture de votre application en ajoutant une fonction Lambda à votre table DynamoDB.

Pour ajouter une fonction Lambda à votre table DynamoDB

1. Depuis la palette de ressources (Ressources), faites glisser la carte de composant améliorée de la fonction Lambda sur le canevas, à droite de la fiche Tableau DynamoDB.



2. Connectez la table DynamoDB à la fonction Lambda. Pour les connecter, cliquez sur le port droit de la carte DynamoDB Table et faites-le glisser sur le port gauche de la carte de fonction Lambda.
3. Choisissez Disposer pour organiser les cartes dans l'affichage en mode canevas.



4. Configurez votre fonction Lambda. Pour le configurer, effectuez l'une des opérations suivantes :
- Dans la vue du canevas, modifiez les propriétés de la fonction dans le panneau des propriétés des ressources. Pour ouvrir le panneau, double-cliquez sur la carte de fonction Lambda. Vous pouvez également sélectionner la carte, puis cliquez sur Détails. Pour plus d'informations sur les propriétés configurables des fonctions Lambda répertoriées dans le panneau des propriétés des ressources, consultez le manuel du [AWS Lambda développeur](#).
 - Dans la vue du modèle, modifiez le code de votre fonction (`AWS::Serverless::Function`). Infrastructure Composer synchronise automatiquement vos modifications sur le canevas. Pour plus d'informations sur la ressource fonctionnelle d'un AWS SAM modèle, voir [::Serverless AWS : :Function](#) dans la référence AWS SAM des ressources et des propriétés.

Étape 4 : Enregistrez votre candidature

Enregistrez votre application en enregistrant manuellement votre modèle d'application sur votre machine locale ou en activant la synchronisation locale.

Pour enregistrer manuellement votre modèle de candidature

1. Dans le menu, sélectionnez Enregistrer > Enregistrer le fichier modèle.
2. Donnez un nom à votre modèle et choisissez un emplacement sur votre ordinateur local pour enregistrer votre modèle. Appuyez sur Enregistrer.

Pour obtenir des instructions sur l'activation de la synchronisation locale, consultez [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#).

Étapes suivantes

Pour commencer à créer votre première application, consultez [Créez votre première application avec Infrastructure Composer](#).

Créez votre première application avec Infrastructure Composer

Dans ce didacticiel, vous allez AWS Infrastructure Composer créer une application sans serveur create, read, update and delete (CRUD) qui gère les utilisateurs d'une base de données.

Pour ce didacticiel, nous utilisons Infrastructure Composer dans le AWS Management Console. Nous vous recommandons d'utiliser Google Chrome or Microsoft Edge, et une fenêtre de navigateur en plein écran.

 Êtes-vous novice dans le domaine du sans serveur ?

Nous vous recommandons d'avoir des connaissances de base sur les points suivants :

- [Architecture basée sur les événements](#)
- [Infrastructure en tant que code \(IaC\)](#)
- [Technologies sans serveur](#)

Pour en savoir plus, consultez [Concepts sans serveur pour AWS Infrastructure Composer](#).

Rubriques

- [Référence des propriétés des ressources](#)
- [Étape 1 : Créez votre projet](#)
- [Étape 2 : ajouter des cartes au canevas](#)
- [Étape 3 : Configuration de votre API passerelle REST API](#)
- [Étape 4 : Configuration de vos fonctions Lambda](#)
- [Étape 5 : Connectez vos cartes](#)

- [Étape 6 : Organiser le canevas](#)
- [Étape 7 : Ajouter et connecter une table DynamoDB](#)
- [Étape 8 : passez en revue votre AWS CloudFormation modèle](#)
- [Étape 9 : Intégrez à vos flux de travail de développement](#)
- [Étapes suivantes](#)

Référence des propriétés des ressources

Lors de la création de votre application, utilisez ce tableau comme référence pour configurer les propriétés de votre Amazon API Gateway et de vos AWS Lambda ressources.

Méthode	Chemin	Nom de la fonction
GET	/objets	getItems
GET	/articles/ {id}	getItem
PUT	/articles/ {id}	updateItem
POST	/article	addItem
DELETE	/articles/ {id}	deleteItem

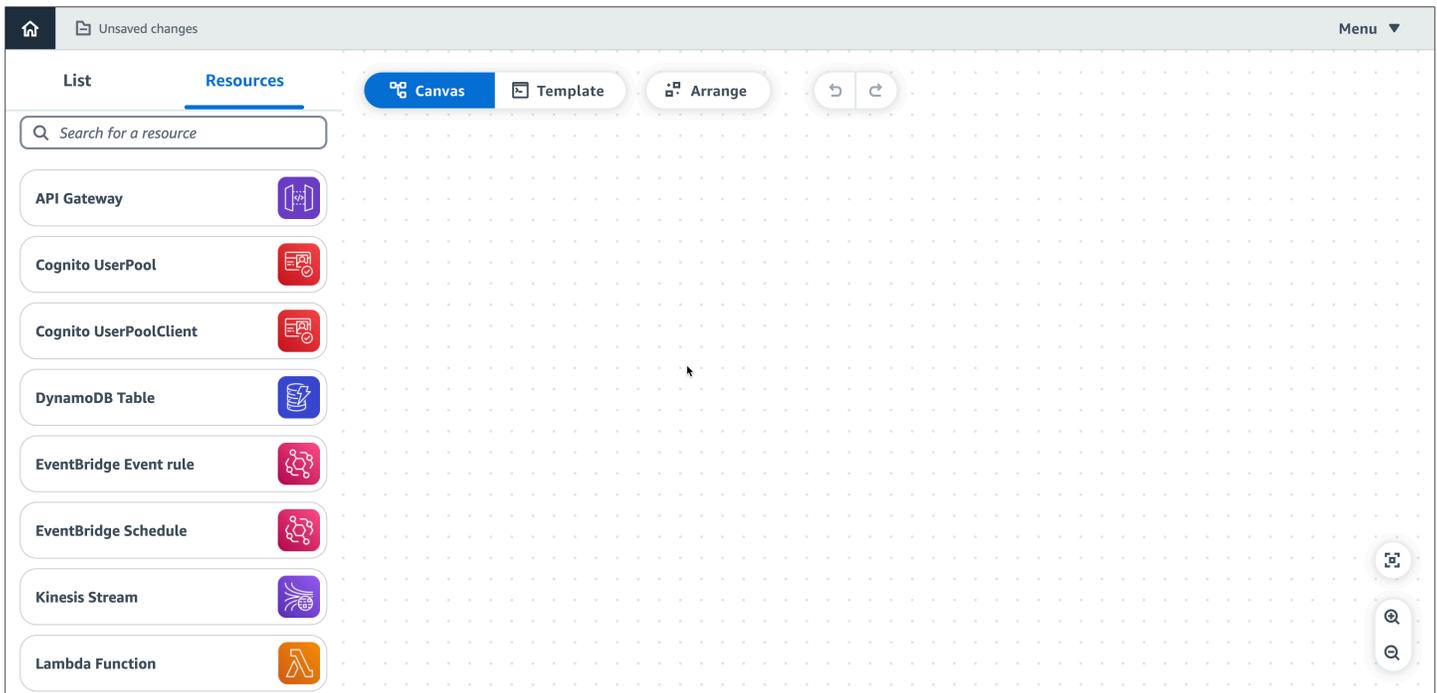
Étape 1 : Créez votre projet

Pour démarrer avec votre application CRUD sans serveur, créez un nouveau projet dans Infrastructure Composer et activez la synchronisation locale.

Pour créer un nouveau projet vierge

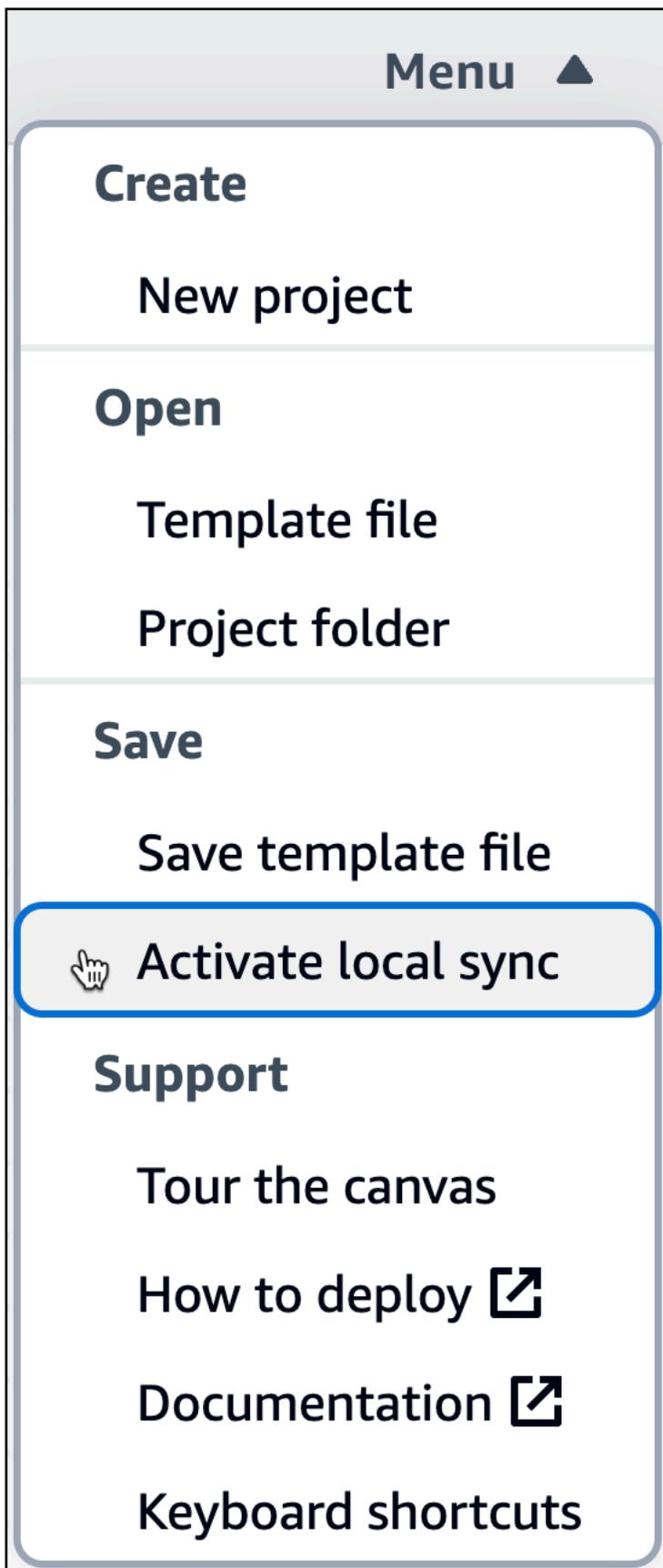
1. Connectez-vous à la [console Infrastructure Composer](#).
2. Sur la page d'accueil, choisissez Créer un projet.

Comme le montre l'image suivante, Infrastructure Composer ouvre le canevas visuel et charge un modèle d'application de départ (vide).



Pour activer la synchronisation locale

1. Dans le menu Infrastructure Composer, sélectionnez Enregistrer > Activer la synchronisation locale.



2. Pour l'emplacement du projet, appuyez sur Sélectionner un dossier et choisissez un répertoire. C'est ici qu'Infrastructure Composer enregistrera et synchronisera vos fichiers modèles et dossiers au fur et à mesure de votre conception.

L'emplacement du projet ne doit pas contenir de modèle de candidature existant.

 Note

La synchronisation locale nécessite un navigateur qui prend en charge l'accès au système de fichiersAPI. Pour de plus amples informations, veuillez consulter [Data Infrastructure Composer a accès à](#).

3. Lorsque vous êtes invité à autoriser l'accès, sélectionnez Afficher les fichiers.
4. Appuyez sur Activer pour activer la synchronisation locale. Lorsque vous êtes invité à enregistrer les modifications, sélectionnez Enregistrer les modifications.

Lorsqu'il est activé, l'indicateur de sauvegarde automatique s'affiche dans le coin supérieur gauche de votre canevas.

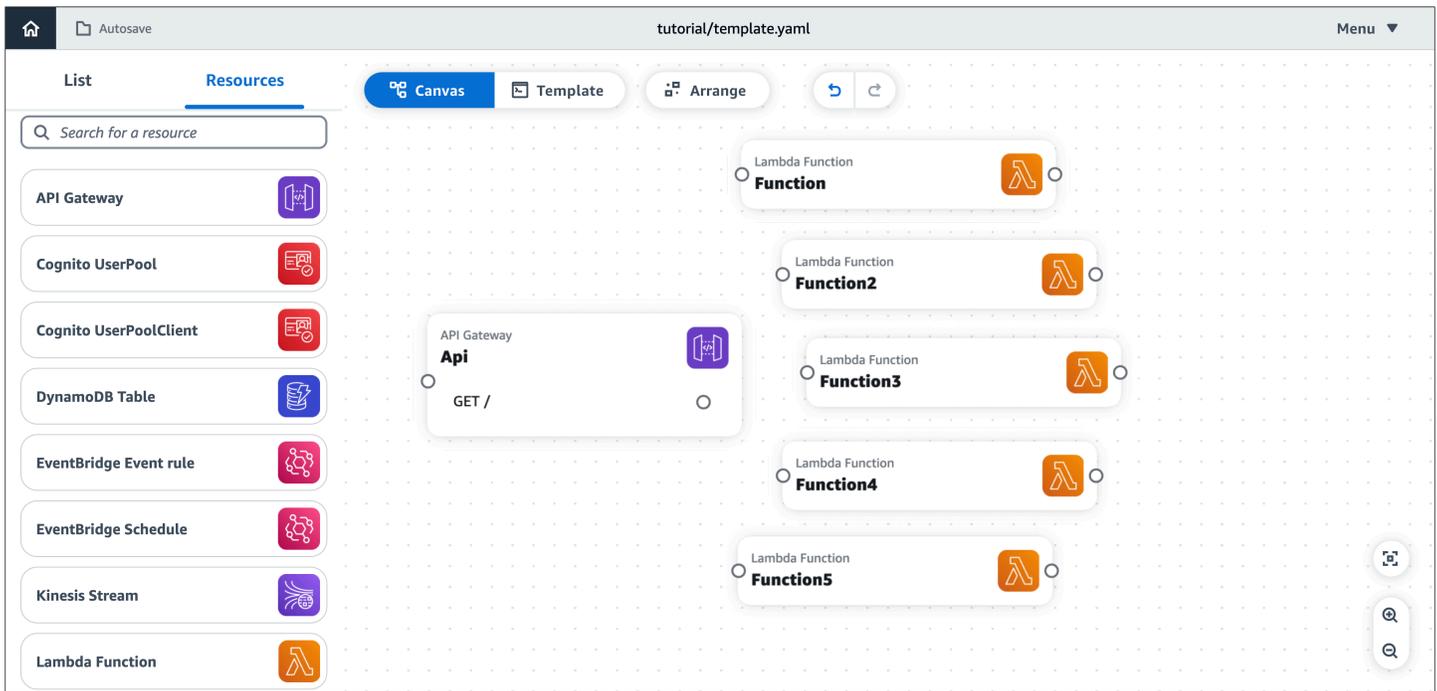
Étape 2 : ajouter des cartes au canevas

Commencez à concevoir l'architecture de votre application à l'aide de cartes de composants améliorées, en commençant par une API passerelle REST API et cinq fonctions Lambda.

Pour ajouter des cartes API Gateway et Lambda au canevas

Dans la palette Ressources, sous la section Composants améliorés, procédez comme suit :

1. Faites glisser une carte APIGateway sur le canevas.
2. Faites glisser une carte de fonction Lambda sur le canevas. Répétez l'opération jusqu'à ce que vous ayez ajouté cinq cartes de fonctions Lambda au canevas.



Étape 3 : Configuration de votre API passerelle REST API

Ajoutez ensuite cinq itinéraires dans votre carte API Gateway.

Pour ajouter des itinéraires à la carte API Gateway

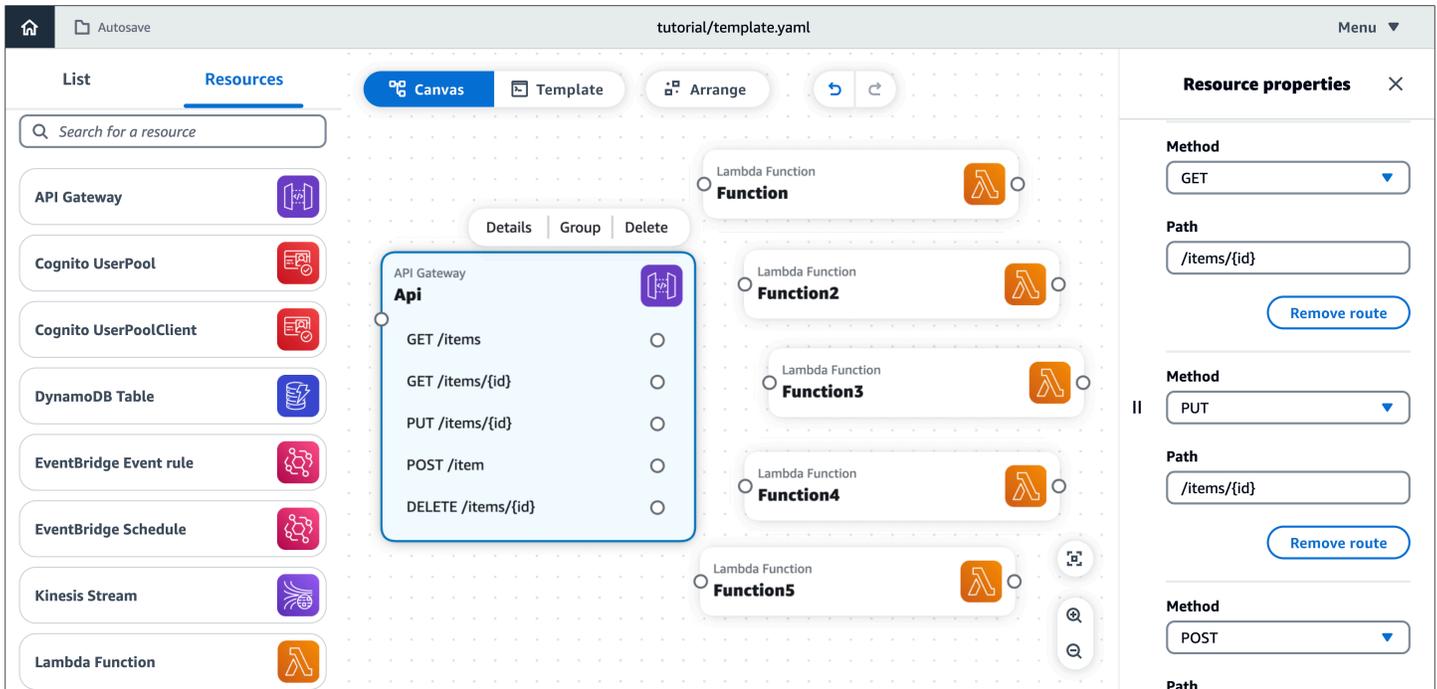
1. Ouvrez le panneau des propriétés des ressources de la carte APIGateway. Pour ouvrir le panneau, double-cliquez sur la carte. Vous pouvez également sélectionner la carte, puis cliquez sur Détails.
2. Dans le panneau des propriétés des ressources, sous Routes, procédez comme suit :

Note

Pour chacune des routes suivantes, utilisez la HTTP méthode et les valeurs de chemin spécifiées dans le [tableau de référence des propriétés des ressources](#).

- a. Pour Méthode, choisissez la HTTP méthode spécifiée. Par exemple, GET.
- b. Dans le champ Path, entrez le chemin spécifié. Par exemple, **/items**.
- c. Choisissez Ajouter une route.

- d. Répétez les étapes précédentes jusqu'à ce que vous ayez ajouté les cinq itinéraires spécifiés.
3. Choisissez Save (Enregistrer).



Étape 4 : Configuration de vos fonctions Lambda

Nommez chacune des cinq fonctions Lambda comme indiqué dans le tableau de [référence des propriétés des ressources](#).

Pour nommer les fonctions Lambda

1. Ouvrez le panneau des propriétés des ressources d'une carte de fonction Lambda. Pour ouvrir le panneau, double-cliquez sur la carte. Vous pouvez également sélectionner la carte, puis cliquez sur Détails.
2. Dans le panneau des propriétés de la ressource, pour Logical ID, entrez un nom de fonction spécifié. Par exemple, **getItems**.
3. Choisissez Save (Enregistrer).
4. Répétez les étapes précédentes jusqu'à ce que vous ayez nommé les cinq fonctions.

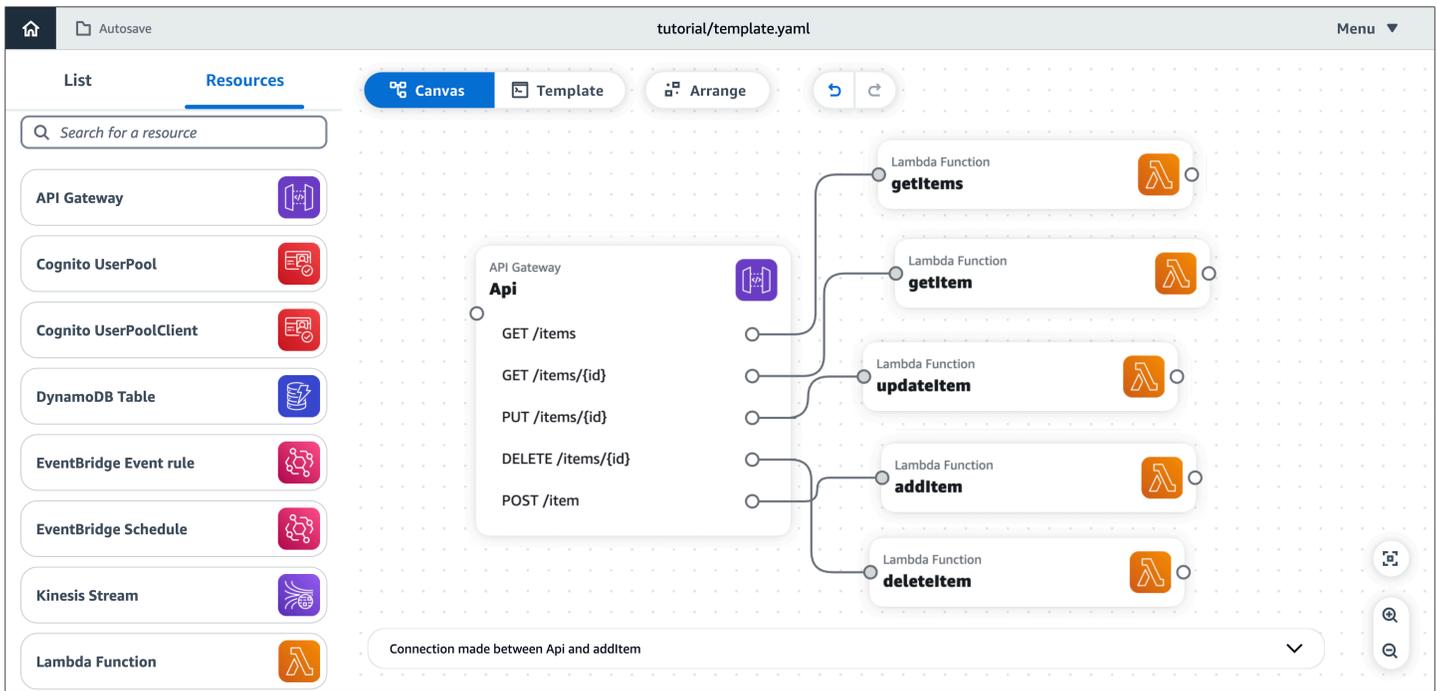
The screenshot displays the AWS Infrastructure Composer interface for a template named 'tutorial/template.yaml'. On the left, a 'Resources' panel lists various AWS services. The main canvas shows an 'API Gateway' resource named 'Api' with five routes. These routes are connected to five 'Lambda Function'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

Étape 5 : Connectez vos cartes

Connectez chaque route de votre carte APIGateway à la carte de fonction Lambda correspondante, comme indiqué dans le tableau de [référence des propriétés des ressources](#).

Pour connecter vos cartes

1. Cliquez sur un port droit de la carte APIGateway et faites-le glisser vers le port gauche de la carte de fonction Lambda spécifiée. Par exemple, cliquez sur le port GET/items et faites-le glisser vers le port gauche de getItems.
2. Répétez l'étape précédente jusqu'à ce que vous ayez connecté les cinq routes de la carte APIGateway aux cartes Lambda Function correspondantes.



Étape 6 : Organiser le canevas

Organisez le canevas visuel en regroupant vos fonctions Lambda et en organisant toutes les cartes.

Pour regrouper vos fonctions

1. Maintenez la touche Shift enfoncée, puis sélectionnez chaque carte de fonction Lambda sur le canevas.
2. Choisissez le groupe.

Pour donner un nom à votre groupe

1. Double-cliquez en haut du groupe, près du nom du groupe (Groupe).

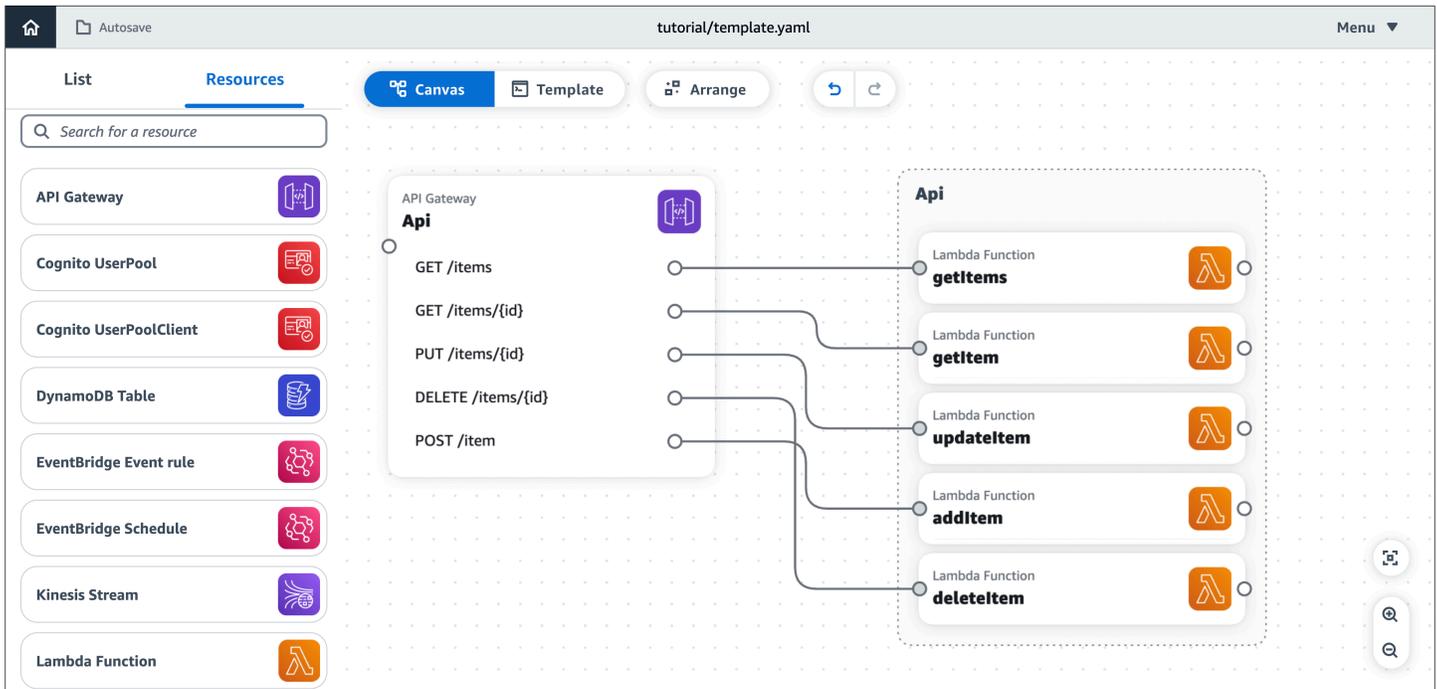
Le panneau des propriétés du groupe s'ouvre.

2. Dans le panneau Propriétés du groupe, dans Nom du groupe, entrez **API**.
3. Choisissez Save (Enregistrer).

Pour organiser vos cartes

Sur le canevas, au-dessus de la zone d'affichage principale, choisissez Arrangement.

Infrastructure Composer organise et aligne toutes les cartes sur le canevas visuel, y compris votre nouveau groupe (API), comme indiqué ici :

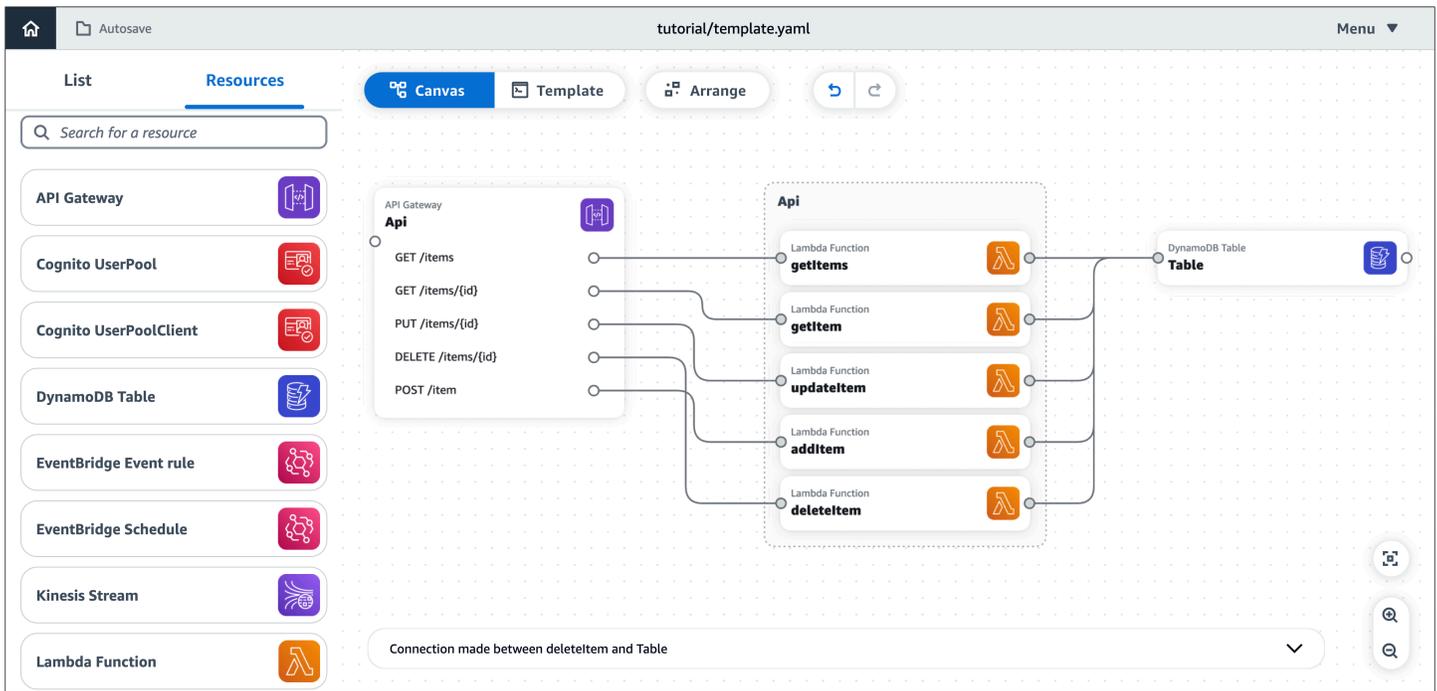


Étape 7 : Ajouter et connecter une table DynamoDB

Ajoutez maintenant une table DynamoDB à l'architecture de votre application et connectez-la à vos fonctions Lambda.

Pour ajouter et connecter une table DynamoDB

1. Dans la palette de ressources (Ressources), sous la section Composants améliorés, faites glisser une carte DynamoDB Table sur le canevas.
2. Cliquez sur le port droit d'une carte de fonction Lambda et faites-le glisser vers le port gauche de la carte DynamoDB Table.
3. Répétez l'étape précédente jusqu'à ce que vous ayez connecté les cinq cartes Lambda Function à la carte DynamoDB Table.
4. (Facultatif) Pour réorganiser et réaligner les cartes sur le canevas, choisissez Arrangement.

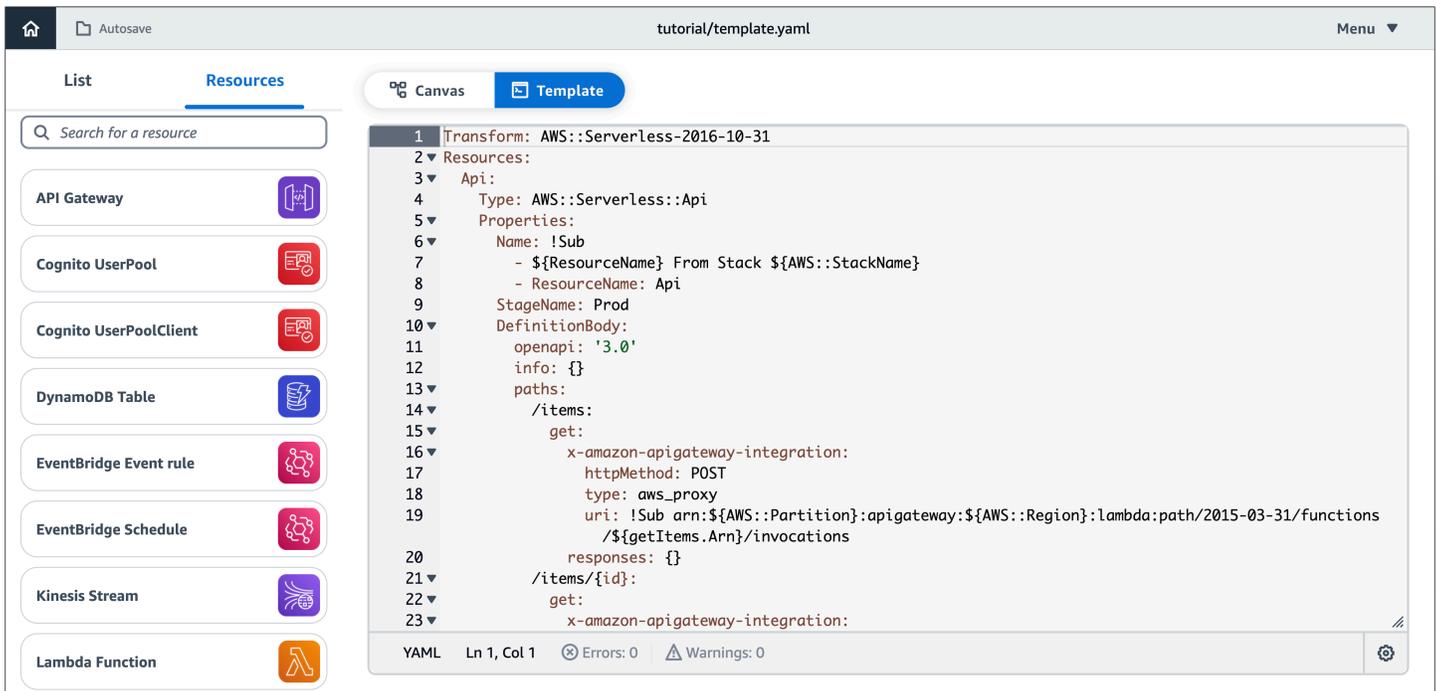


Étape 8 : passez en revue votre AWS CloudFormation modèle

Félicitations ! Vous avez conçu avec succès une application sans serveur prête à être déployée. Enfin, choisissez Modèle pour consulter le AWS CloudFormation modèle qu'Infrastructure Composer a automatiquement généré pour vous.

Dans le modèle, Infrastructure Composer a défini les éléments suivants :

- La Transform déclaration, qui spécifie le modèle en tant que modèle AWS Serverless Application Model (AWS SAM). Pour plus d'informations, consultez [l'anatomie du AWS SAM modèle](#) dans le guide du AWS Serverless Application Model développeur.
- Une `AWS::Serverless::Api` ressource qui indique votre API passerelle REST API avec ses cinq routes.
- Cinq `AWS::Serverless::Function` ressources, qui spécifient les configurations de vos fonctions Lambda, y compris leurs variables d'environnement et leurs politiques d'autorisation.
- Une `AWS::DynamoDB::Table` ressource qui spécifie votre table DynamoDB et ses propriétés.
- La Metadata section, qui contient des informations sur votre groupe de ressources (API). Pour plus d'informations sur cette section, consultez la section [Métadonnées](#) dans le guide de AWS CloudFormation l'utilisateur.



Étape 9 : Intégrez à vos flux de travail de développement

Utilisez le fichier modèle et les répertoires de projet créés par Infrastructure Composer pour des tests et des déploiements plus approfondis.

- Grâce à la synchronisation locale, vous pouvez connecter Infrastructure Composer IDE à votre machine locale pour accélérer le développement. Pour en savoir plus, consultez [Connectez la console Infrastructure Composer à votre console locale IDE](#).
- Avec la synchronisation locale, vous pouvez utiliser l'interface de ligne de commande de AWS Serverless Application Model (AWS SAM CLI) sur votre machine locale pour tester et déployer votre application. Pour en savoir plus, consultez [Déployez votre application sans serveur Infrastructure Composer dans le cloud AWS](#).

Étapes suivantes

Vous êtes maintenant prêt à créer vos propres applications avec Infrastructure Composer. Pour plus de détails sur l'utilisation d'Infrastructure Composer, reportez-vous à [Comment composer AWS Infrastructure Composer](#). Lorsque vous êtes prêt à déployer votre application, reportez-vous à [Déployez votre application sans serveur Infrastructure Composer dans le cloud AWS](#).

Où vous pouvez utiliser Infrastructure Composer

Vous pouvez utiliser Infrastructure Composer depuis sa console AWS Toolkit for Visual Studio Code, depuis et dans Infrastructure Composer en mode CloudFormation console. Bien que chacune varie selon des cas d'utilisation légèrement différents, il s'agit dans l'ensemble d'expériences similaires. Cette section fournit des détails sur chaque expérience.

Cette rubrique présente [Utilisation de la AWS Infrastructure Composer console](#) une présentation complète de l'expérience par défaut de la console. [CloudFormation mode console](#) Cette rubrique fournit des détails sur une version d'Infrastructure Composer intégrée au flux de travail de AWS CloudFormation stack. [AWS Toolkit for Visual Studio Code](#) fournit des informations sur l'accès et l'utilisation d'Infrastructure Composer dans VS Code.

Rubriques

- [Utilisation de la AWS Infrastructure Composer console](#)
- [Utilisation d'Infrastructure Composer en mode CloudFormation console](#)
- [À l'aide de Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code](#)

Utilisation de la AWS Infrastructure Composer console

Cette section fournit des détails sur l'accès et l'utilisation AWS Infrastructure Composer depuis la console Infrastructure Composer. Il s'agit de l'expérience par défaut pour Infrastructure Composer et c'est un bon moyen de se familiariser avec Infrastructure Composer. Vous pouvez également intégrer la console Infrastructure Composer à votre console locale IDE. Pour plus de détails, consultez [Connectez la console Infrastructure Composer à votre console locale IDE](#).

Vous pouvez également [accéder à Infrastructure Composer à partir du AWS kit d'outils de VS Code](#), et vous pouvez utiliser un [mode d'Infrastructure Composer spécialement conçu pour être utilisé dans AWS CloudFormation](#).

Pour une documentation générale sur l'utilisation d'Infrastructure Composer, consultez [Comment composer](#).

Rubriques

- [AWS Infrastructure Composer aperçu visuel de la console](#)
- [Gérez votre projet depuis la console Infrastructure Composer](#)

- [Connectez la console Infrastructure Composer à votre console locale IDE](#)
- [Autoriser l'accès aux pages Web aux fichiers locaux dans Infrastructure Composer](#)
- [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#)
- [Importer des fonctions dans Infrastructure Composer depuis la console Lambda](#)
- [Exporter une image du canevas visuel d'Infrastructure Composer](#)

AWS Infrastructure Composer aperçu visuel de la console

Cette section fournit un aperçu visuel de la AWS Infrastructure Composer console.

Rubriques

- [Page d'accueil](#)
- [Designer visuel et toile visuelle](#)

Page d'accueil

L'image suivante représente la page d'accueil de la console Infrastructure Composer :

The screenshot shows the AWS Infrastructure Composer console interface. 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 is titled 'Infrastructure Composer' and includes a 'New!' notification stating that the service is now part of the AWS Toolkit for Visual Studio Code. Below the notification are three buttons: 'Download VS Code Extension', 'Open demo', and 'Create project'. The 'About' section describes the service'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

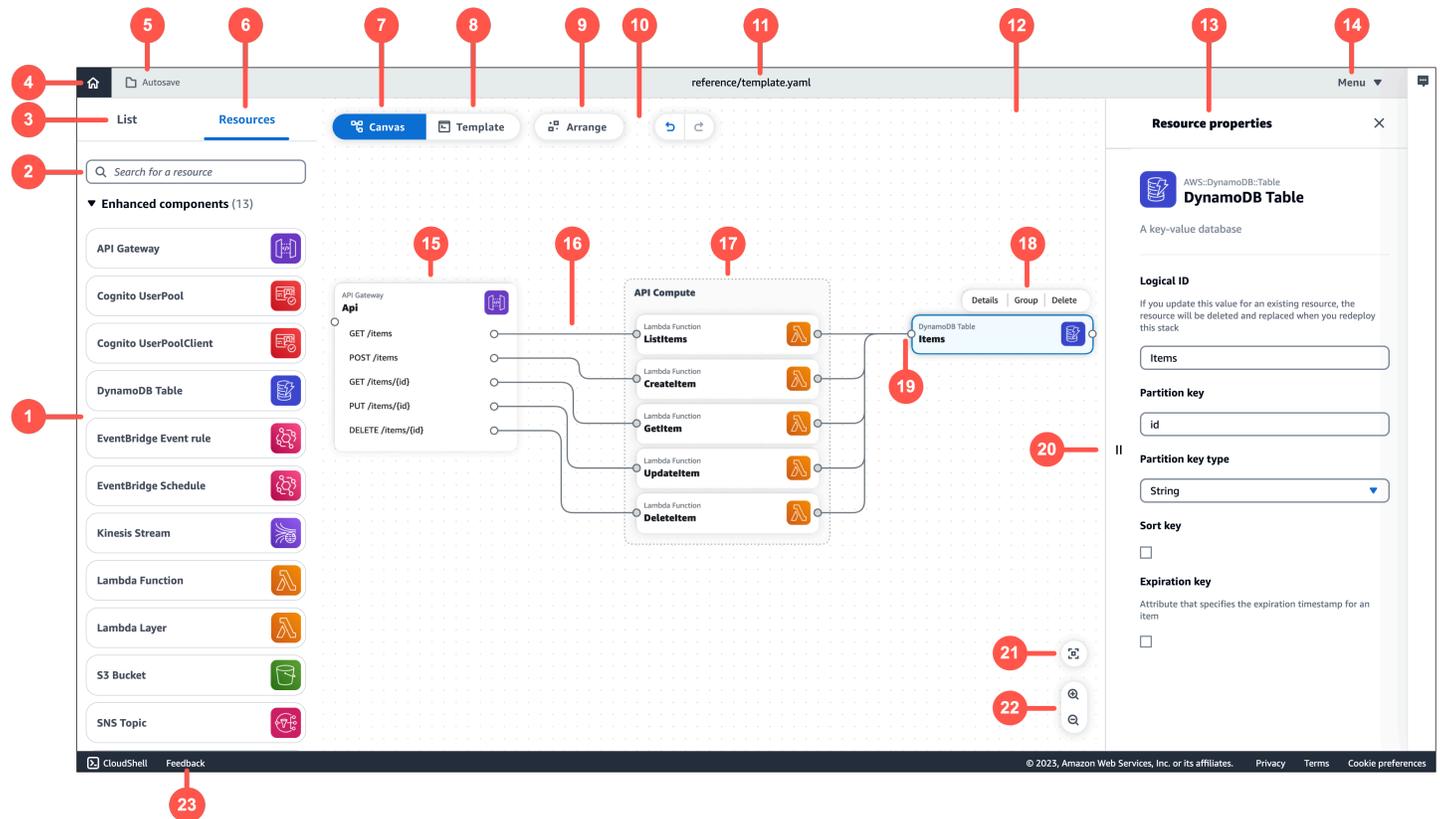
 At the bottom of the 'About' section is a 'Start building' section with a plus icon. To the right of the text is a large visual canvas displaying a complex architecture diagram. The diagram shows various AWS services interconnected, including API Gateway, Lambda, S3, IAM, and others, illustrating the visual design capabilities of the tool.

1. Documentation — Accédez à la documentation d'Infrastructure Composer.
2. Canevas : accédez au canevas et créez ou chargez un projet.
3. Démo — Ouvrez l'application de démonstration Infrastructure Composer.

4. Créer un projet : créez ou chargez un projet.
5. Commencer à créer : liens rapides pour démarrer la création d'une application.
6. Commentaires — Cliquez ici pour envoyer des commentaires.

Designer visuel et toile visuelle

L'image suivante représente le concepteur visuel et le canevas visuel d'Infrastructure Composer :



1. Palette de ressources : affiche les cartes que vous pouvez utiliser pour créer.
2. Barre de recherche de ressources : recherchez des cartes que vous pouvez ajouter au canevas.
3. Liste : affiche une arborescence des ressources de votre application.
4. Accueil — Cliquez ici pour accéder à la page d'accueil d'Infrastructure Composer.
5. État de sauvegarde : indique si les modifications d'Infrastructure Composer sont enregistrées sur votre machine locale. Les États incluent :
 - Enregistrement automatique : la synchronisation locale est activée et votre projet est automatiquement synchronisé et enregistré.
 - Modifications enregistrées : votre modèle de demande est enregistré sur votre ordinateur local.

- Modifications non enregistrées : votre modèle d'application contient des modifications qui ne sont pas enregistrées sur votre ordinateur local.
6. Ressources : affiche la palette de ressources.
 7. Canevas : affiche la vue du canevas de votre application dans la zone d'affichage principale.
 8. Modèle : affiche la vue du modèle de votre application dans la zone d'affichage principale.
 9. Organiser : organise l'architecture de votre application dans le canevas.
 10. Annuler et rétablir : effectuez des actions d'annulation et de rétablissement lorsque cela est possible.
 11. Nom du modèle : indique le nom du modèle que vous êtes en train de concevoir.
 12. Zone d'affichage principale : affiche le canevas ou le modèle en fonction de votre sélection.
 13. Panneau des propriétés des ressources : affiche les propriétés pertinentes pour la carte sélectionnée dans le canevas. Ce panneau est dynamique. Les propriétés affichées changeront au fur et à mesure que vous configurerez votre carte.
 14. Menu — Fournit des options générales telles que les suivantes :
 - Créer un projet
 - Ouvrir un fichier modèle ou un projet
 - Enregistrer un fichier modèle
 - [Activer la synchronisation locale](#)
 - [Toile d'exportation](#)
 - Obtenez de l'aide
 - Raccourcis clavier
 15. Carte : affiche une vue de votre carte sur le canevas.
 16. Ligne — Représente une connexion entre des cartes.
 17. Groupe — Regroupe les cartes sélectionnées pour une organisation visuelle.
 18. Actions de la carte : indique les actions que vous pouvez effectuer sur votre carte.
 - a. Détails — Affiche le panneau des propriétés de la ressource.
 - b. Grouper — Regroupez les cartes sélectionnées.
 - c. Supprimer — Supprime la carte de votre canevas.
 19. Port : points de connexion à d'autres cartes.
 20. Champs de propriétés de ressources : ensemble organisé de champs de propriétés à configurer pour vos cartes.

21.Recentrer : recentrez votre schéma d'application sur le canevas visuel.

22.Zoom — Zoomez en avant et en arrière sur votre toile.

23.Commentaires — Cliquez ici pour envoyer des commentaires.

Gérez votre projet depuis la console Infrastructure Composer

Cette rubrique fournit des conseils sur les tâches de base que vous effectuez pour gérer votre projet à partir de la console Infrastructure Composer. Cela inclut les tâches courantes telles que la création d'un nouveau projet, l'enregistrement d'un projet et l'importation d'un projet ou d'un modèle. Vous pouvez également charger un projet existant si vous activez le [mode de synchronisation locale](#). Après avoir activé le mode de synchronisation locale, vous pouvez effectuer les opérations suivantes :

- Créez un nouveau projet composé d'un modèle de départ et d'une structure de dossiers.
- Chargez un projet existant en choisissant un dossier parent contenant votre modèle de projet et vos fichiers.
- Utilisez Infrastructure Composer pour gérer vos modèles et dossiers

Avec le mode de synchronisation locale, Infrastructure Composer enregistre automatiquement les modifications apportées au modèle et au dossier de votre projet sur votre machine locale. Si votre navigateur ne prend pas en charge le mode de synchronisation locale, ou si vous préférez utiliser Infrastructure Composer sans que le mode de synchronisation local soit activé, vous pouvez créer un nouveau modèle ou charger un modèle existant. Pour enregistrer les modifications, vous devez exporter le modèle vers votre ordinateur local.

Note

Infrastructure Composer prend en charge les applications suivantes :

- Un AWS Serverless Application Model modèle AWS CloudFormation OR qui définit votre code d'infrastructure.
- Structure de dossiers qui organise les fichiers de votre projet, tels que le code de fonction Lambda, les fichiers de configuration et les dossiers de compilation.

Rubriques

- [Création d'un nouveau projet dans la console Infrastructure Composer](#)

- [Importer un dossier de projet existant dans la console Infrastructure Composer](#)
- [Importer un modèle de projet existant dans la console Infrastructure Composer](#)
- [Enregistrer un modèle de projet existant dans la console Infrastructure Composer](#)

Création d'un nouveau projet dans la console Infrastructure Composer

Lorsque vous créez un nouveau projet, Infrastructure Composer génère un modèle de départ. Lorsque vous concevez votre application sur le canevas, votre modèle est modifié. Pour enregistrer votre travail, vous devez exporter votre modèle ou activer le mode de synchronisation locale.

Pour créer un projet

1. Connectez-vous à la [console Infrastructure Composer](#).
2. Sur la page d'accueil, choisissez Créer un projet.

Note

Vous pouvez également charger un fichier existant dans Infrastructure Composer, mais vous devez d'abord [activer le mode de synchronisation local](#). Une fois activé, voir [Chargez un projet Infrastructure Composer existant avec la synchronisation locale activée](#) comment charger un projet existant.

Importer un dossier de projet existant dans la console Infrastructure Composer

En mode de synchronisation locale, vous pouvez importer le dossier parent d'un projet existant. Si votre projet contient plusieurs modèles, vous pouvez choisir le modèle à charger.

Pour importer un projet existant depuis la page d'accueil

1. Connectez-vous à la [console Infrastructure Composer](#).
2. Sur la page d'accueil, choisissez Charger un CloudFormation modèle.
3. Pour Emplacement du projet, choisissez Sélectionner un dossier. Sélectionnez le dossier parent de votre projet, puis sélectionnez Sélectionner.

 Note

Si vous ne recevez pas cette invite, il est possible que votre navigateur ne prenne pas en charge l'accès au système de fichiersAPI, qui est requis pour le mode de synchronisation local. Pour de plus amples informations, veuillez consulter [Autoriser l'accès aux pages Web aux fichiers locaux dans Infrastructure Composer](#).

4. Lorsque votre navigateur vous le demande, sélectionnez Afficher les fichiers.
5. Pour Fichier modèle, choisissez votre modèle dans la liste déroulante. Si votre projet contient un seul modèle, Infrastructure Composer le sélectionne automatiquement pour vous.
6. Sélectionnez Create (Créer).

Pour importer un projet existant depuis le canevas

1. Dans le canevas, choisissez Menu pour ouvrir le menu.
2. Dans la section Ouvrir, choisissez le dossier du projet.

 Note

Si l'option Dossier de projet n'est pas disponible, il est possible que votre navigateur ne prenne pas en charge l'accès au système de fichiersAPI, qui est requis pour le mode de synchronisation local. Pour de plus amples informations, veuillez consulter [Autoriser l'accès aux pages Web aux fichiers locaux dans Infrastructure Composer](#).

3. Pour Emplacement du projet, choisissez Sélectionner un dossier. Sélectionnez le dossier parent de votre projet, puis sélectionnez Sélectionner.
4. Lorsque votre navigateur vous le demande, sélectionnez Afficher les fichiers.
5. Pour Fichier modèle, choisissez votre modèle dans la liste déroulante. Si votre projet contient un seul modèle, Infrastructure Composer le sélectionne automatiquement pour vous.
6. Sélectionnez Create (Créer).

Lorsque vous importez un dossier de projet existant, Infrastructure Composer active le mode de synchronisation locale. Les modifications apportées au modèle ou aux fichiers de votre projet sont automatiquement enregistrées sur votre ordinateur local.

Importer un modèle de projet existant dans la console Infrastructure Composer

Lorsque vous importez un AWS SAM modèle AWS CloudFormation ou un modèle existant, Infrastructure Composer génère automatiquement une visualisation de l'architecture de votre application sur le canevas.

Vous pouvez importer un modèle de projet depuis votre machine locale.

Pour importer un modèle de projet existant

1. Connectez-vous à la [console Infrastructure Composer](#).
2. Choisissez Créer un projet pour ouvrir un canevas vierge.
3. Choisissez Menu pour ouvrir le menu.
4. Dans la section Ouvrir, sélectionnez Fichier modèle.
5. Sélectionnez votre modèle et choisissez Ouvrir.

Pour enregistrer les modifications apportées à votre modèle, vous devez l'exporter ou activer le mode de synchronisation locale.

Enregistrer un modèle de projet existant dans la console Infrastructure Composer

Si vous n'utilisez pas le mode de synchronisation local, vous devez exporter votre modèle pour enregistrer vos modifications. Si le mode de synchronisation locale est activé, il n'est pas nécessaire d'enregistrer manuellement votre modèle. Les modifications sont automatiquement enregistrées sur votre ordinateur local.

Pour enregistrer un modèle de projet existant

1. Dans le canevas Infrastructure Composer, choisissez Menu pour ouvrir le menu.
2. Dans la section Enregistrer, choisissez Enregistrer le fichier modèle.
3. Donnez un nom à votre modèle.
4. Sélectionnez un emplacement pour enregistrer votre modèle.
5. Choisissez Save (Enregistrer).

Connectez la console Infrastructure Composer à votre console locale IDE

Pour connecter la console Infrastructure Composer à votre environnement de développement intégré local (IDE), utilisez le mode de synchronisation local. Ce mode synchronise et enregistre automatiquement les données sur votre machine locale. Pour plus d'informations sur le mode de synchronisation locale, consultez [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#). Pour obtenir des instructions sur l'utilisation du mode de synchronisation local, consultez [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#).

Note

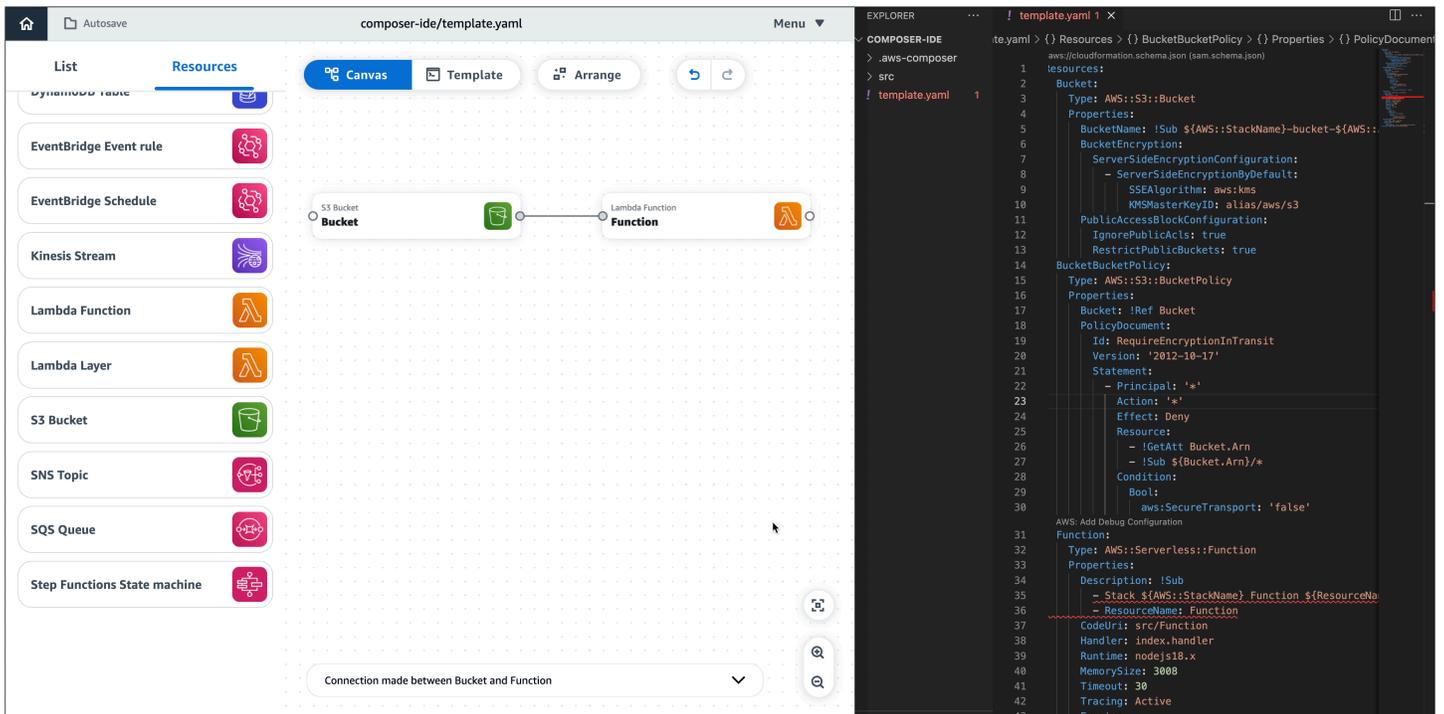
L'option Activer la synchronisation locale n'est pas disponible dans tous les navigateurs. Il est disponible dans Google Chrome et Microsoft Edge.

Avantages de l'utilisation d'Infrastructure Composer avec votre environnement local IDE

Lorsque vous concevez dans Infrastructure Composer, votre modèle local et votre répertoire de projet sont automatiquement synchronisés et enregistrés.

Vous pouvez utiliser votre local IDE pour afficher les modifications et modifier vos modèles. Les modifications que vous apportez localement sont automatiquement synchronisées avec Infrastructure Composer.

Vous pouvez utiliser des outils locaux tels que l'interface de ligne de commande de AWS Serverless Application Model (AWS SAM CLI) pour créer, tester, déployer votre application, etc. L'exemple suivant montre comment vous pouvez glisser-déposer des ressources sur le canevas visuel d'Infrastructure Composer qui, à son tour, crée un balisage dans votre AWS SAM modèle dans votre local IDE.



Intégrez Infrastructure Composer à votre environnement local IDE

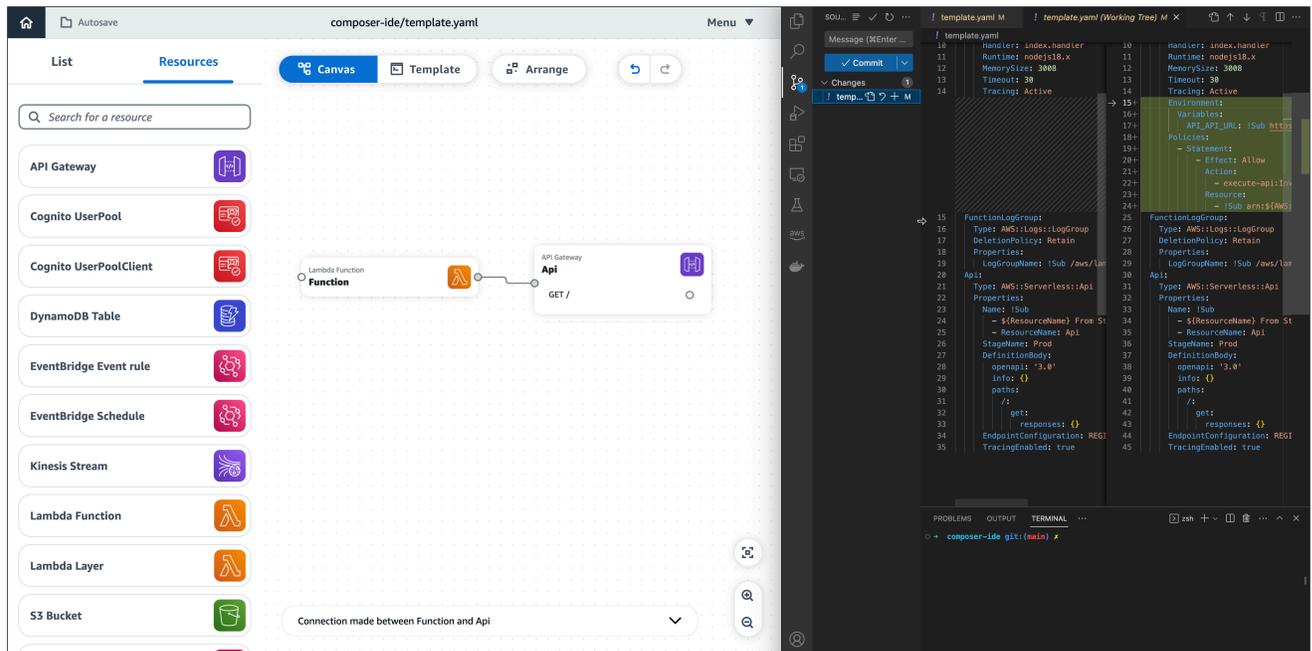
Pour intégrer Infrastructure Composer à votre environnement local IDE

1. Dans Infrastructure Composer, créez ou chargez un projet, puis activez la synchronisation locale en sélectionnant le bouton Menu en haut à droite de l'écran, puis en choisissant Activer la synchronisation locale.

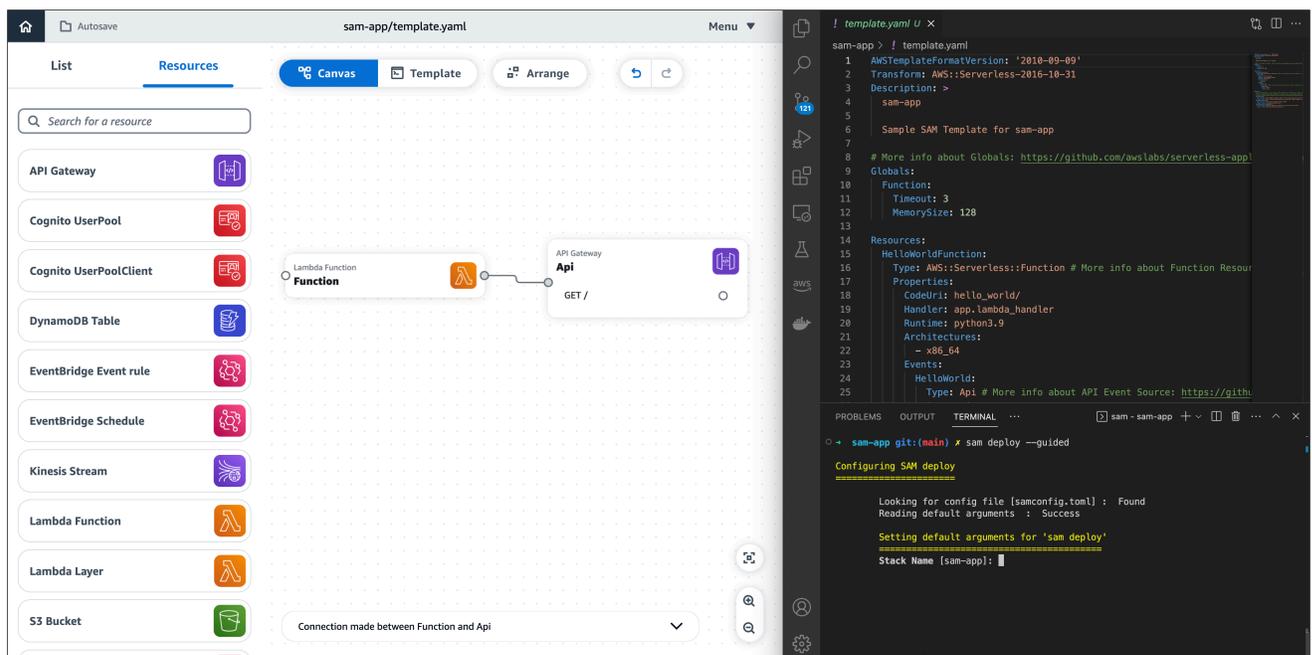
Note

L'option Activer la synchronisation locale n'est pas disponible dans tous les navigateurs. Il est disponible dans Google Chrome et Microsoft Edge.

2. Dans votre localIDE, ouvrez le même dossier de projet qu'Infrastructure Composer.
3. Utilisez Infrastructure Composer avec votre localIDE. Les mises à jour effectuées dans Infrastructure Composer seront automatiquement synchronisées avec votre machine locale. Voici quelques exemples de ce que vous pouvez faire :
 - a. Utilisez le système de contrôle de version de votre choix pour suivre les mises à jour effectuées par Infrastructure Composer.



- b. Utilisez-le AWS SAM CLI localement pour créer, tester, déployer votre application, etc. Pour en savoir plus, consultez [Déployez votre application sans serveur Infrastructure Composer dans le cloud AWS.](#)



Autoriser l'accès aux pages Web aux fichiers locaux dans Infrastructure Composer

La console Infrastructure Composer prend en charge le [mode de synchronisation locale](#) et [les fonctions d'importation depuis la console Lambda](#). Pour utiliser ces fonctionnalités, un navigateur Web prenant en charge l'accès au système de fichiers API est requis. Toute version récente de Google Chrome et Microsoft Edge prend en charge toutes les fonctionnalités de File System Access API et peut être utilisée avec le mode de synchronisation locale dans Infrastructure Composer.

L'accès au système de fichiers API permet aux pages Web d'accéder à votre système de fichiers local afin de lire, d'écrire ou d'enregistrer des fichiers. Cette fonctionnalité est désactivée par défaut et nécessite votre autorisation par le biais d'un message visuel pour l'autoriser. Une fois accordé, cet accès est maintenu pendant toute la durée de la session de navigation de votre page Web.

Pour en savoir plus sur l'accès au système de fichiers API, voir :

- [Accès au système de fichiers API](#) dans les documents Web MDN.
- [L'accès au système de fichiers API : simplifier l'accès aux fichiers locaux](#) sur le site web.dev.

mode de synchronisation local

Le mode de synchronisation locale vous permet de synchroniser et d'enregistrer automatiquement vos fichiers modèles et dossiers de projet localement lorsque vous concevez dans Infrastructure Composer. Pour utiliser cette fonctionnalité, un navigateur Web prenant en charge l'accès au système de fichiers API est requis.

Data Infrastructure Composer a accès à

Infrastructure Composer obtient un accès en lecture et en écriture au dossier de projet que vous autorisez, ainsi qu'à tous les dossiers enfants de ce dossier de projet. Cet accès est utilisé pour créer, mettre à jour et enregistrer les fichiers modèles, les dossiers de projet et les répertoires de sauvegarde générés lors de la conception. Les données accessibles par Infrastructure Composer ne sont utilisées à aucune autre fin et ne sont stockées nulle part en dehors de votre système de fichiers local.

Accès aux données sensibles

L'accès au système de fichiers API exclut ou limite l'accès à des répertoires spécifiques susceptibles de contenir des données sensibles. Une erreur se produira si vous sélectionnez l'un de ces

répertoires à utiliser avec le mode de synchronisation local d'Infrastructure Composer. Vous pouvez choisir un autre répertoire local auquel vous connecter ou utiliser Infrastructure Composer dans son mode par défaut avec la synchronisation locale désactivée.

Pour plus d'informations, y compris des exemples de répertoires sensibles, voir [Utilisateurs donnant accès à des fichiers plus ou plus sensibles que prévu dans le projet de](#) rapport du groupe communautaire du W3C sur l'accès au système de fichiers.

Si vous utilisez Windows Subsystem for Linux (WSL), l'accès au système de fichiers API exclut l'accès à l'ensemble Linux répertoire en raison de son emplacement dans votre Windows système. Vous pouvez utiliser Infrastructure Composer avec la synchronisation locale désactivée ou configurer une solution pour synchroniser les fichiers de projet depuis votre WSL répertoire vers un répertoire de travail dans Windows. Utilisez ensuite le mode de synchronisation locale d'Infrastructure Composer avec votre Windows annuaire.

Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer

Cette section fournit des informations sur l'utilisation du mode de synchronisation local d'Infrastructure Composer pour synchroniser et enregistrer automatiquement votre projet sur votre machine locale.

Nous vous recommandons d'utiliser la synchronisation locale pour les raisons suivantes :

Vous pouvez activer la synchronisation locale pour un nouveau projet ou charger un projet existant avec la synchronisation locale activée.

- Par défaut, vous devez enregistrer manuellement votre modèle d'application au fur et à mesure de sa conception. Utilisez la synchronisation locale pour enregistrer automatiquement votre modèle d'application sur votre machine locale à mesure que vous apportez des modifications.
- La synchronisation locale gère et synchronise automatiquement les dossiers de votre projet, votre dossier de sauvegarde et [les fichiers externes pris en charge](#) sur votre machine locale.
- Lorsque vous utilisez la synchronisation locale, vous pouvez connecter Infrastructure Composer à votre local IDE pour accélérer le développement. Pour en savoir plus, consultez [Connectez la console Infrastructure Composer à votre console locale IDE](#).

Ce que le mode de synchronisation locale permet d'économiser

Le mode de synchronisation locale synchronise et enregistre automatiquement les éléments suivants sur votre ordinateur local :

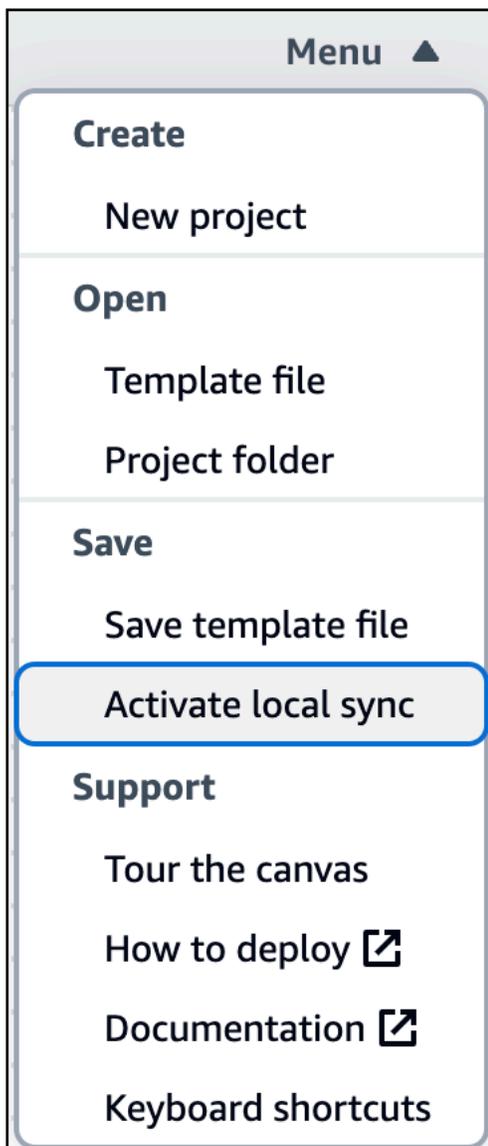
- Fichier de modèle d'application : modèle AWS CloudFormation ou AWS Serverless Application Model (AWS SAM) qui contient votre infrastructure sous forme de code (IaC).
- Dossiers de projets : structure de répertoire générale qui organise vos AWS Lambda fonctions.
- Répertoire de sauvegarde : répertoire de sauvegarde nommé `.aws-composer`, créé à la racine de l'emplacement de votre projet. Ce répertoire contient une copie de sauvegarde du fichier de modèle d'application et des dossiers de projet.
- Fichiers externes : fichiers externes pris en charge que vous pouvez utiliser dans Infrastructure Composer. Pour en savoir plus, consultez [Référencer des fichiers externes dans Infrastructure Composer](#).

Configuration requise pour le navigateur

Le mode de synchronisation locale nécessite un navigateur qui prend en charge l'accès au système de fichiers API. Pour de plus amples informations, veuillez consulter [Autoriser l'accès aux pages Web aux fichiers locaux dans Infrastructure Composer](#).

Activation du mode de synchronisation local

Le mode de synchronisation locale est désactivé par défaut. Vous pouvez activer le mode de synchronisation locale via le menu Infrastructure Composer.



Pour obtenir des instructions sur l'activation de la synchronisation locale et les projets de chargement existants, consultez les rubriques suivantes :

- [Activer la synchronisation locale dans Infrastructure Composer](#)
- [Chargez un projet Infrastructure Composer existant avec la synchronisation locale activée](#)

Activer la synchronisation locale dans Infrastructure Composer

Pour activer la synchronisation locale, procédez comme suit :

1. Sur la page d'[accueil](#) d'Infrastructure Composer, sélectionnez Créer un projet.

2. Dans le menu Infrastructure Composer, sélectionnez Activer la synchronisation locale.
3. Pour l'emplacement du projet, appuyez sur Sélectionner un dossier et choisissez un répertoire. C'est ici qu'Infrastructure Composer enregistre et synchronise vos fichiers modèles et dossiers au fur et à mesure de votre conception.

 Note

L'emplacement du projet ne doit pas contenir de modèle de candidature existant.

4. Lorsque vous êtes invité à autoriser l'accès, sélectionnez Afficher les fichiers.
5. Appuyez sur Activer. Lorsque vous êtes invité à enregistrer les modifications, sélectionnez Enregistrer les modifications.

Lorsqu'il est activé, l'indicateur de sauvegarde automatique s'affiche dans le coin supérieur gauche de votre canevas.

Chargez un projet Infrastructure Composer existant avec la synchronisation locale activée

Pour charger un projet existant avec la synchronisation locale activée, procédez comme suit :

1. Sur la page d'[accueil](#) d'Infrastructure Composer, sélectionnez Charger un AWS CloudFormation modèle.
2. Dans le menu Infrastructure Composer, sélectionnez Ouvrir > Dossier du projet.
3. Pour l'emplacement du projet, appuyez sur Sélectionner un dossier et choisissez le dossier racine de votre projet.
4. Lorsque vous êtes invité à autoriser l'accès, sélectionnez Afficher les fichiers.
5. Pour le fichier modèle, sélectionnez votre modèle d'application et appuyez sur Créer.
6. Lorsque vous êtes invité à enregistrer les modifications, sélectionnez Enregistrer les modifications.

Lorsqu'il est activé, l'indicateur de sauvegarde automatique s'affiche dans le coin supérieur gauche de votre canevas.

Importer des fonctions dans Infrastructure Composer depuis la console Lambda

Infrastructure Composer fournit une intégration avec la AWS Lambda console. Vous pouvez importer une fonction Lambda depuis la console Lambda vers la console Infrastructure Composer. Utilisez ensuite le canevas Infrastructure Composer pour poursuivre la conception de l'architecture de votre application.

- Cette intégration nécessite un navigateur qui prend en charge l'accès au système de fichiers API. Pour de plus amples informations, veuillez consulter [Autoriser l'accès aux pages Web aux fichiers locaux dans Infrastructure Composer](#).
- Lorsque vous importez votre fonction Lambda dans Infrastructure Composer, vous devez activer le mode de synchronisation local pour enregistrer les modifications. Pour de plus amples informations, veuillez consulter [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#).

Pour commencer à utiliser cette intégration, consultez la section [Utilisation AWS Lambda avec AWS Infrastructure Composer](#) dans le guide du AWS Lambda développeur.

Exporter une image du canevas visuel d'Infrastructure Composer

Cette rubrique décrit la fonctionnalité de canevas d'exportation de la AWS Infrastructure Composer console.

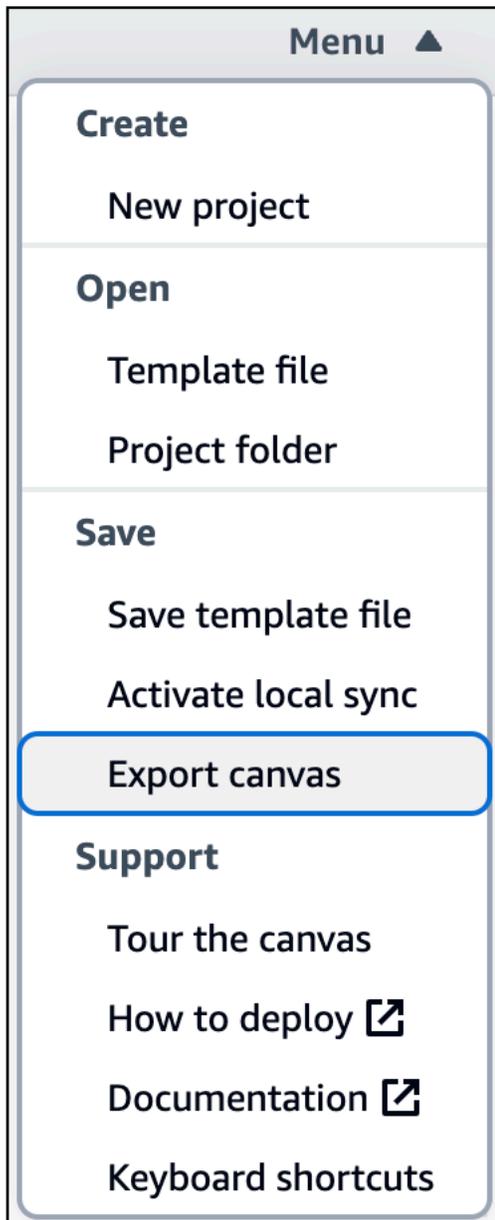
Pour un aperçu visuel de toutes les fonctionnalités d'Infrastructure Composer, voir [AWS Infrastructure Composer aperçu visuel de la console](#).

À propos de Export Canvas

La fonction d'exportation du canevas exporte le canevas de votre application sous forme d'image vers votre machine locale.

- Infrastructure Composer supprime les éléments de l'interface utilisateur du concepteur visuel et exporte uniquement le diagramme de votre application.
- Le format de fichier image par défaut est png.
- Le fichier est exporté vers l'emplacement de téléchargement par défaut de votre ordinateur local.

Vous pouvez accéder à la fonction d'exportation du canevas depuis le menu.



Exportation de toile

Lorsque vous exportez votre canevas, Infrastructure Composer affiche un message d'état.

Si l'exportation est réussie, le message suivant s'affiche :



Canvas export successful
Check your downloads folder.

En cas d'échec de l'exportation, un message d'erreur s'affichera. Si un message d'erreur s'affiche, réessayez d'exporter.



Canvas export error
Unexpected error occurred

Utilisation d'Infrastructure Composer en mode CloudFormation console

Infrastructure Composer en mode CloudFormation console est l'outil recommandé pour visualiser vos AWS CloudFormation modèles. Vous pouvez également utiliser cet outil pour créer et modifier AWS CloudFormation des modèles.

En quoi ce mode est-il différent de celui de la console Infrastructure Composer ?

Infrastructure Composer en mode CloudFormation console possède généralement les mêmes fonctionnalités que la [console Infrastructure Composer par défaut](#), à quelques différences près.

- Ce mode est intégré au flux de travail de pile de la AWS CloudFormation console. Cela vous permet d'utiliser Infrastructure Composer directement dans AWS CloudFormation.
- [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#), une fonctionnalité qui synchronise et enregistre automatiquement les données sur votre machine locale, n'est pas prise en charge.

- Les cartes associées à Lambda (fonction Lambda et couche Lambda) nécessitent des compilations de code et des solutions de packaging qui ne sont pas disponibles dans ce mode.

Note

Ces cartes et la synchronisation locale peuvent être utilisées dans la [console Infrastructure Composer](#) ou le AWS Toolkit for Visual Studio Code.

Lorsque vous ouvrez Infrastructure Composer depuis la AWS CloudFormation console, Infrastructure Composer s'ouvre en mode CloudFormation console. Dans ce mode, vous pouvez utiliser Infrastructure Composer pour visualiser, créer et mettre à jour vos modèles.

Comment accéder à Infrastructure Composer en mode CloudFormation console

Infrastructure Composer en mode CloudFormation console est une mise à niveau de AWS CloudFormation Designer. Nous vous recommandons d'utiliser Infrastructure Composer pour visualiser vos AWS CloudFormation modèles. Vous pouvez également utiliser cet outil pour créer et modifier AWS CloudFormation des modèles.

1. Accédez à la [console Cloudformation](#) et connectez-vous.
2. Sélectionnez Infrastructure Composer dans le menu de navigation de gauche. Cela vous amènera à Infrastructure Composer en mode CloudFormation console.

Note

Pour plus d'informations sur l'utilisation d'Infrastructure Composer en mode CloudFormation console, consultez [Utilisation d'Infrastructure Composer en mode CloudFormation console](#).

Visualisez un déploiement dans Infrastructure Composer en mode CloudFormation console

Suivez les instructions de cette rubrique pour visualiser un modèle AWS CloudFormation stack/ Infrastructure Composer déployé.

1. Accédez à la [AWS CloudFormation console](#) et connectez-vous.
2. Sélectionnez la pile que vous souhaitez modifier.
3. Sélectionnez l'onglet Modèle.
4. Sélectionnez Infrastructure Composer.

Infrastructure Composer visualisera votre stack/modèle. Des modifications peuvent également être apportées ici.

Création d'un nouveau modèle dans Infrastructure Composer en mode CloudFormation console

Suivez les instructions de cette rubrique pour créer un nouveau modèle.

1. Accédez à la [AWS CloudFormation console](#) et connectez-vous.
2. Sélectionnez Infrastructure Composer dans le menu de navigation de gauche. Cela permettra d'ouvrir Infrastructure Composer en mode CloudFormation console.
3. Faites glisser, déposez, configurez et connectez les ressources ([cartes](#)) dont vous avez besoin à partir de la palette Ressources.

Note

Consultez [Comment composer](#) pour plus de détails sur l'utilisation d'Infrastructure Composer et notez que les cartes associées à Lambda (fonction Lambda et couche Lambda) nécessitent des versions de code et des solutions de packaging qui ne sont pas disponibles dans Infrastructure Composer en mode console. CloudFormation Ces cartes peuvent être utilisées dans la [console Infrastructure Composer](#) ou dans le AWS Toolkit for Visual Studio Code. Pour plus d'informations sur l'utilisation de ces outils, reportez-vous à [Où vous pouvez utiliser Infrastructure Composer](#).

4. Double-cliquez sur les cartes pour utiliser le panneau des propriétés des ressources afin de définir la manière dont les cartes sont configurées.
5. [Connectez vos cartes](#) pour définir le flux de travail piloté par les événements de votre application.
6. Sélectionnez Modèle pour afficher et modifier votre code d'infrastructure. Les modifications sont automatiquement synchronisées avec la vue de votre canevas.
7. Une fois que votre modèle est prêt à être exporté dans une pile, sélectionnez Créer un modèle.

8. Cliquez sur le CloudFormation bouton Confirmer et exporter vers. Cela vous ramènera au flux de travail de création de pile avec un message confirmant que votre modèle a été importé avec succès.

 Note

Seuls les modèles contenant des ressources peuvent être exportés.

9. Dans le flux de travail Create stack, sélectionnez Next.
10. Entrez un nom de pile, passez en revue les paramètres répertoriés, puis sélectionnez Next.

 Note

Le nom de la pile doit commencer par une lettre et ne contenir que des lettres, des chiffres et des tirets.

11. Sélectionnez Suivant après avoir fourni les informations suivantes :
 - Tags associés à la pile
 - Permissions cumulées
 - Les options de défaillance de la pile

 Note

Pour obtenir des conseils sur la gestion des piles, consultez les [AWS CloudFormation meilleures pratiques](#) du Guide de l'AWS CloudFormation utilisateur.

12. Vérifiez que les informations de votre pile sont correctes, vérifiez les accusés de réception en bas de page et sélectionnez le bouton Soumettre.

AWS CloudFormation commencera à créer la pile en fonction des données de votre modèle.

Mettre à jour une pile existante dans Infrastructure Composer en mode CloudFormation console

Suivez les instructions de cette rubrique pour mettre à jour une AWS CloudFormation pile existante.

Note

Si votre fichier est enregistré localement, nous vous recommandons d'utiliser [AWS Toolkit for Visual Studio Code](#).

1. Accédez à la [AWS CloudFormation console](#) et connectez-vous.
2. Sélectionnez la pile que vous souhaitez modifier.
3. Cliquez sur le bouton Mettre à jour. Vous serez alors redirigé vers l'assistant de mise à jour.
4. Sur la droite, sélectionnez Modifier dans Infrastructure Composer.
5. Sélectionnez le bouton ci-dessous intitulé Modifier dans Infrastructure Composer. Cela vous amènera à Infrastructure Composer en mode CloudFormation console.
6. Ici, vous pouvez glisser-déposer, configurer et connecter des ressources ([cartes](#)) à partir de la palette Ressources.

Note

Consultez [Comment composer](#) pour plus de détails sur l'utilisation d'Infrastructure Composer et notez que les cartes associées à Lambda (fonction Lambda et couche Lambda) nécessitent des versions de code et des solutions de packaging qui ne sont pas disponibles dans Infrastructure Composer en mode console. CloudFormation Ces cartes peuvent être utilisées dans la [console Infrastructure Composer](#) ou dans le AWS Toolkit for Visual Studio Code. Pour plus d'informations sur l'utilisation de ces outils, reportez-vous à [Où vous pouvez utiliser Infrastructure Composer](#).

7. Lorsque vous êtes prêt à exporter les modifications vers AWS CloudFormation, sélectionnez Mettre à jour le modèle.
8. Sélectionnez Confirmer et passez à CloudFormation. Cela vous ramènera au flux de travail Update Stack avec un message confirmant que votre modèle a été importé avec succès.

Note

Seuls les modèles contenant des ressources peuvent être exportés.

9. Dans le flux de travail Update Stack, sélectionnez Next.
10. Passez en revue tous les paramètres répertoriés et sélectionnez Suivant.

11. Sélectionnez Suivant après avoir fourni les informations suivantes :

- Tags associés à la pile
- Permissions cumulées
- Les options de défaillance de la pile

 Note

Pour obtenir des conseils sur la gestion des piles, consultez les [AWS CloudFormation meilleures pratiques](#) du Guide de l'AWS CloudFormation utilisateur.

12. Vérifiez que les informations de votre pile sont correctes, vérifiez les accusés de réception en bas de page et sélectionnez le bouton Soumettre.

AWS CloudFormation commencera à mettre à jour la pile en fonction des mises à jour que vous avez apportées à votre modèle.

À l'aide de Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code

Cette section décrit comment vous pouvez utiliser AWS Infrastructure Composer à partir du [AWS Toolkit for Visual Studio Code](#). Cela inclut un aperçu visuel d'Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code. Il comprend également des instructions indiquant comment accéder à cette expérience et synchroniser votre projet depuis VS Code vers le AWS cloud. Pour effectuer la synchronisation, utilisez la `sam sync` commande du AWS SAM CLI. Cette section fournit également des conseils sur l'utilisation Amazon Q alors que vous étiez dans Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code.

Pour obtenir des conseils supplémentaires sur l'utilisation d'Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code, reportez-vous à [Comment composer](#). Le contenu de cette section s'applique à cette expérience, ainsi qu'à l'expérience de la console Infrastructure Composer.

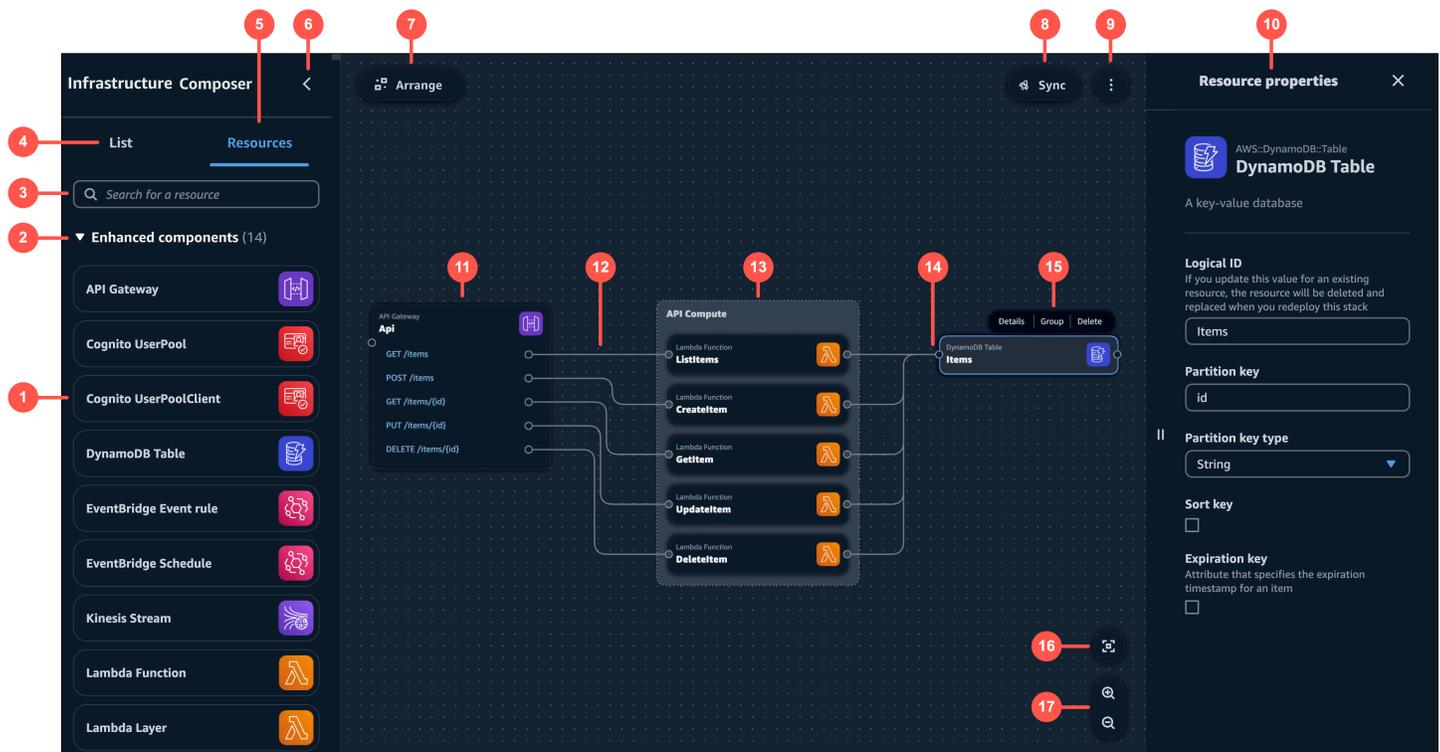
Rubriques

- [Présentation visuelle d'Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code](#)
- [Accédez à Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code](#)
- [Synchronisez Infrastructure Composer pour le déployer sur AWS Cloud](#)

- [Utilisation AWS Infrastructure Composer avec Amazon Q Developer](#)

Présentation visuelle d'Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code

Le concepteur visuel d'Infrastructure Composer AWS Toolkit for Visual Studio Code inclut un canevas visuel, qui inclut des composants numérotés dans l'image suivante et répertoriés ci-dessous.



1. Palette de ressources : affiche les cartes que vous pouvez utiliser pour créer.
2. Catégories de cartes — Les cartes sont organisées par catégories propres à Infrastructure Composer.
3. Barre de recherche de ressources : recherchez des cartes que vous pouvez ajouter au canevas.
4. Liste : affiche une arborescence des ressources de votre application.
5. Ressources : affiche la palette de ressources.
6. Bascule vers le volet gauche : masque ou affiche le volet gauche.
7. Organiser : organise l'architecture de votre application dans le canevas.
8. Sync — Lance le AWS Serverless Application Model (AWS SAM) CLI `aws sam sync` commande pour déployer votre application.

9. Menu — Fournit des options générales telles que les suivantes :

- Toile d'exportation
- Visitez la toile
- Liens vers la documentation
- Raccourcis clavier

10. Panneau des propriétés des ressources : affiche les propriétés pertinentes pour la carte sélectionnée dans le canevas. Ce panneau est dynamique. Les propriétés affichées changeront au fur et à mesure que vous configurerez votre carte.

11. Carte : affiche une vue de votre carte sur le canevas.

12. Ligne — Représente une connexion entre des cartes.

13. Groupe — Un groupe de cartes. Vous pouvez regrouper les cartes pour une organisation visuelle.

14. Port : points de connexion à d'autres cartes.

15. Actions de la carte : indique les actions que vous pouvez effectuer sur votre carte.

- Détails — Affiche le panneau des propriétés des ressources.
- Grouper — Regroupez les cartes sélectionnées.
- Supprimer — Supprime la carte de votre canevas et de votre modèle.

16. Recentrer : recentrez votre schéma d'application sur le canevas visuel.

17. Zoom — Zoomez en avant et en arrière sur votre toile.

Accédez à Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code

Suivez les instructions de cette rubrique pour accéder à Infrastructure Composer depuis le AWS Toolkit for Visual Studio Code.

Note

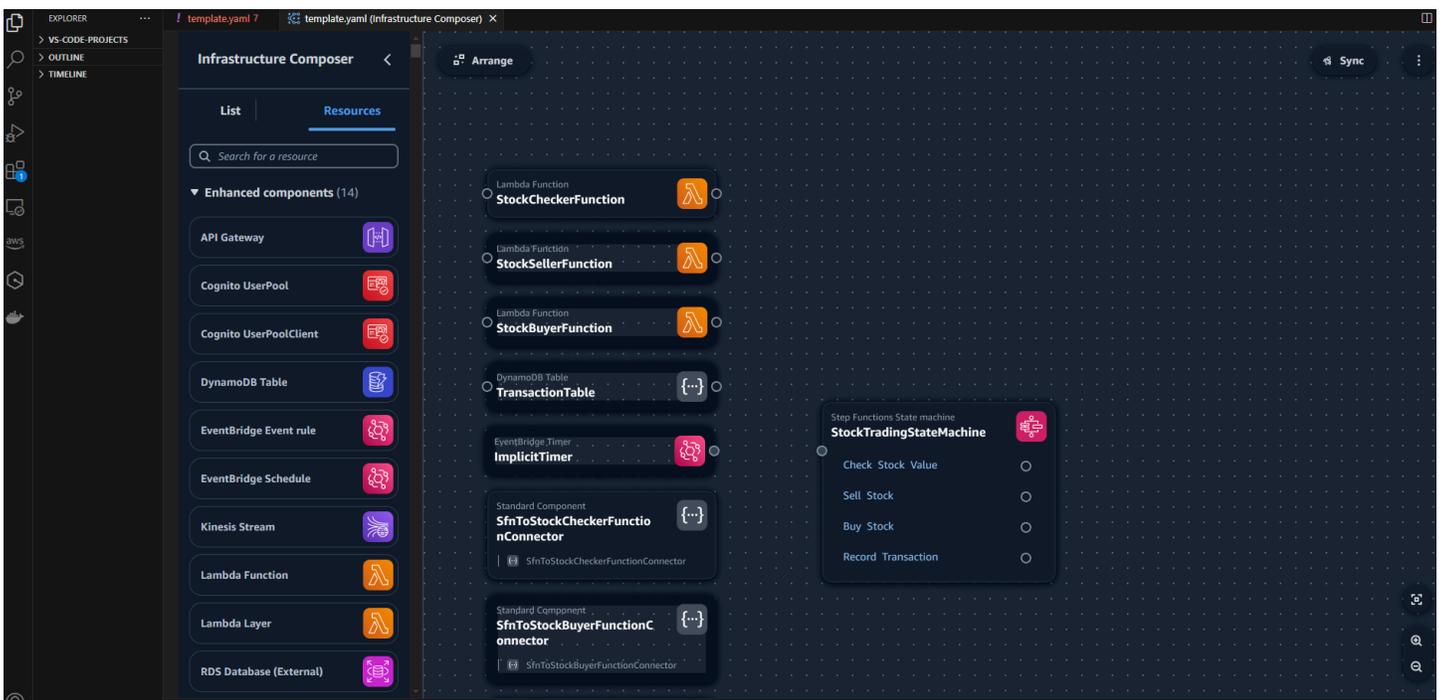
Avant de pouvoir accéder à Infrastructure Composer depuis le AWS Toolkit for Visual Studio Code, vous devez d'abord télécharger et installer le Toolkit for VS Code. Pour obtenir des instructions, consultez la section [Téléchargement du Toolkit for VS Code](#).

Pour accéder à Infrastructure Composer depuis le Toolkit for VS Code

Vous pouvez accéder à Infrastructure Composer de l'une des manières suivantes :

1. En sélectionnant le bouton Infrastructure Composer dans n'importe quel AWS CloudFormation AWS SAM modèle.
2. Dans le menu contextuel, cliquez avec le bouton droit de la souris sur votre AWS SAM modèle AWS CloudFormation ou sur votre modèle.
3. À partir de la palette de commandes VS Code.

Voici un exemple d'accès à Infrastructure Composer à partir du bouton Infrastructure Composer :



Pour plus d'informations sur l'accès à Infrastructure Composer, voir [Accès AWS Infrastructure Composer depuis le kit d'outils](#).

Synchronisez Infrastructure Composer pour le déployer sur AWS Cloud

Utilisez le bouton AWS Infrastructure Composer de synchronisation situé dans le AWS Toolkit for Visual Studio Code pour déployer votre application dans le AWS Cloud.

Le bouton de synchronisation lance la `sam sync` commande depuis l'interface de ligne de commande (CLI) de AWS SAM.

La `sam sync` commande peut déployer de nouvelles applications ou synchroniser rapidement les modifications que vous apportez localement au AWS Cloud. La course à pied `sam sync` peut inclure les éléments suivants :

- Création de votre application `sam build` pour préparer vos fichiers d'application locaux pour le déploiement en créant ou en mettant à jour un `.aws-sam` répertoire local.
- Pour les ressources qui soutiennent le AWS service APIs, le AWS SAM CLI utilisera le APIs pour déployer vos modifications. Le AWS SAM CLI fait cela pour mettre à jour rapidement vos ressources dans le cloud.
- Si nécessaire, le AWS SAM CLI effectue un AWS CloudFormation déploiement pour mettre à jour l'ensemble de votre stack par le biais d'un ensemble de modifications.

La `sam sync` commande convient parfaitement aux environnements de développement rapide, lorsque la mise à jour rapide de vos ressources cloud peut être bénéfique pour vos flux de travail de développement et de test.

Pour en savoir plus `sam sync`, consultez la section [Utilisation de sam sync](#) dans le manuel du AWS Serverless Application Model développeur.

Configuration

Pour utiliser la fonction de synchronisation dans Infrastructure Composer, vous devez disposer du AWS SAM CLI installé sur votre machine locale. Pour obtenir des instructions, reportez-vous à la section [Installation du AWS SAM CLI](#) dans le Manuel du développeur AWS Serverless Application Model .

Lorsque vous utilisez la fonction de synchronisation d'Infrastructure Composer, AWS SAM CLI fait référence à votre fichier de configuration pour obtenir les informations dont il a besoin pour synchroniser votre application avec le AWS Cloud. Pour obtenir des instructions sur la création, la modification et l'utilisation des fichiers de configuration, voir [Configurer les paramètres du projet](#) dans le Guide du AWS Serverless Application Model développeur.

Synchronisez et déployez votre application

Pour synchroniser votre application avec AWS Cloud

1. Sélectionnez le bouton de synchronisation sur le canevas d'Infrastructure Composer.

2. Il se peut que vous receviez un message vous demandant de confirmer que vous travaillez avec une pile de développement. Sélectionnez OK pour continuer.
3. Infrastructure Composer peut vous demander de configurer les options suivantes :
 - Région AWS— La région avec laquelle synchroniser votre application.
 - AWS CloudFormation nom de la pile : nom de votre AWS CloudFormation pile. Vous pouvez sélectionner un nom de pile existant ou en créer un nouveau.
 - Compartiment Amazon Simple Storage Service (Amazon S3) : nom de votre compartiment Amazon S3. Le AWS SAM CLI emballera et stockera vos fichiers d'application et votre code de fonction ici. Vous pouvez sélectionner un bucket existant ou en créer un nouveau.

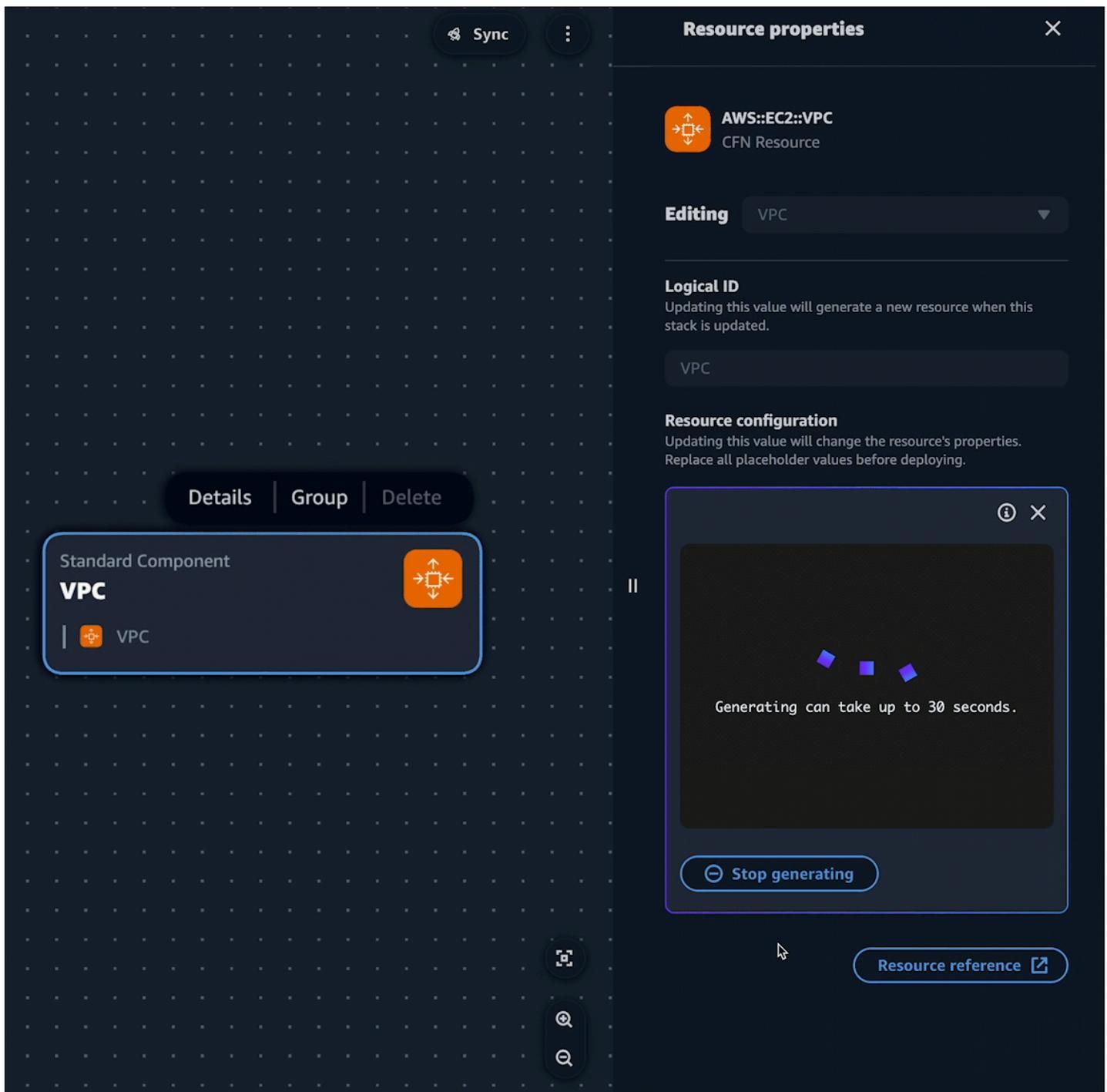
Le compositeur d'infrastructure lancera le AWS SAM CLI `sam sync` et ouvrira une fenêtre de terminal dans votre fenêtre IDE pour afficher sa progression.

Utilisation AWS Infrastructure Composer avec Amazon Q Developer

AWS Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code fournit une intégration avec Amazon Q. Vous pouvez utiliser Amazon Q dans Infrastructure Composer pour générer le code d'infrastructure pour vos AWS ressources lors de la conception de votre application.

Amazon Q est un générateur de code à usage général basé sur l'apprentissage automatique. Pour en savoir plus, consultez [Qu'est-ce que Amazon Q?](#) dans le Amazon Q Developer Guide de l'utilisateur.

Pour les cartes de ressources et de composants standard, vous pouvez utiliser Amazon Q pour générer des suggestions de code d'infrastructure pour vos ressources.



Les cartes de ressources et de composants standard peuvent représenter une AWS CloudFormation ressource ou un ensemble de AWS CloudFormation ressources. Pour en savoir plus, consultez [Configurer et modifier des cartes dans Infrastructure Composer](#).

Configuration

Pour utiliser Amazon Q dans Infrastructure Composer, vous devez vous authentifier auprès Amazon Q dans le kit d'outils. Pour obtenir des instructions, consultez [Getting started with Amazon Q dans VS Code et JetBrains](#) dans le Amazon Q Developer Guide de l'utilisateur.

Utilisation Amazon Q Developer dans Infrastructure Composer

Vous pouvez utiliser ... Amazon Q Developer depuis le panneau des propriétés des ressources de toute ressource standard ou carte de composant standard.

Pour utiliser Amazon Q dans Infrastructure Composer

1. À partir d'une ressource standard ou d'une carte de composant standard, ouvrez le panneau des propriétés de la ressource.
2. Localisez le champ Configuration des ressources. Ce champ contient le code d'infrastructure de la carte.
3. Cliquez sur le bouton Générer des suggestions. Amazon Q générera une suggestion.

Note

Le code généré à ce stade ne remplacera pas le code d'infrastructure existant de votre modèle.

4. Pour générer d'autres suggestions, sélectionnez Régénérer. Vous pouvez passer d'un échantillon à l'autre pour comparer les résultats.
5. Pour sélectionner une option, choisissez Sélectionner. Vous pouvez modifier le code ici avant de l'enregistrer dans votre application. Pour quitter sans enregistrer, sélectionnez l'icône de sortie (X).
6. Pour enregistrer le code dans votre modèle d'application, sélectionnez Enregistrer dans le panneau des propriétés des ressources.

En savoir plus

En savoir plus sur Amazon Q, voir [Qu'est-ce que Amazon Q?](#) dans le Amazon Q Developer Guide de l'utilisateur.

Comment composer AWS Infrastructure Composer

Cette section décrit les principes de base de l'utilisation d'Infrastructure Composer à partir du [Console Infrastructure Composer CloudFormation mode console](#), et du [AWS Toolkit for Visual Studio Code](#). Plus précisément, les rubriques de cette section fournissent des informations clés sur la façon de composer une application avec Infrastructure Composer, ainsi que des détails sur les fonctionnalités et les raccourcis supplémentaires. Il existe quelques variantes de fonctionnalités entre les expériences console et VS Code, et les rubriques de cette section identifient et décrivent ces variations là où elles se produisent.

Après avoir rédigé votre demande, vous serez prêt à consulter les informations [Déployez votre application sans serveur Infrastructure Composer dans le cloud AWS](#) relatives au déploiement de votre application.

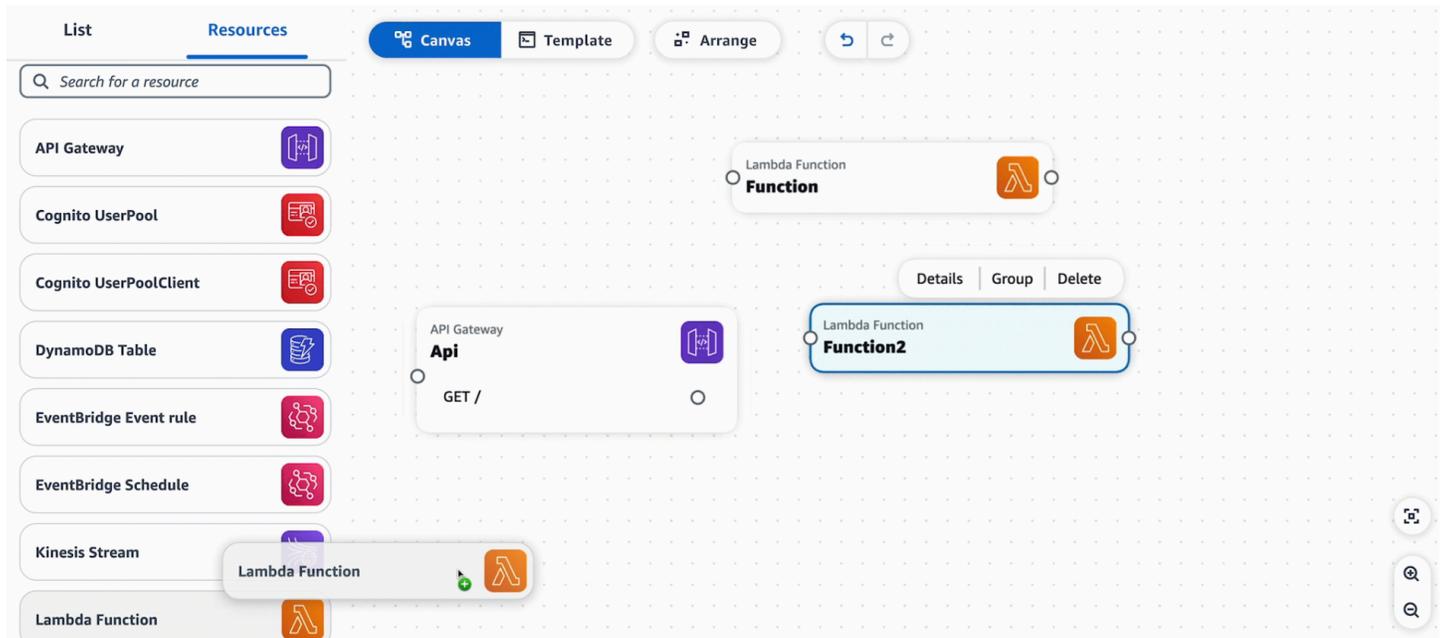
Rubriques

- [Placez des cartes sur le canevas visuel d'Infrastructure Composer](#)
- [Regroupez les cartes sur le canevas visuel d'Infrastructure Composer](#)
- [Connectez les cartes sur le canevas visuel d'Infrastructure Composer](#)
- [Déconnecter les cartes dans Infrastructure Composer](#)
- [Disposez les cartes sur le canevas visuel d'Infrastructure Composer](#)
- [Configurer et modifier des cartes dans Infrastructure Composer](#)
- [Supprimer des cartes dans Infrastructure Composer](#)
- [Afficher les mises à jour du code avec le Change Inspector dans Infrastructure Composer](#)
- [Référencer des fichiers externes dans Infrastructure Composer](#)
- [Intégrer Infrastructure Composer à Amazon Virtual Private Cloud \(AmazonVPC\)](#)

Placez des cartes sur le canevas visuel d'Infrastructure Composer

Cette section décrit comment sélectionner et faire glisser les [cartes](#) Infrastructure Composer dans son canevas visuel. Avant de commencer, identifiez les ressources dont votre application a besoin et la manière dont elles doivent interagir. Pour obtenir des conseils à ce sujet, consultez [Créez votre première application avec Infrastructure Composer](#).

Pour ajouter une carte à votre application, faites-la glisser depuis la palette de ressources et déposez-la sur le canevas visuel.



Vous pouvez choisir entre deux types de cartes : les cartes de [composants améliorées](#) et les cartes de [ressources laC standard](#).

Après avoir placé vos cartes sur le canevas visuel, vous serez prêt à les regrouper, à les connecter, à les organiser et à les configurer. Consultez les rubriques suivantes pour plus d'informations à ce sujet :

- [Regroupez les cartes sur le canevas visuel d'Infrastructure Composer](#)
- [Connectez les cartes sur le canevas visuel d'Infrastructure Composer](#)
- [Disposez les cartes sur le canevas visuel d'Infrastructure Composer](#)
- [Configurer et modifier des cartes dans Infrastructure Composer](#)

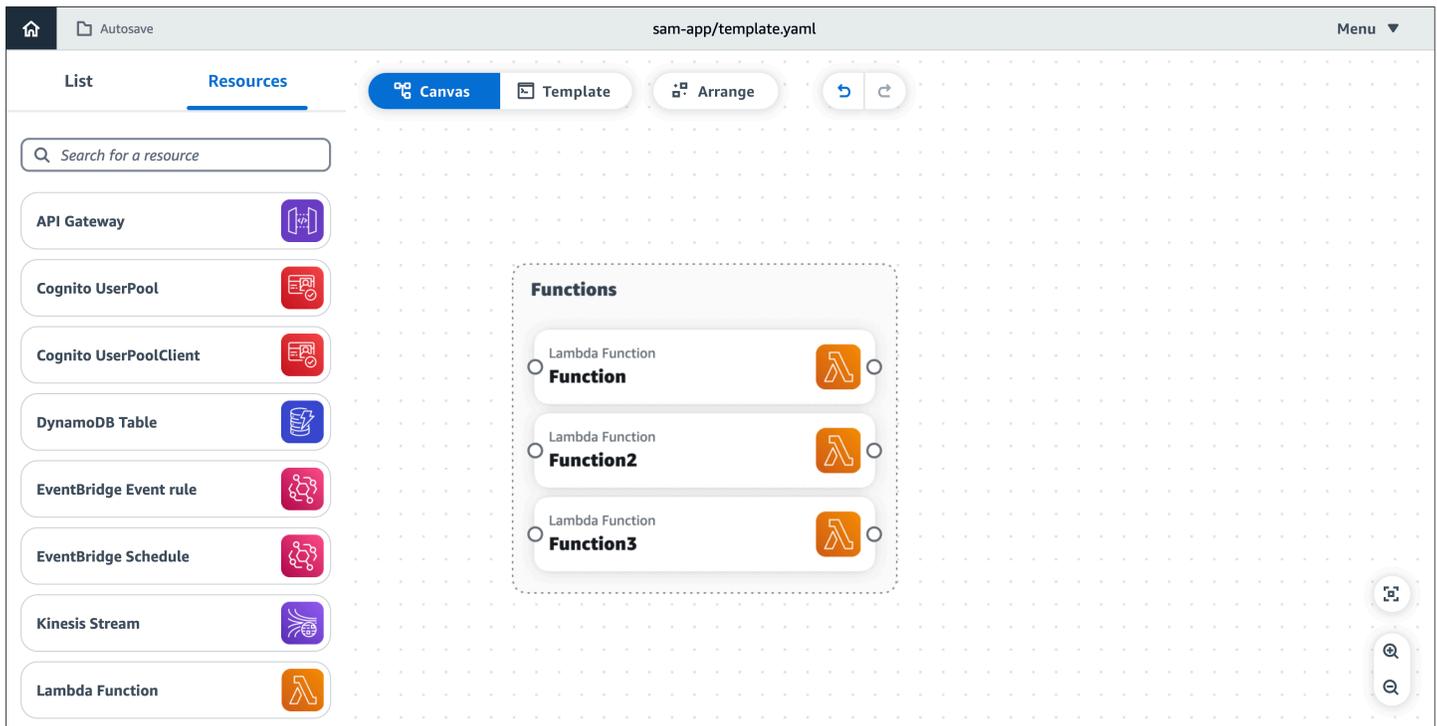
Regroupez les cartes sur le canevas visuel d'Infrastructure Composer

Cette rubrique contient des informations sur le regroupement des cartes de composants améliorées et des cartes de composants standard. Les cartes de regroupement vous permettent de classer et d'organiser vos ressources sans avoir à réfléchir au code ou au balisage que vous devez écrire.

Regroupement de cartes de composants améliorées

Il existe deux manières de regrouper les cartes de composants améliorées :

- Tout en appuyant sur Shift, sélectionnez les cartes à regrouper. Choisissez ensuite Grouper dans le menu des actions sur les ressources.
- sélectionnez la carte que vous souhaitez inclure dans un groupe. Dans le menu qui apparaît, sélectionnez Grouper. Cela créera un groupe dans lequel vous pourrez glisser-déposer d'autres cartes.



Regroupement d'une carte de composant standard dans une autre

L'exemple suivant montre comment une carte de composant standard peut être regroupée dans une autre carte à partir du panneau des propriétés des ressources :

The image shows the AWS Infrastructure Composer interface. On the left is a canvas with a grid pattern. A 'Standard Component' card for 'Function' is placed on the canvas. This card has an orange Lambda icon and two sub-components: 'Role' (with a red icon) and 'Function' (with an orange icon). Above the card are buttons for 'Details', 'Group', and 'Delete'. On the right is the 'Resource properties' panel, titled 'Resource properties' with a close button. It shows the resource 'AWS::Lambda::Function' (CFN Resource) with the Lambda icon. Below this are two dropdown menus: 'Editing' (set to 'Function') and 'Logical ID' (set to 'Function' with a checkmark). A note below the 'Logical ID' dropdown states: 'Updating this value will generate a new resource when this stack is updated.' Below the dropdowns is a text input field containing 'Function'. Underneath is the 'Resource configuration' section, with a note: 'Updating this value will change the resource's properties. Replace all placeholder values before deploying.' Below this is a code editor with the following content:

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

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

Dans le champ Configuration des ressources du panneau Propriétés des ressources, le `Role` a été référencé dans la fonction Lambda. Cela entraîne le regroupement de la carte Rôle dans la carte Fonction sur le canevas.

Connectez les cartes sur le canevas visuel d'Infrastructure Composer

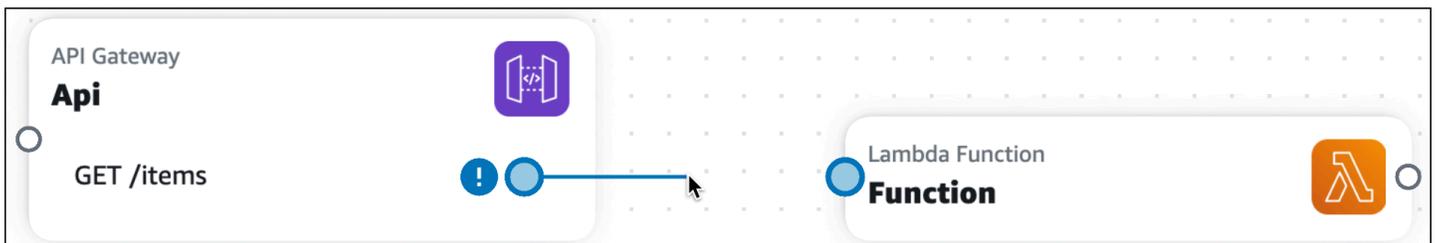
Utilisez cette rubrique pour comprendre comment connecter des cartes dans Infrastructure Composer. Cette section contient des informations sur la connexion de cartes de composants améliorées et de cartes de composants standard. Il fournit également quelques exemples illustrant les différentes manières dont les cartes peuvent être connectées.

Connexion de cartes de composants améliorées

Sur les cartes de composants améliorées, les ports identifient visuellement les endroits où les connexions peuvent être établies.

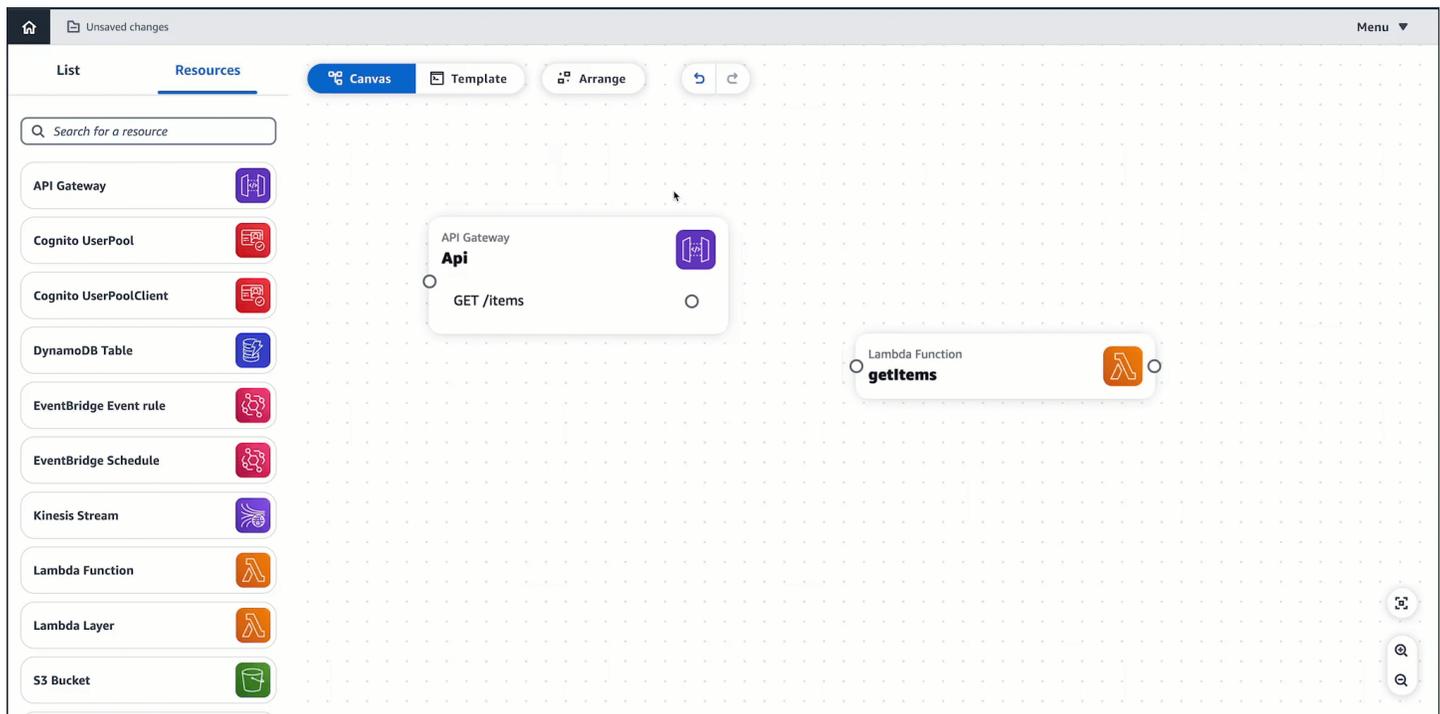
- Un port situé sur le côté droit d'une carte indique que la carte a la possibilité d'invoquer une autre carte.
- Un port situé sur le côté gauche d'une carte indique qu'il est possible que la carte soit invoquée par une autre carte.

Connectez les cartes entre elles en cliquant sur le port droit d'une carte et en le faisant glisser sur le port gauche d'une autre carte.



Lorsque vous créez une connexion, un message s'affiche pour vous indiquer si la connexion a été établie avec succès. Sélectionnez le message pour voir ce qu'Infrastructure Composer a modifié pour configurer une connexion. Si la connexion a échoué, vous pouvez sélectionner la vue Modèle pour mettre à jour manuellement votre code d'infrastructure afin de configurer la connexion.

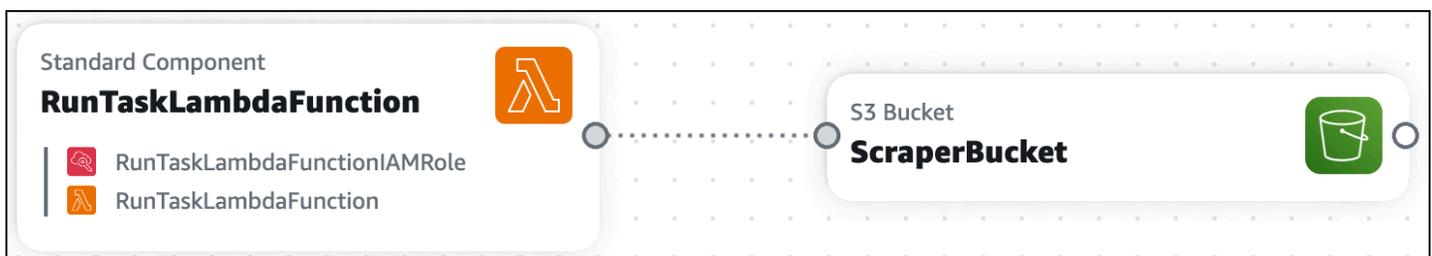
- En cas de succès, cliquez sur le message pour afficher l'inspecteur des modifications. Vous pouvez voir ici ce qu'Infrastructure Composer a modifié pour configurer votre connexion.
- En cas d'échec, un message s'affiche. Vous pouvez sélectionner la vue Modèle et mettre à jour manuellement votre code d'infrastructure pour configurer la connexion.



Lorsque vous connectez des cartes de composants améliorées, Infrastructure Composer crée automatiquement le code d'infrastructure dans votre modèle pour établir la relation événementielle entre vos ressources.

Connexion de cartes de composants standard (cartes de ressources IaC standard)

Les cartes de ressources IaC standard n'incluent pas de ports permettant de créer des connexions avec d'autres ressources. Lors de [la configuration des cartes](#), vous spécifiez les relations basées sur les événements dans le modèle de votre application. Infrastructure Composer détectera automatiquement ces connexions et les visualisera avec une ligne en pointillés entre vos cartes. Voici un exemple de connexion entre une carte de composant standard et une carte de composant améliorée :



L'exemple suivant montre comment une fonction Lambda peut être connectée à un Amazon API Gateway Rest : API

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

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

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

  LambdaExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
```

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

Dans l'exemple ci-dessus, l'extrait de code répertorié ci-dessous `Integration:` indique la relation événementielle qui relie les deux cartes. `ApiGatewayMethod:`

Exemples de connexion de cartes dans Infrastructure Composer

Utilisez les exemples de cette section pour comprendre comment les cartes peuvent être connectées dans Infrastructure Composer.

Invoquer une AWS Lambda fonction lorsqu'un article est placé dans un compartiment Amazon Simple Storage Service (Amazon S3)

Dans cet exemple, une carte bucket Amazon S3 est connectée à une carte fonctionnelle Lambda. Lorsqu'un élément est placé dans le compartiment Amazon S3, la fonction est invoquée. La fonction peut ensuite être utilisée pour traiter l'élément ou déclencher d'autres événements dans votre application.



Cette interaction nécessite la définition d'un événement pour la fonction. Voici ce que propose 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

```

Invoquer un compartiment Amazon S3 à partir d'une fonction Lambda

Dans cet exemple, une carte de fonction Lambda invoque une carte de compartiment Amazon S3. La fonction Lambda peut être utilisée pour effectuer des CRUD opérations sur des éléments du compartiment Amazon S3.



Cette interaction nécessite les éléments suivants, fournis par Infrastructure Composer :

- IAM politiques qui permettent à la fonction Lambda d'interagir avec le compartiment Amazon S3.
- Variables d'environnement qui influencent le comportement de la fonction Lambda.

```

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

```

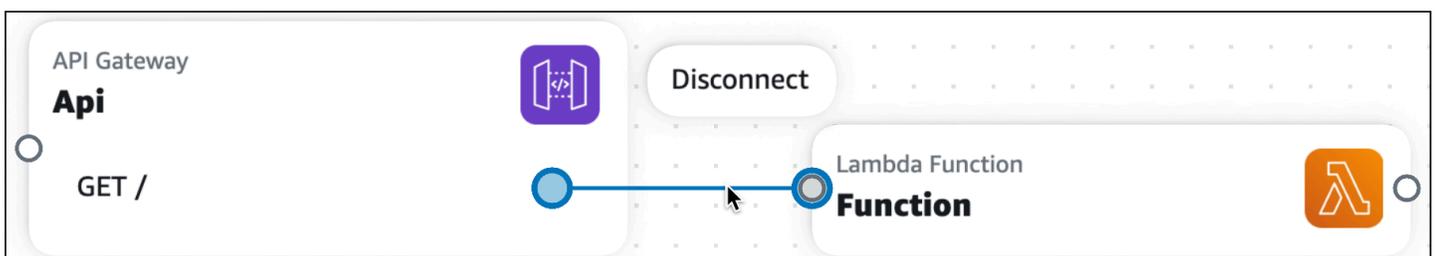
```
...
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    ...
  MyBucketBucketPolicy:
    Type: AWS::S3::BucketPolicy
    ...
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Environment:
        Variables:
          BUCKET_NAME: !Ref MyBucket
          BUCKET_ARN: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

Déconnecter les cartes dans Infrastructure Composer

Dans Infrastructure Composer, vous connectez et déconnectez AWS des ressources à l'aide de cartes de composants améliorées et de cartes de composants standard. Cette section décrit comment déconnecter les deux types de cartes.

Cartes de composants améliorées

Pour déconnecter les cartes de composants améliorées, sélectionnez la ligne et choisissez **Déconnecter**.



Infrastructure Composer modifiera automatiquement votre modèle pour supprimer la relation basée sur les événements de votre application.

Cartes à composants standard

Les cartes de composants standard n'incluent pas de ports permettant de créer des connexions avec d'autres ressources. Lors de [la configuration des cartes](#), vous spécifiez les relations basées sur les événements dans le modèle de votre application. Infrastructure Composer détectera automatiquement ces connexions et les visualisera avec une ligne en pointillés entre vos cartes. Pour déconnecter une carte de composant standard, supprimez la relation basée sur les événements dans le modèle de votre application.

L'exemple suivant montre une fonction Lambda connectée à un serveur Rest 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
```

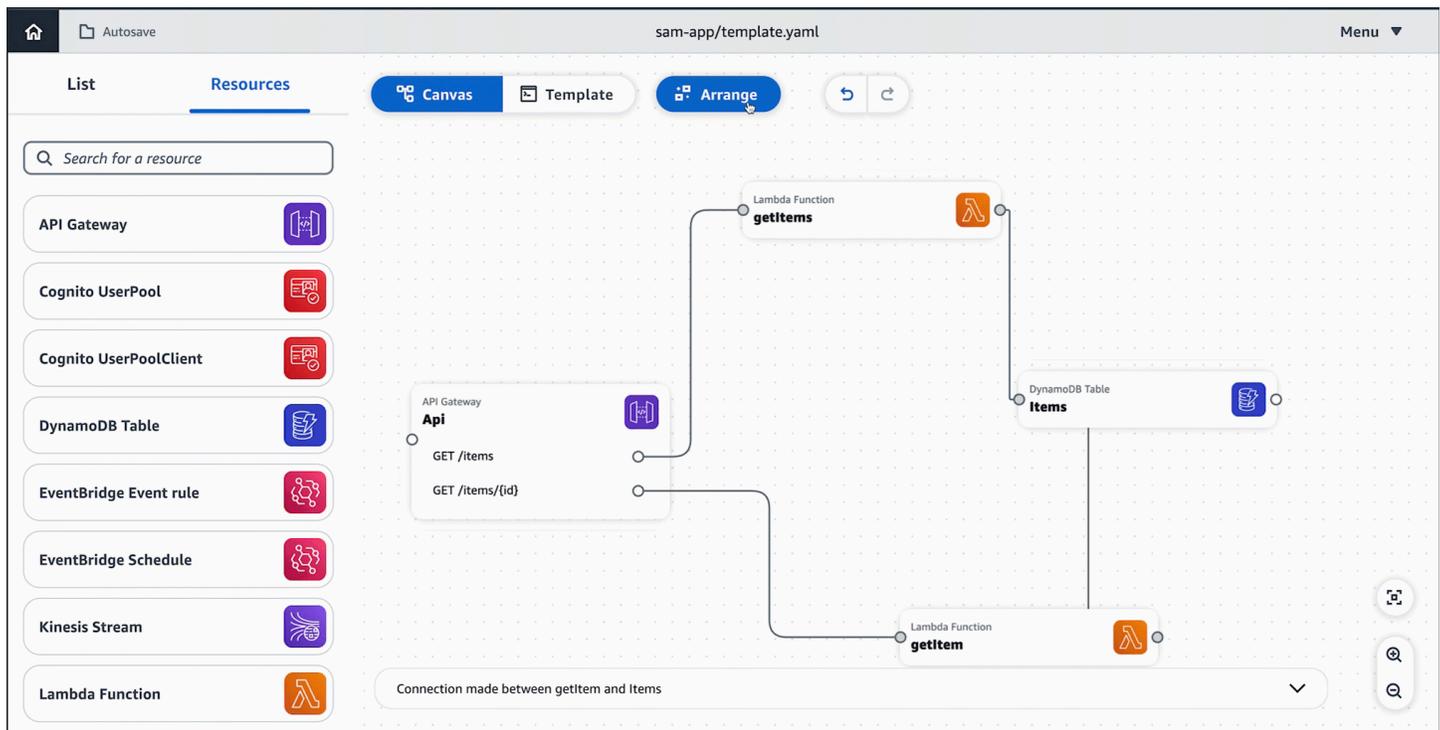
```
Runtime: nodejs14.x
Code:
  S3Bucket: your-bucket-name
  S3Key: your-lambda-zip-file.zip

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
```

Pour supprimer la connexion entre les deux cartes, supprimez les références `MyLambdaFunction` répertoriées ci-dessous `ApiGatewayMethod:. Integration`

Disposez les cartes sur le canevas visuel d'Infrastructure Composer

Sélectionnez `Disposition` pour disposer et organiser visuellement les cartes sur le canevas. L'utilisation du bouton `Arrangement` est particulièrement utile lorsqu'il y a de nombreuses cartes et connexions sur le canevas.



Configurer et modifier des cartes dans Infrastructure Composer

Dans Infrastructure Composer, les cartes représentent les ressources que vous utilisez pour concevoir l'architecture de votre application. Lorsque vous configurez une carte dans Infrastructure Composer, vous définissez les détails des ressources de votre application. Cela inclut des détails tels que l'identifiant logique et la clé de partition d'une carte. La façon dont ces informations sont définies varie entre les cartes à composants améliorées et les cartes standard.

Une carte de composants améliorée est un ensemble de AWS CloudFormation ressources qui ont été combinées dans une seule carte organisée qui améliore la facilité d'utilisation et les fonctionnalités et est conçue pour une grande variété de cas d'utilisation. Une carte de ressources laC standard représente une AWS CloudFormation ressource unique. Chaque carte de ressources laC standard, une fois glissée sur le canevas, est étiquetée Composant standard.

Cette rubrique fournit des informations détaillées sur la configuration des cartes de composants améliorées et des cartes de composants standard.

Note

Cette rubrique concerne l'utilisation de cartes depuis la console Infrastructure Composer, l'AWS Toolkit for Visual Studio Code extension et en mode CloudFormation console dans

Infrastructure Composer. Les cartes associées à Lambda (fonction Lambda et couche Lambda) nécessitent des compilations de code et des solutions de packaging qui ne sont pas disponibles dans Infrastructure Composer en mode console. CloudFormation Pour de plus amples informations, veuillez consulter [Utilisation d'Infrastructure Composer en mode CloudFormation console](#).

Rubriques

- [Cartes de composants améliorées dans Infrastructure Composer](#)
- [Cartes standard dans Infrastructure Composer](#)

Cartes de composants améliorées dans Infrastructure Composer

Pour configurer des cartes de composants améliorées, Infrastructure Composer fournit un formulaire dans le panneau des propriétés des ressources. Ce formulaire a été spécialement conçu pour vous guider dans la configuration de chaque carte de composant améliorée. Au fur et à mesure que vous remplissez le formulaire, Infrastructure Composer modifie votre code d'infrastructure.

Certaines cartes de composants améliorées comportent des fonctionnalités supplémentaires. Cette section passe en revue les principes de base de l'utilisation des cartes de composants améliorées et fournit des détails sur les cartes dotées de fonctionnalités supplémentaires.

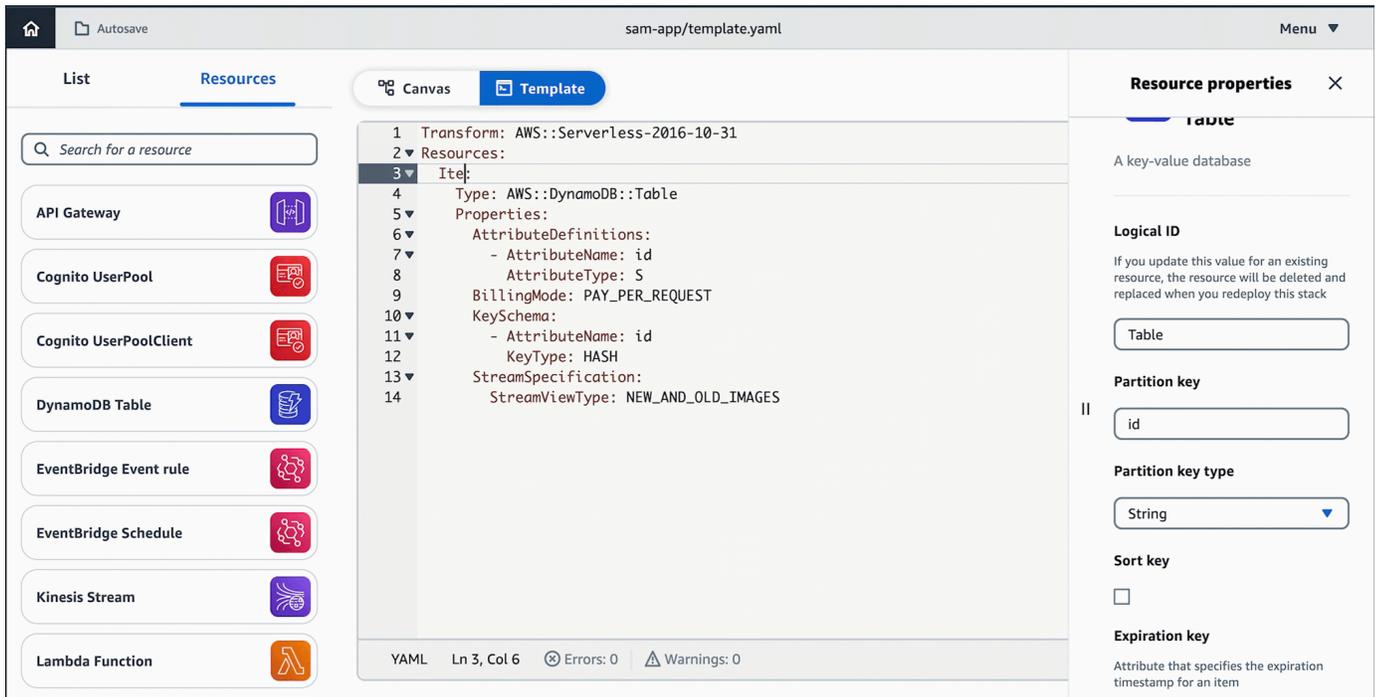
Pour plus d'informations sur les cartes de composants améliorées, consultez [Cartes de composants améliorées dans Infrastructure Composer](#) et [Cartes de composants améliorées dans Infrastructure Composer](#)

Procédure

Le panneau des propriétés des ressources rationalise la configuration et ajoute des guides qui simplifient la configuration des cartes. Pour utiliser ce panneau, effectuez les opérations suivantes :

1. Double-cliquez sur une carte pour afficher le panneau des propriétés des ressources.
2. Cliquez sur une carte et sélectionnez Détails pour afficher le panneau des propriétés des ressources.
3. Pour Infrastructure Composer à partir du AWS Management Console, sélectionnez Modèle pour afficher le code de votre application. Configurez directement à partir d'ici.

L'image suivante montre comment cela peut être fait :



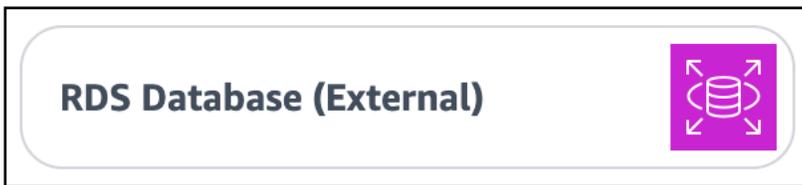
Utilisation d'Infrastructure Composer avec Amazon Relational Database Service (AmazonRDS)

AWS Infrastructure Composer propose une intégration avec Amazon Relational Database Service (RDSAmazon). À l'aide de la carte de composants améliorée de RDSbase de données (externe) d'Infrastructure Composer, vous pouvez connecter votre application à Amazon RDS DB clusters, instances et proxys définis sur un autre modèle AWS CloudFormation or AWS Serverless Application Model (AWS SAM).

La carte des composants améliorés de la RDSbase de données (externe) représente les RDS ressources Amazon définies sur un autre modèle. Cela consiste notamment à :

- Amazon RDS DB cluster ou instance défini sur un autre modèle
- Amazon RDS DB proxy

La carte de composant améliorée de la RDSbase de données (externe) est disponible dans la palette Ressources.



Pour utiliser cette carte, faites-la glisser sur le canevas Infrastructure Composer, configurez-la et connectez-la à d'autres ressources.

Vous pouvez connecter votre application à l'Amazon externe RDS DB cluster ou instance via une fonction Lambda.

Prérequis

Pour utiliser cette fonctionnalité, vous devez satisfaire aux exigences suivantes :

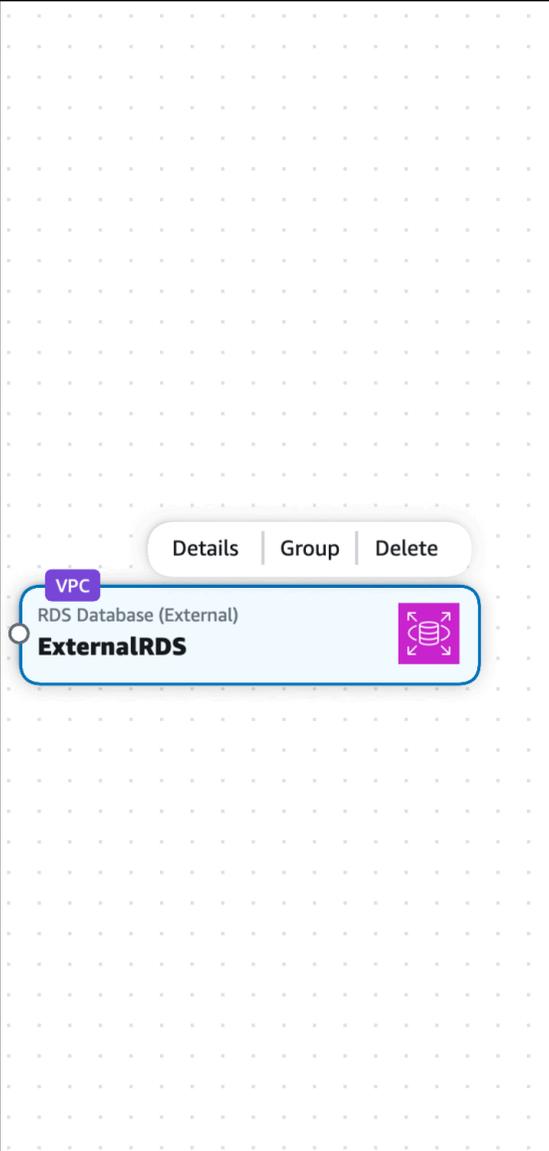
1. Votre Amazon externe RDS DB le cluster, l'instance ou le proxy doivent être utilisés AWS Secrets Manager pour gérer le mot de passe utilisateur. Pour en savoir plus, consultez la section [Gestion des mots de passe avec Amazon RDS et AWS Secrets Manager](#) le guide de RDS l'utilisateur Amazon.
2. Votre application dans Infrastructure Composer doit être un nouveau projet ou doit avoir été créée à l'origine dans Infrastructure Composer.

Procédure

Étape 1 : Configuration de la carte de RDS base de données externe

Dans la palette Ressources, faites glisser une carte de composant amélioré RDSde base de données (externe) sur le canevas.

Sélectionnez la carte et choisissez Détails ou double-cliquez sur la carte pour afficher le panneau des propriétés des ressources. Le panneau des propriétés des ressources de la carte apparaîtra :



The screenshot shows a card titled 'RDS Database (External)' with a 'VPC' tag and the name 'ExternalRDS'. The card is on a grid background and has three buttons: 'Details', 'Group', and 'Delete'.

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}.

Vous pouvez configurer les éléments suivants ici :

- ID logique : nom unique pour votre Amazon externe RDS DB cluster, instance ou proxy. Cet identifiant ne doit pas nécessairement correspondre à la valeur d'identifiant logique de votre Amazon externe RDS DB ressource.
- Secret de base de données : identifiant du AWS Secrets Manager secret associé à votre Amazon RDS DB cluster, instance ou proxy. Ce champ accepte les valeurs suivantes :
 - Valeur statique : identifiant unique du secret de base de données, tel que le secretARN. Voici un exemple : `arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-name-1a2b3c` Pour plus d'informations, consultez [Concepts AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

- Valeur de sortie : lorsqu'un secret de Secrets Manager est déployé AWS CloudFormation, une valeur de sortie est créée. Vous pouvez spécifier la valeur de sortie ici à l'aide de la fonction [Fn::ImportValue](#) intrinsèque. Par exemple, `!ImportValue MySecret`.
- Valeur du magasin de SSM paramètres : vous pouvez stocker votre secret dans le magasin de SSM paramètres et spécifier sa valeur à l'aide d'une référence dynamique. Par exemple, `{{resolve:ssm:MySecret}}`. Pour plus d'informations, consultez [SSM les paramètres](#) dans le guide de AWS CloudFormation l'utilisateur.
- Nom d'hôte de la base de données — Le nom d'hôte qui peut être utilisé pour se connecter à votre Amazon RDS DB cluster, instance ou proxy. Cette valeur est spécifiée dans le modèle externe qui définit votre RDS ressource Amazon. Les valeurs suivantes sont acceptées :
 - Valeur statique : identifiant unique du nom d'hôte de la base de données, tel que l'adresse du point de terminaison. Voici un exemple : `mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-2.rds.amazonaws.com`
 - Valeur de sortie : valeur de sortie d'un Amazon déployé RDS DB cluster, instance ou proxy. Vous pouvez spécifier la valeur de sortie à l'aide de la fonction [Fn::ImportValue](#) intrinsèque. Par exemple, `!ImportValue myStack-myDatabase-abcd1234`.
 - Valeur du magasin de SSM paramètres : vous pouvez stocker le nom d'hôte de la base de données dans le magasin de SSM paramètres et spécifier sa valeur à l'aide d'une référence dynamique. Par exemple, `{{resolve:ssm:MyDatabase}}`.
- Port de base de données — Le numéro de port qui peut être utilisé pour se connecter à votre Amazon RDS DB cluster, instance ou proxy. Cette valeur est spécifiée dans le modèle externe qui définit votre RDS ressource Amazon. Les valeurs suivantes sont acceptées :
 - Valeur statique : port de base de données. Par exemple, `3306`.
 - Valeur de sortie : valeur de sortie d'un Amazon déployé RDS DB cluster, instance ou proxy. Par exemple, `!ImportValue myStack-MyRDSInstancePort`.
 - Valeur provenant du magasin de SSM paramètres : vous pouvez stocker le nom d'hôte de la base de données dans le magasin de SSM paramètres et spécifier sa valeur à l'aide d'une référence dynamique. Par exemple, `{{resolve:ssm:MyRDSInstancePort}}`.

 Note

Seule la valeur d'ID logique doit être configurée ici. Vous pouvez configurer les autres propriétés au moment du déploiement si vous le souhaitez.

Étape 2 : Connecter une carte de fonction Lambda

Dans la palette Ressources, faites glisser une carte de composant améliorée de la fonction Lambda sur le canevas.

Connectez le port gauche de la carte Lambda Function au port droit de la carte de RDSbase de données (externe).



Infrastructure Composer fournira votre modèle pour faciliter cette connexion.

Ce que fait Infrastructure Composer pour créer votre connexion

Lorsque vous avez terminé la procédure décrite ci-dessus, Infrastructure Composer exécute des actions spécifiques pour connecter votre fonction Lambda à votre base de données.

Lors de la spécification de l'Amazon externe RDS DB cluster, instance ou proxy

Lorsque vous faites glisser une carte RDSde base de données (externe) sur le canevas, Infrastructure Composer met à jour les Parameters sections Metadata et de votre modèle selon les besoins. Voici un exemple :

```
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
```

[Les métadonnées](#) sont une section de AWS CloudFormation modèle utilisée pour stocker les informations relatives à votre modèle. Les métadonnées spécifiques à Infrastructure Composer sont

stockées sous la clé de `AWS::Composer::ExternalResources` métadonnées. Infrastructure Composer stocke ici les valeurs que vous spécifiez pour votre Amazon RDS DB cluster, instance ou proxy.

La section [Paramètres](#) d'un AWS CloudFormation modèle est utilisée pour stocker des valeurs personnalisées qui peuvent être insérées dans l'ensemble de votre modèle lors du déploiement. Selon le type de valeurs que vous fournissez, Infrastructure Composer peut stocker des valeurs ici pour votre Amazon RDS DB cluster, instance ou proxy et spécifiez-les dans l'ensemble de votre modèle.

Les valeurs de chaîne de la `Parameters` section `Metadata` et utilisent la valeur d'ID logique que vous spécifiez sur votre carte RDS de base de données (externe). Si vous mettez à jour l'ID logique, les valeurs des chaînes changeront.

Lors de la connexion de la fonction Lambda à votre base de données

Lorsque vous connectez une carte de fonction Lambda à la carte de RDS de base de données (externe), Infrastructure Composer fournit des variables d'environnement et AWS Identity and Access Management (IAM) des politiques. Voici un exemple :

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

Les variables d'[environnement](#) sont des variables qui peuvent être utilisées par votre fonction au moment de l'exécution. Pour en savoir plus, consultez la section [Utilisation des variables d'environnement Lambda](#) dans le Guide du AWS Lambda développeur.

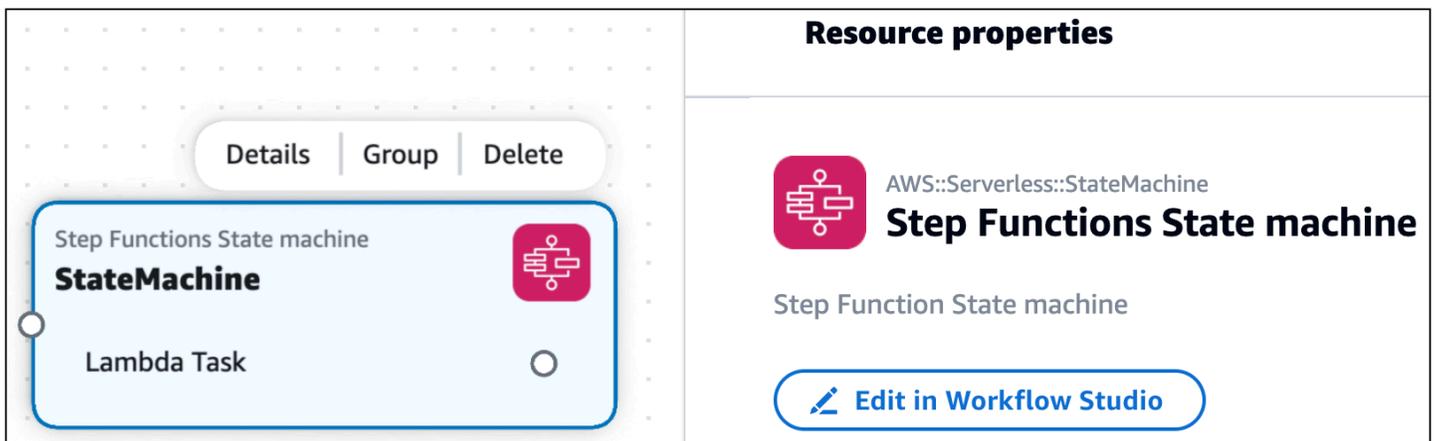
[Les politiques](#) fournissent des autorisations pour votre fonction. Infrastructure Composer crée ici une politique pour autoriser l'accès en lecture depuis votre fonction à Secrets Manager afin d'obtenir votre mot de passe pour accéder à Amazon RDS DB cluster, instance ou proxy.

Utilisation AWS Infrastructure Composer avec AWS Step Functions

AWS Infrastructure Composer propose une intégration avec [AWS Step Functions Workflow Studio](#). Utilisez Infrastructure Composer pour effectuer les opérations suivantes :

- Launch Step Functions Workflow Studio directement dans Infrastructure Composer.
- Créez et gérez de nouveaux flux de travail ou importez des flux de travail existants dans Infrastructure Composer.
- Intégrez vos flux de travail à d'autres AWS ressources à l'aide du canevas Infrastructure Composer.

L'image suivante représente une carte machine Step Functions State



Avec Step Functions Workflow Studio dans Infrastructure Composer, vous pouvez utiliser les avantages de deux puissants concepteurs visuels en un seul endroit. Lorsque vous concevez votre flux de travail et votre application, Infrastructure Composer crée votre infrastructure sous forme de code (IaC) pour vous guider vers le déploiement.

Rubriques

- [Stratégies IAM](#)
- [Commencer à utiliser Step Functions Workflow Studio dans Infrastructure Composer](#)
- [Utilisation de Step Functions Workflow Studio dans Infrastructure Composer](#)
- [En savoir plus](#)

Stratégies IAM

Lorsque vous connectez des tâches de votre flux de travail à des ressources, Infrastructure Composer crée automatiquement les AWS Identity and Access Management (IAM) politiques requises pour autoriser l'interaction entre vos ressources. Voici un exemple :

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

Si nécessaire, vous pouvez ajouter d'autres IAM politiques à votre modèle.

Commencer à utiliser Step Functions Workflow Studio dans Infrastructure Composer

Pour commencer, vous pouvez créer de nouveaux flux de travail ou importer des flux de travail existants.

Pour créer un nouveau flux de travail

1. Depuis la palette Resources, faites glisser une carte de composant améliorée par Step Functions State Machine sur le canevas.



Lorsque vous faites glisser une carte de machine Step Functions State sur le canevas, Infrastructure Composer crée ce qui suit :

- Une [AWS::Serverless::StateMachine](#) ressource qui définit votre machine à états. Par défaut, Infrastructure Composer crée un flux de travail standard. Pour créer un flux de travail express, remplacez la Type valeur de votre modèle par STANDARDEXPRESS.
 - Une [AWS::Logs::LogGroup](#) ressource qui définit un groupe de CloudWatch journaux Amazon pour votre machine d'état.
2. Ouvrez le panneau des propriétés des ressources de la carte et sélectionnez Modifier dans Workflow Studio pour ouvrir Workflow Studio dans Infrastructure Composer.

Step Functions Workflow Studio s'ouvre en mode Design. Pour en savoir plus, consultez la section [Mode conception](#) dans le guide du AWS Step Functions développeur.

 Note

Vous pouvez modifier Infrastructure Composer pour enregistrer la définition de votre machine à états dans un fichier externe. Pour en savoir plus, consultez [Utilisation de fichiers externes](#).

3. Créez votre flux de travail et choisissez Enregistrer. Pour sortir Workflow Studio, choisissez Return to Infrastructure Composer.

Infrastructure Composer définit votre flux de travail à l'aide de la `Definition` propriété de la `AWS::Serverless::StateMachine` ressource.

4. Vous pouvez modifier votre flux de travail en effectuant l'une des opérations suivantes :
 - Ouvrir Workflow Studio et modifiez à nouveau votre flux de travail.
 - Pour Infrastructure Composer depuis la console, vous pouvez ouvrir la vue Modèle de votre application et modifier votre modèle. Si vous utilisez la synchronisation locale, vous pouvez modifier votre flux de travail dans votre localIDE. Infrastructure Composer détectera vos modifications et mettra à jour votre flux de travail dans Infrastructure Composer.
 - Pour Infrastructure Composer depuis le Toolkit for VS Code, vous pouvez directement modifier votre modèle. Infrastructure Composer détectera vos modifications et mettra à jour votre flux de travail dans Infrastructure Composer.

Pour importer des flux de travail existants

Vous pouvez importer des flux de travail à partir d'applications définies à l'aide de modèles AWS Serverless Application Model (AWS SAM). Utilisez n'importe quelle machine à états définie avec le type de `AWS::Serverless::StateMachine` ressource, et elle sera visualisée sous la forme d'une carte de composants améliorée Step Functions State Machine que vous pourrez utiliser pour lancer Workflow Studio.

La `AWS::Serverless::StateMachine` ressource peut définir des flux de travail à l'aide de l'une des propriétés suivantes :

- [Definition](#)— Le flux de travail est défini dans le AWS SAM modèle en tant qu'objet.
- [DefinitionUri](#)— Le flux de travail est défini sur un fichier externe à l'aide de l'[Amazon States Language](#). Le chemin local du fichier est ensuite spécifié avec cette propriété.

Propriété de définition

Infrastructure Composer depuis la console

Pour les flux de travail définis à l'aide de cette `Definition` propriété, vous pouvez importer un seul modèle ou l'ensemble du projet.

- **Modèle** : pour obtenir des instructions sur l'importation d'un modèle, consultez [Importer un modèle de projet existant dans la console Infrastructure Composer](#). Pour enregistrer les modifications que vous apportez dans Infrastructure Composer, vous devez exporter votre modèle.
- **Projet** — Lorsque vous importez un projet, vous devez activer la synchronisation locale. Les modifications que vous apportez sont automatiquement enregistrées sur votre ordinateur local. Pour obtenir des instructions sur l'importation d'un projet, consultez [Importer un dossier de projet existant dans la console Infrastructure Composer](#).

Compositeur d'infrastructure issu du Toolkit for VS Code

Pour les flux de travail définis à l'aide de cette `Definition` propriété, vous pouvez ouvrir Infrastructure Composer à partir de votre modèle. Pour obtenir des instructions, consultez [Accédez à Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code](#).

DefinitionUri propriété

Infrastructure Composer depuis la console

Pour les flux de travail définis à l'aide de cette `DefinitionUri` propriété, vous devez importer le projet et activer la synchronisation locale. Pour obtenir des instructions sur l'importation d'un projet, consultez [Importer un dossier de projet existant dans la console Infrastructure Composer](#).

Compositeur d'infrastructure issu du Toolkit for VS Code

Pour les flux de travail définis à l'aide de cette `DefinitionUri` propriété, vous pouvez ouvrir Infrastructure Composer à partir de votre modèle. Pour obtenir des instructions, consultez [Accédez à Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code](#).

Utilisation de Step Functions Workflow Studio dans Infrastructure Composer

Créez des flux de travail

Infrastructure Composer utilise des substitutions de définitions pour mapper les tâches du flux de travail aux ressources de votre application. Pour en savoir plus sur les substitutions de définitions, consultez [DefinitionSubstitutions](#) le guide du AWS Serverless Application Model développeur.

Lorsque vous créez des tâches dans Workflow Studio, spécifiez une substitution de définition pour chaque tâche. Vous pouvez ensuite connecter les tâches aux ressources sur le canevas Infrastructure Composer.

Pour spécifier une substitution de définition dans Workflow Studio

1. Ouvrez l'onglet Configuration de la tâche et recherchez le champ `APIParamètres`.

The screenshot displays the AWS Infrastructure Composer interface. On the left, a workflow diagram is shown on a grid background. It starts with a 'Start' node, followed by a 'Lambda: Invoke Check Stock Value' task. This leads to a 'Choice state Choice' with two paths: one 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.

On the right, the configuration panel for the 'Check Stock Value' state is visible. It includes tabs for 'Configuration', 'Input', 'Output', and 'Error handling'. The 'Configuration' tab is active, showing the following fields:

- State name:** Check Stock Value
- API:** Lambda: Invoke
- Integration type:** Optimized
- API Parameters:** Edit as JSON (toggle)
- Function name:** Enter a CloudFormation substitution (dropdown menu)
- Substitution:** \${LambdaFunction1}

Below the substitution field, a note 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."

2. Si le champ API Paramètres comporte une option déroulante, choisissez Entrer une AWS CloudFormation substitution. Entrez ensuite un nom unique.

Pour les tâches qui se connectent à la même ressource, spécifiez la même substitution de définition pour chaque tâche. Pour utiliser une substitution de définition existante, choisissez Sélectionner une AWS CloudFormation substitution et sélectionnez la substitution à utiliser.

3. Si le champ API Paramètres contient un JSON objet, modifiez l'entrée qui indique le nom de la ressource pour utiliser une substitution de définition. Dans l'exemple suivant, nous passons "MyDynamoDBTable" à "\${RecordTransaction}".

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' task. A 'Choice state Choice' follows, with a transition labeled '\$.stock_price <= 50' leading to a 'Lambda: Invoke Buy Stock' task, and a 'Default' transition leading to a 'Lambda: Invoke Sell Stock' task. Both tasks lead to a 'DynamoDB: PutItem Record Transaction' task, which ends at an 'End' node.

On the right, the configuration panel for the state 'Record Transaction' is shown. It includes tabs for 'Configuration', 'Input', 'Output', and 'Error handling'. The 'Configuration' tab is active, showing the state name 'Record Transaction', the API 'DynamoDB: PutItem', and the integration type 'Optimized'. Below this, the 'API Parameters' section shows a JSON object:

```

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

```

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

4. Sélectionnez Enregistrer et revenez à Infrastructure Composer.

Les tâches de votre flux de travail seront visualisées sur la carte machine Step Functions State.

The screenshot shows a card for a 'Step Functions State machine' named 'StateMachine'. The card features a pink icon of a state machine and a list of tasks with radio buttons to their right:

- Check Stock Value
- Buy Stock
- Sell Stock
- Record Transaction

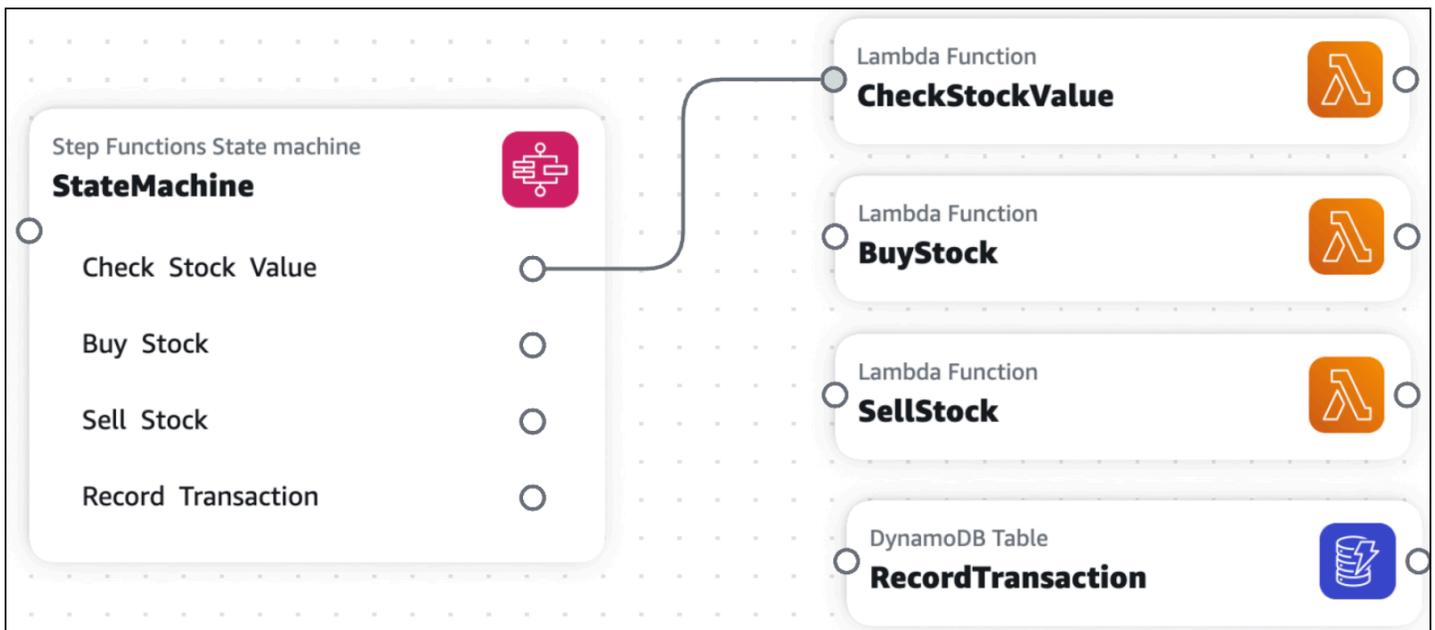
Connect les ressources aux tâches du flux de travail

Vous pouvez créer des connexions dans Infrastructure Composer entre les tâches de flux de travail prises en charge et les cartes Infrastructure Composer prises en charge.

- Tâches de flux de travail prises en charge : Services AWS les tâches correspondantes sont optimisées pour Step Functions. Pour en savoir plus, consultez la section [Intégrations optimisées pour Step Functions](#) dans le Guide du AWS Step Functions développeur.
- Cartes Infrastructure Composer prises en charge : les cartes de composants améliorées sont prises en charge. Pour en savoir plus sur les cartes dans Infrastructure Composer, consultez [Configurer et modifier des cartes dans Infrastructure Composer](#).

Lors de la création d'une connexion, Service AWS la tâche et la carte doivent correspondre. Par exemple, vous pouvez connecter une tâche de flux de travail qui appelle une fonction Lambda à une carte de composant améliorée pour une fonction Lambda.

Pour créer une connexion, cliquez et faites glisser le port d'une tâche vers le port gauche d'une carte de composant améliorée.



Infrastructure Composer mettra automatiquement à jour votre `DefinitionSubstitution` valeur pour définir votre connexion. Voici un exemple :

```
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:
        ...
```

Utilisation de fichiers externes

Lorsque vous créez un flux de travail à partir de la carte Step Functions State machine, Infrastructure Composer enregistre votre définition de machine à états dans votre modèle à l'aide de la `Definition` propriété. Vous pouvez configurer Infrastructure Composer pour enregistrer la définition de votre machine d'état sur un fichier externe.

Note

Pour utiliser cette fonctionnalité avec Infrastructure Composer depuis le AWS Management Console, la synchronisation locale doit être activée. Pour de plus amples informations, veuillez consulter [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#).

Pour enregistrer la définition de votre machine à états dans un fichier externe

1. Ouvrez le panneau des propriétés des ressources de votre carte machine Step Functions State.
2. Sélectionnez l'option Utiliser un fichier externe pour la définition de la machine à états.

3. Fournissez un chemin et un nom relatifs pour le fichier de définition de votre machine à états.
4. Choisissez Save (Enregistrer).

Infrastructure Composer effectuera les opérations suivantes :

1. Déplacez la définition de votre machine à états du `Definition` champ vers votre fichier externe.
2. Enregistrez la définition de votre machine à états dans un fichier externe à l'aide de l'Amazon States Language.
3. Modifiez votre modèle pour référencer le fichier externe à l'aide du `DefinitionUri` champ.

En savoir plus

Pour en savoir plus sur Step Functions dans Infrastructure Composer, consultez ce qui suit :

- [En utilisant Workflow Studio dans Infrastructure Composer](#) dans le guide du AWS Step Functions développeur.
- [DefinitionSubstitutions dans les AWS SAM modèles](#) du Guide du AWS Step Functions développeur.

Cartes standard dans Infrastructure Composer

Toutes les AWS CloudFormation ressources peuvent être utilisées sous forme de cartes de ressources laC standard dans la palette Ressources. Après avoir été glissée sur le canevas visuel, une carte de ressources laC standard devient une carte de composant standard. Cela signifie simplement que la carte est une ou plusieurs ressources laC standard. Pour plus d'exemples et de détails, consultez les rubriques de cette section.

Vous pouvez modifier le code de votre infrastructure par le biais de la vue Modèle et de la fenêtre des propriétés des ressources. Par exemple, voici un exemple de modèle de départ d'une ressource `Alexa::ASK::Skill` laC standard :

```
Resources:
  Skill:
    Type: Alexa::ASK::Skill
    Properties:
      AuthenticationConfiguration:
        RefreshToken: <String>
```

```
ClientSecret: <String>
ClientId: <String>
VendorId: <String>
SkillPackage:
  S3Bucket: <String>
  S3Key: <String>
```

Un modèle de départ de carte de ressources IaC standard comprend les éléments suivants :

- Type AWS CloudFormation de ressource.
- Propriétés requises ou couramment utilisées.
- Type requis de valeur à fournir pour chaque propriété.

Note

Vous pouvez utiliser ... Amazon Q pour générer des suggestions de code d'infrastructure pour les cartes de ressources standard. Pour en savoir plus, consultez [Utilisation AWS Infrastructure Composer avec Amazon Q Developer](#).

Procédure

Vous pouvez modifier le code d'infrastructure pour chaque ressource d'une carte de composant standard via le panneau des propriétés des ressources.

Pour modifier une carte de composant standard

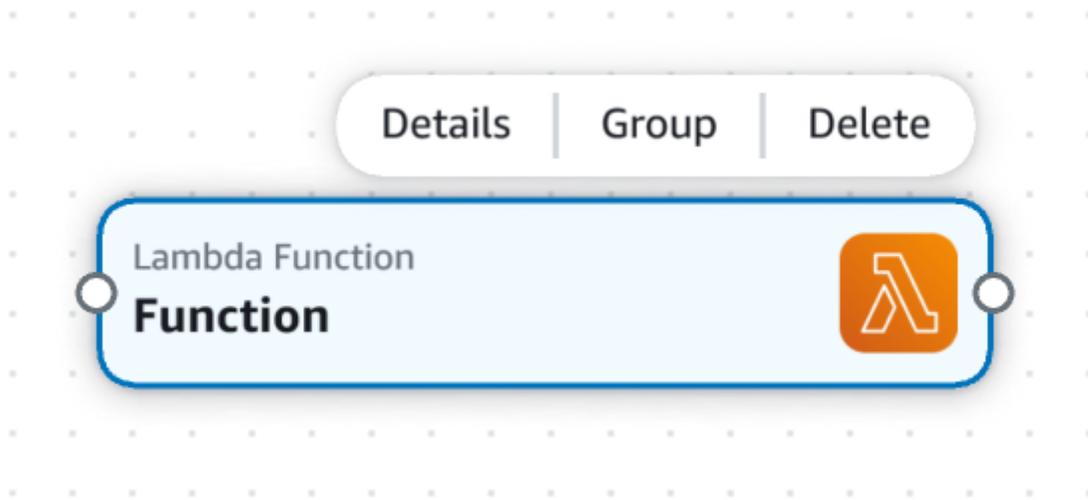
1. Ouvrez le panneau des propriétés des ressources de la carte de composant iAC standard.
2. Dans le champ Édition, sélectionnez la ressource iAc standard à modifier dans la liste déroulante.
3. Modifiez votre code d'infrastructure et enregistrez.

Supprimer des cartes dans Infrastructure Composer

Cette section fournit des instructions pour supprimer des cartes dans AWS Infrastructure Composer.

Cartes de composants améliorées

Pour supprimer une fiche de composant améliorée, sélectionnez une carte que vous avez placée sur le canevas visuel. Dans le menu Actions de la carte, sélectionnez Supprimer.



Cartes à composants standard

Pour supprimer des cartes de composants standard, vous devez supprimer manuellement le code d'infrastructure de chaque AWS CloudFormation ressource de votre modèle. Voici un moyen simple d'y parvenir :

1. Prenez note de l'ID logique de la ressource à supprimer.
2. Sur votre modèle, localisez la ressource par son identifiant logique dans la `Outputs` section `Resources` ou.
3. Supprimez la ressource de votre modèle. Cela inclut l'ID logique de la ressource et ses valeurs imbriquées, telles que `Type` et `Properties`.
4. Vérifiez la vue Canvas pour vérifier que la ressource a été supprimée de votre canevas.

Afficher les mises à jour du code avec le Change Inspector dans Infrastructure Composer

Lorsque vous concevez dans la console Infrastructure Composer, votre code d'infrastructure est automatiquement créé. Utilisez le Change Inspector pour consulter les mises à jour du code de votre modèle et découvrir ce qu'Infrastructure Composer crée pour vous.

Cette rubrique traite de l'utilisation d'Infrastructure Composer à partir de l' AWS Toolkit for Visual Studio Code extension AWS Management Console ou.

Le Change Inspector est un outil visuel intégré à Infrastructure Composer qui affiche les mises à jour récentes du code.

- Lorsque vous concevez votre application, des messages s'affichent en bas du canevas visuel. Ces messages fournissent des commentaires sur les actions que vous effectuez.
- Lorsque cela est pris en charge, vous pouvez développer un message pour afficher le Change Inspector.
- Le Change Inspector affiche les modifications de code depuis votre dernière interaction.

L'exemple suivant illustre le fonctionnement de l'inspecteur des modifications :

The screenshot displays the AWS Infrastructure Composer interface. On the left, there is a 'Resources' sidebar with a search bar and a list of resource types including API Gateway, Cognito UserPool, DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, Lambda Function, Lambda Layer, and S3 Bucket. The main canvas shows a visual diagram with a 'Lambda Function' resource connected to an 'S3 Bucket' resource. A 'Change Inspector' window is open, displaying a code diff for the bucket's IAM policy. The diff shows the addition of several permissions: s3:GetObject, s3:GetObjectAcl, s3:GetObjectLegalHold, and s3:GetObjectRetention. The right sidebar shows the 'Resource properties' for the selected S3 Bucket, including options for Logical ID, static website hosting, and public access.

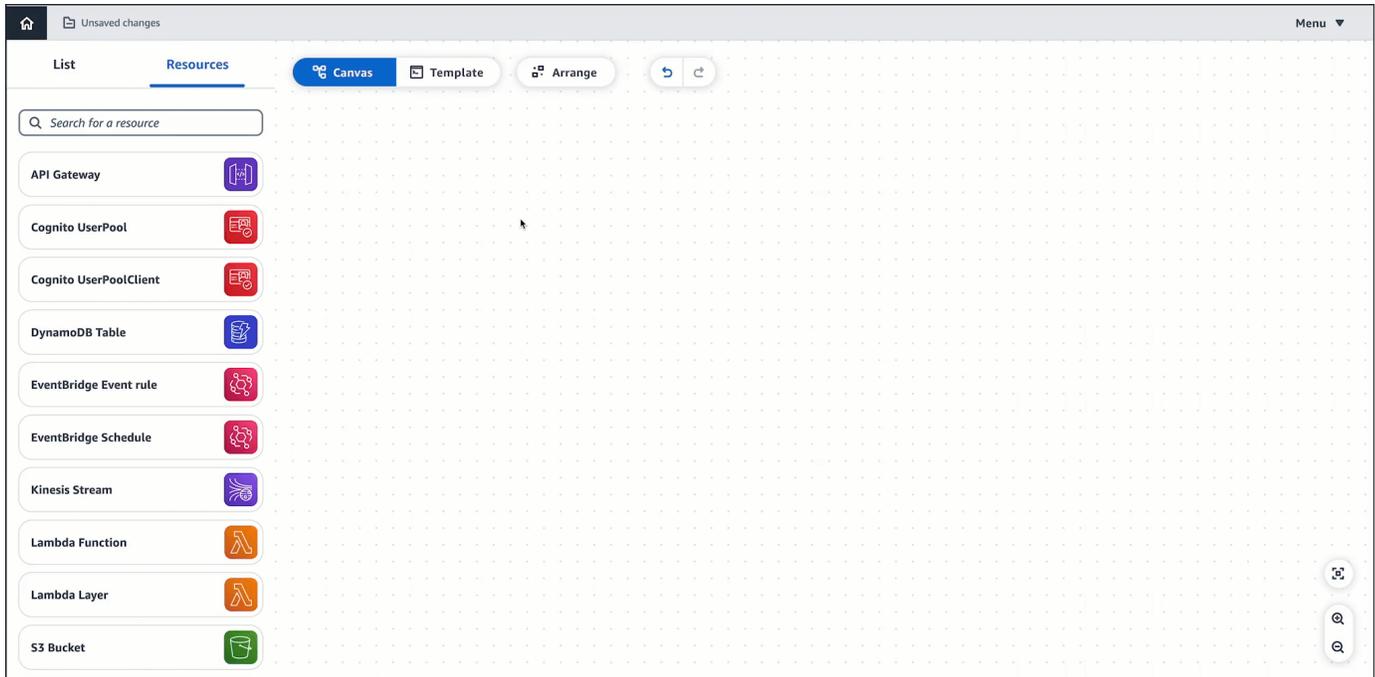
Avantages du Change Inspector

Le Change Inspector est un excellent moyen de visualiser le code modèle créé pour vous par Infrastructure Composer. C'est également un excellent moyen d'apprendre à écrire du code d'infrastructure. Lorsque vous concevez des applications dans Infrastructure Composer, consultez les mises à jour du code dans le Change Inspector pour en savoir plus sur le code nécessaire pour configurer votre conception.

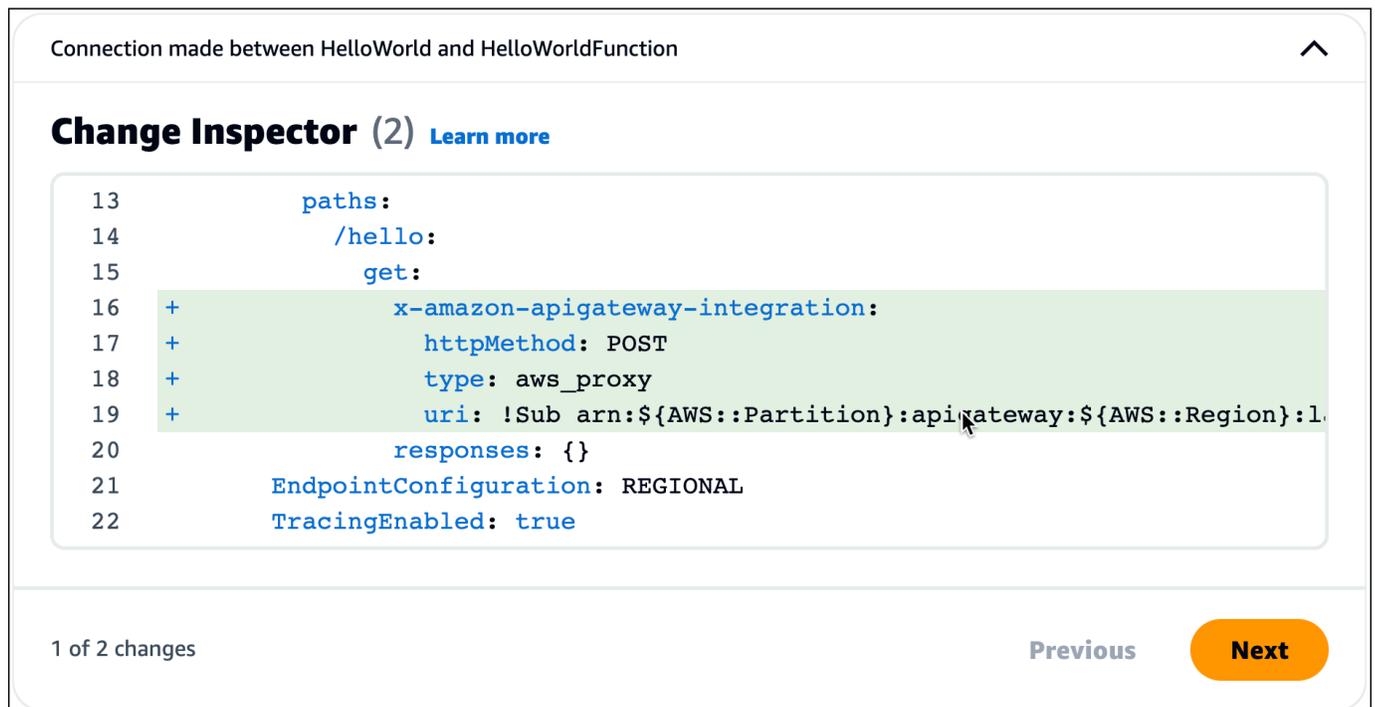
Procédure

Pour utiliser le Change Inspector

1. Développez un message pour faire apparaître le Change Inspector.



2. Consultez le code qui a été automatiquement composé pour vous.



- a. Le code surligné en vert indique le code récemment ajouté.
 - b. Le code surligné en rouge indique le code récemment supprimé.
 - c. Les numéros de ligne indiquent l'emplacement dans votre modèle.
3. Lorsque plusieurs sections de votre modèle ont été mises à jour, le Change Inspector les organise. Sélectionnez les boutons Précédent et Suivant pour afficher toutes les modifications.

Connection made between HelloWorld and HelloWorldFunction

Change Inspector (2) [Learn more](#)

```
13     paths:
14       /hello:
15         get:
16     +       x-amazon-apigateway-integration:
17     +         httpMethod: POST
18     +         type: aws_proxy
19     +         uri: !Sub arn:${AWS::Partition}:apigateway:${AWS::Region}:l
20           responses: {}
21     EndpointConfiguration: REGIONAL
22     TracingEnabled: true
```

1 of 2 changes

Previous **Next**

Note

Pour Infrastructure Composer depuis la console, vous pouvez afficher les modifications de code dans le contexte de l'ensemble de votre modèle, à l'aide de la vue des modèles. Vous pouvez également synchroniser Infrastructure Composer avec un modèle local IDE et afficher l'intégralité de votre modèle sur votre machine locale. Pour en savoir plus, consultez [Connectez la console Infrastructure Composer à votre console locale IDE](#).

En savoir plus

Pour plus d'informations sur le code créé par Infrastructure Composer, consultez ce qui suit :

- [Connexions par carte dans Infrastructure Composer](#).

Référencer des fichiers externes dans Infrastructure Composer

Vous pouvez utiliser des fichiers externes avec vos modèles AWS Serverless Application Model (AWS SAM) pour réutiliser le code répété et organiser vos projets. Par exemple, vous pouvez disposer de plusieurs REST API ressources Amazon API Gateway décrites par un OpenAPI spécification. Au lieu de reproduire le OpenAPI code de spécification dans votre modèle, vous pouvez créer un fichier externe et le référencer pour chacune de vos ressources.

AWS Infrastructure Composer prend en charge les cas d'utilisation de fichiers externes suivants :

- Passerelle API REST API ressources définies par des entités externes OpenAPI fichiers de spécifications.
- AWS Step Functions ressources de machine d'état définies par des fichiers de définition de machine d'état externes.

Pour en savoir plus sur la configuration de fichiers externes pour les ressources prises en charge, consultez les rubriques suivantes :

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

Note

Pour référencer des fichiers externes avec Infrastructure Composer depuis la console Infrastructure Composer, vous devez utiliser Infrastructure Composer en mode synchronisation locale. Pour de plus amples informations, veuillez consulter [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#).

Rubriques

- [Bonnes pratiques pour les fichiers de référence externes d'Infrastructure Composer](#)
- [Création d'une référence de fichier externe dans Infrastructure Composer](#)
- [Charger un projet avec une référence de fichier externe dans Infrastructure Composer](#)
- [Création d'une application qui référence un fichier externe dans Infrastructure Composer](#)
- [Référencer un OpenAPI fichier externe de spécification avec Infrastructure Composer](#)

Bonnes pratiques pour les fichiers de référence externes d'Infrastructure Composer

Utiliser Infrastructure Composer avec un outil local IDE

Lorsque vous utilisez Infrastructure Composer avec un local IDE en mode de synchronisation local, vous pouvez utiliser votre local IDE pour afficher et modifier des fichiers externes. Le contenu des fichiers externes pris en charge référencés sur votre modèle sera automatiquement mis à jour dans le canevas Infrastructure Composer. Pour en savoir plus, consultez [Connectez la console Infrastructure Composer à votre console locale IDE](#).

Conservez les fichiers externes dans le répertoire parent de votre projet

Vous pouvez créer des sous-répertoires dans le répertoire parent de votre projet pour organiser vos fichiers externes. Infrastructure Composer ne peut pas accéder aux fichiers externes stockés dans un répertoire en dehors du répertoire parent de votre projet.

Déployez votre application à l'aide du AWS SAM CLI

Lorsque vous déployez votre application sur le AWS Cloud, les fichiers externes locaux doivent d'abord être téléchargés vers un emplacement accessible, tel qu'Amazon Simple Storage Service (Amazon S3). Vous pouvez utiliser le AWS SAM CLI pour faciliter automatiquement ce processus. Pour en savoir plus, consultez la section [Télécharger des fichiers locaux lors du déploiement](#) dans le Guide du AWS Serverless Application Model développeur.

Création d'une référence de fichier externe dans Infrastructure Composer

Vous pouvez créer une référence de fichier externe à partir du panneau des propriétés des ressources prises en charge.

Pour créer une référence de fichier externe

1. À partir d'une carte de composants améliorée de APIGateway ou Step Functions, sélectionnez Détails pour afficher le panneau des propriétés des ressources.
2. Recherchez et sélectionnez l'option Utiliser un fichier externe.
3. Spécifiez le chemin relatif vers le fichier externe. Il s'agit du chemin de votre `template.yaml` fichier vers le fichier externe.

Par exemple, pour référencer le fichier `api-spec.yaml` externe à partir de la structure du projet suivant, spécifiez-le `./api-spec.yaml` comme chemin relatif.

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

Note

Si le fichier externe et son chemin spécifié n'existent pas, Infrastructure Composer le créera.

4. Enregistrez vos modifications.

Charger un projet avec une référence de fichier externe dans Infrastructure Composer

Suivez les étapes répertoriées sur cette page pour charger un projet Infrastructure Composer avec une référence de fichier externe.

Depuis la console Infrastructure Composer

1. Suivez les étapes répertoriées dans [Importer un modèle de projet existant dans la console Infrastructure Composer](#).
2. Confirmez qu'Infrastructure Composer vous invite à vous connecter au dossier racine de votre projet

Si votre navigateur prend en charge l'accès au système de fichiers API, Infrastructure Composer vous demandera de vous connecter au dossier racine de votre projet. Infrastructure Composer ouvrira votre projet en mode de synchronisation locale pour prendre en charge votre fichier externe. Si le fichier externe référencé n'est pas pris en charge, vous recevrez un message d'erreur. Pour plus d'informations sur les messages d'erreur, consultez [Résolution des problèmes](#).

Extrait du Toolkit for VS Code

1. Suivez les étapes répertoriées dans [Accédez à Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code](#).
2. Ouvrez le modèle que vous souhaitez afficher dans Infrastructure Composer.

Lorsque vous accédez à Infrastructure Composer à partir d'un modèle, Infrastructure Composer détecte automatiquement votre fichier externe. Si le fichier externe référencé n'est pas pris en charge, vous recevrez un message d'erreur. Pour plus d'informations sur les messages d'erreur, consultez [Résolution des problèmes](#).

Création d'une application qui référence un fichier externe dans Infrastructure Composer

Cet exemple utilise le AWS SAM CLI pour créer une application qui fait référence à un fichier externe pour sa définition de machine à états. Vous chargez ensuite votre projet dans Infrastructure Composer avec votre fichier externe correctement référencé.

Exemple

1. Tout d'abord, utilisez la AWS SAM CLI `sam init` commande pour initialiser une nouvelle application nommée `demo`. Pendant le flux interactif, sélectionnez le modèle de démarrage rapide du flux de travail en plusieurs étapes.

```
$ 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
```

```
...
```

Cette application fait référence à un fichier externe pour la définition de la machine à états.

```
...
```

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

Le fichier externe se trouve dans le `statemachine` sous-répertoire de notre application.

```
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. Chargez ensuite votre application dans Infrastructure Composer depuis la console. Sur la page d'accueil d'Infrastructure Composer, sélectionnez Charger un CloudFormation modèle.
3. Sélectionnez notre dossier de demo projet et autorisez l'invite à afficher les fichiers. Sélectionnez notre `template.yaml` fichier, puis sélectionnez Créer. Lorsque vous y êtes invité, sélectionnez Enregistrer les modifications.

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 détecte automatiquement le fichier de définition de la machine d'état externe et le charge. Sélectionnez notre `StockTradingStateMachineressource` et choisissez `Détails` pour afficher le panneau des propriétés de la ressource. Ici, vous pouvez voir qu'Infrastructure Composer s'est automatiquement connecté à notre fichier de définition de machine à états externe.

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 including Lambda Functions (StockCheckerFunction, StockSellerFunction, StockBuyerFunction), a DynamoDB Table (TransactionTable), and an EventBridge Timer (ImplicitTimer). A 'Step Functions State machine' resource (StockTradingStateMachine) is highlighted, and its 'Details' panel is open on the right. This panel shows the state machine definition in JSON format, including tasks like 'Check Stock Value', 'Sell Stock', 'Buy Stock', and 'Record Transaction'. The 'Use external file for state machine definition' checkbox is checked.

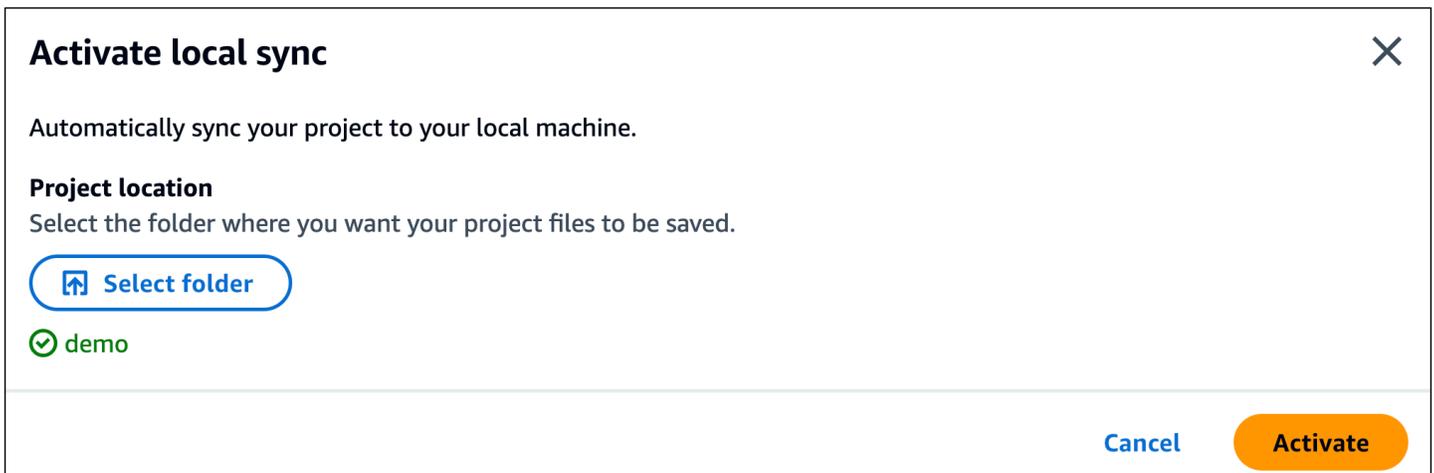
Toute modification apportée au fichier de définition de la machine d'état sera automatiquement reflétée dans Infrastructure Composer.

Référez un OpenAPI fichier externe de spécification avec Infrastructure Composer

Cet exemple utilise Infrastructure Composer depuis la console pour faire référence à un OpenAPI fichier de spécification qui définit une API passerelle REST API.

Créez d'abord un nouveau projet depuis la page d'accueil d'Infrastructure Composer.

Activez ensuite la synchronisation locale en sélectionnant Activer la synchronisation locale dans le menu. Créez un nouveau dossier nommé demo, autorisez l'invite à afficher les fichiers, puis sélectionnez Activer. Lorsque vous y êtes invité, sélectionnez Enregistrer les modifications.



Ensuite, faites glisser une carte Amazon API Gateway sur le canevas. Sélectionnez Détails pour afficher le panneau des propriétés des ressources.

The screenshot shows the AWS Infrastructure Composer interface. On the left, there is a 'Resources' list with various AWS services. In the center, a canvas displays an API Gateway resource named 'Api' with a 'GET /' endpoint. On the right, the 'Resource properties' panel is open, showing configuration options for the API Gateway. The 'Logical ID' is set to 'Api'. The 'Default authorizer' is set to 'None'. The 'Method' is set to 'GET'. The 'Path' field is visible at the bottom of the panel.

Dans le panneau des propriétés de la ressource, configurez les éléments suivants et enregistrez.

- Sélectionnez l'option Utiliser un fichier externe pour la définition de l'API.
- Entrée `./api-spec.yaml` en tant que chemin relatif vers le fichier externe

Use external file for api definition



Relative path to external file

`./api-spec.yaml`

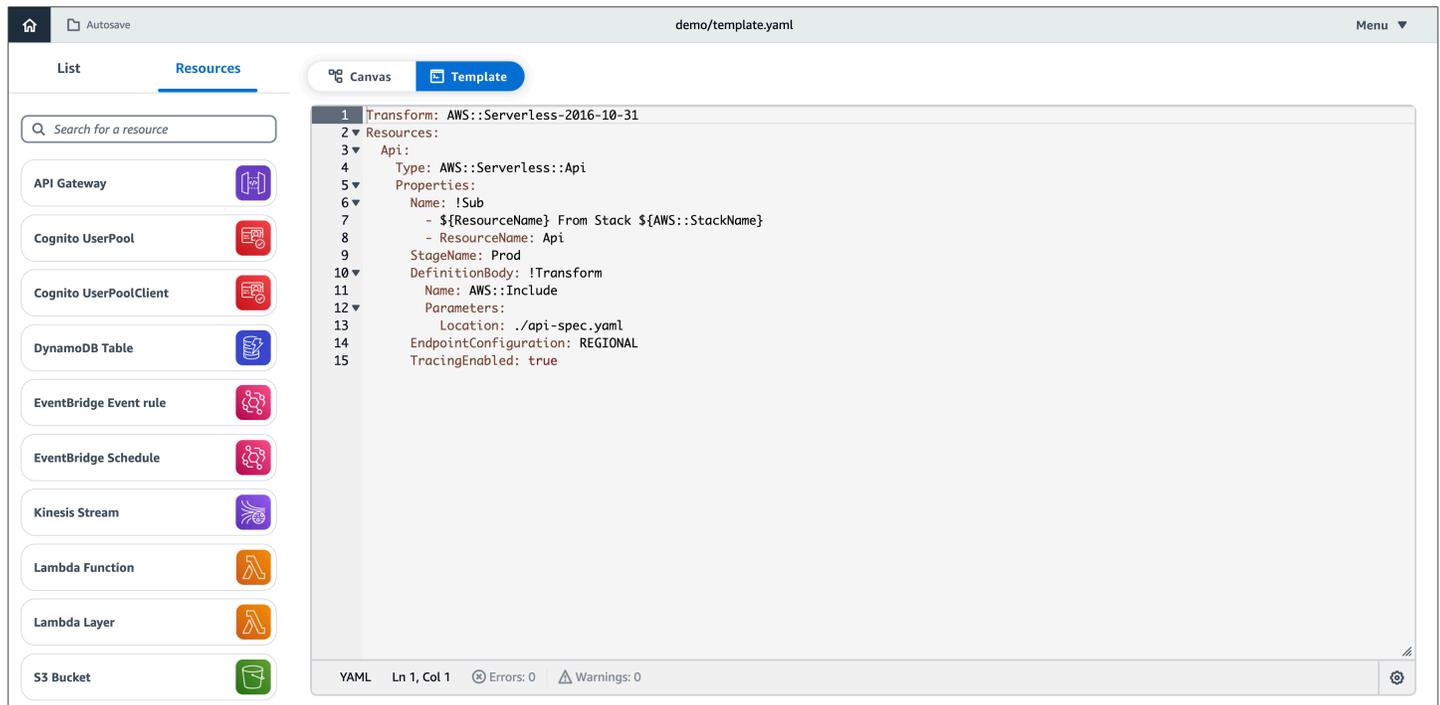
Cela crée le répertoire suivant sur notre machine locale :

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

Vous pouvez maintenant configurer le fichier externe sur notre machine locale. À l'aide de notre IDE, ouvrez le fichier `api-spec.yaml` situé dans le dossier de votre projet. Remplacez son contenu par ce qui suit :

```
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
```

Dans la vue Modèle d'Infrastructure Composer, vous pouvez voir qu'Infrastructure Composer a automatiquement mis à jour votre modèle pour référencer le fichier externe.



Intégrer Infrastructure Composer à Amazon Virtual Private Cloud (AmazonVPC)

AWS Infrastructure Composer propose une intégration avec le service Amazon Virtual Private Cloud (AmazonVPC). À l'aide d'Infrastructure Composer, vous pouvez effectuer les opérations suivantes :

- Identifiez les ressources de votre canevas qui se trouvent dans une VPC balise visuelle VPC via une balise.
- Configurez AWS Lambda les fonctions VPCs à partir d'un modèle externe.

L'image suivante montre un exemple d'application avec une fonction Lambda configurée avec un VPC



Pour en savoir plus sur AmazonVPC, consultez [Qu'est-ce qu'Amazon VPC ?](#) dans le guide de VPC l'utilisateur Amazon.

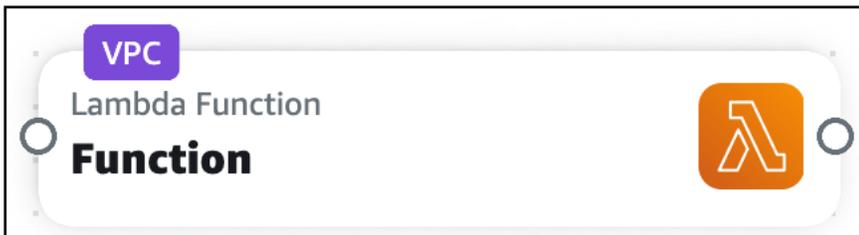
Rubriques

- [Identifier les ressources du compositeur d'infrastructure et les informations connexes dans un VPC](#)
- [Configurer les fonctions Lambda avec des fonctions externes VPCs dans Infrastructure Composer](#)
- [Paramètres dans les modèles importés pour un modèle externe VPC avec Infrastructure Composer](#)
- [Ajout de nouveaux paramètres aux modèles importés avec Infrastructure Composer](#)
- [Configurer une fonction Lambda et une fonction VPC définie dans un autre modèle avec Infrastructure Composer](#)

Identifier les ressources du compositeur d'infrastructure et les informations connexes dans un VPC

Pour intégrer Infrastructure Composer à AmazonVPC, vous devez d'abord identifier les ressources dans a VPC et les informations nécessaires pour effectuer une intégration. Cela inclut également les informations de configuration relatives aux groupes de sécurité, aux identificateurs de sous-réseaux, aux types de paramètres, aux types et aux SSM types de valeurs statiques.

Infrastructure Composer visualise les ressources dans un tag à VPC l'aide d'une VPCbalise. Cette étiquette est appliquée aux cartes sur le canevas. Voici un exemple de fonction Lambda avec une VPC balise :



VPCles balises sont appliquées aux cartes sur le canevas lorsque vous effectuez les opérations suivantes :

- Configurez une fonction Lambda avec un compositeur intégré VPC à l'infrastructure.
- Importez un modèle contenant des ressources configurées avec unVPC.

Identifiants de groupes de sécurité et de sous-réseaux

Une fonction Lambda peut être configurée avec plusieurs groupes de sécurité et sous-réseaux. Pour configurer un groupe de sécurité ou un sous-réseau pour une fonction Lambda, entrez une valeur et un type.

- Valeur — Identifiant du groupe de sécurité ou du sous-réseau. Les valeurs acceptées varient en fonction du type.
- Type — Les types de valeurs suivants sont autorisés :
 - Nom du paramètre
 - AWS Systems Manager (SSM) Magasin de paramètres
 - Valeur statique

Type de paramètre

La `Parameters` section d'un AWS CloudFormation modèle peut être utilisée pour stocker des informations sur les ressources dans plusieurs modèles. Pour plus d'informations sur les paramètres, consultez la section [Paramètres](#) du guide de AWS CloudFormation l'utilisateur.

Pour le type de paramètre, vous pouvez fournir un nom de paramètre. Dans l'exemple suivant, nous fournissons une valeur de nom de `PrivateSubnet1` paramètre :

Subnet IDs
List of VPC subnet identifiers

Value	Type
<input type="text" value="PrivateSubnet1"/>	<input type="text" value="Parameter"/>

Lorsque vous fournissez un nom de paramètre, Infrastructure Composer le définit dans la `Parameters` section de votre modèle. Infrastructure Composer référence ensuite le paramètre dans votre ressource de fonction Lambda. Voici un exemple :

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

```

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

```

SSM type

Le SSM Parameter Store fournit un stockage hiérarchique sécurisé pour la gestion des données de configuration et la gestion des secrets. Pour plus d'informations, veuillez consulter la rubrique [AWS Systems Manager Parameter Store](#) dans le Guide de l'utilisateur AWS Systems Manager .

Pour le SSMtype, vous pouvez fournir les valeurs suivantes :

- Référence dynamique à une valeur du SSM Parameter Store.
- ID logique d'une `AWS::SSM::Parameter` ressource définie dans votre modèle.

Référence dynamique

Vous pouvez référencer une valeur depuis le magasin de SSM paramètres à l'aide d'une référence dynamique au format suivant : `{{resolve:ssm:reference-key}}` Pour plus d'informations, consultez [SSMles paramètres](#) dans le guide de AWS CloudFormation l'utilisateur.

Infrastructure Composer crée le code d'infrastructure pour configurer votre fonction Lambda avec la valeur du SSM Parameter Store. Voici un exemple :

```

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

```

ID logique

Vous pouvez référencer une `AWS::SSM::Parameter` ressource dans le même modèle par ID logique.

Voici un exemple de `AWS::SSM::Parameter` ressource nommée `PrivateSubnet1Parameter` qui stocke l'ID de sous-réseau pour `PrivateSubnet1` :

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

Voici un exemple de cette valeur de ressource fournie par un ID logique pour la fonction Lambda :

Subnet IDs

List of VPC subnet identifiers

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

Infrastructure Composer crée le code d'infrastructure pour configurer votre fonction Lambda avec le SSM paramètre :

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

```
PrivateSubnet1Parameter:
  Type: AWS::SSM::Parameter
  Properties:
    ...
```

Type de valeur statique

Lorsqu'un groupe de sécurité ou un sous-réseau est déployé AWS CloudFormation, une valeur d'ID est créée. Vous pouvez fournir cet ID sous forme de valeur statique.

Pour le type de valeur statique, les valeurs suivantes sont valides :

- Pour les groupes de sécurité, fournissez leGroupId. Pour plus d'informations, consultez la section [Valeurs renvoyées](#) dans le Guide de AWS CloudFormation l'utilisateur. Voici un exemple :
sg-0b61d5c742dc2c773
- Pour les sous-réseaux, fournissez le. SubnetId Pour plus d'informations, consultez la section [Valeurs renvoyées](#) dans le Guide de AWS CloudFormation l'utilisateur. Voici un exemple :
subnet-01234567890abcdef

Infrastructure Composer crée le code d'infrastructure pour configurer votre fonction Lambda avec la valeur statique. Voici un exemple :

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

Utilisation de plusieurs types

Pour les groupes de sécurité et les sous-réseaux, vous pouvez utiliser plusieurs types ensemble. Voici un exemple qui configure trois groupes de sécurité pour une fonction Lambda en fournissant des valeurs de différents types :

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 référence les trois valeurs sous la `SecurityGroupIds` propriété :

```

...
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

```

Configurer les fonctions Lambda avec des fonctions externes VPCs dans Infrastructure Composer

Pour commencer à configurer une fonction Lambda avec un défini sur un VPC autre modèle, utilisez la carte de composants améliorée de la fonction Lambda. Cette carte représente une fonction Lambda utilisant le type de `AWS::Serverless::Function` ressource AWS Serverless Application Model (AWS SAM).

Pour configurer une fonction Lambda à l'aide VPC d'un modèle externe

1. Dans le panneau des propriétés des ressources de la fonction Lambda, développez la section déroulante des VPCparamètres (avancés).
2. Sélectionnez Attribuer à un externe VPC.
3. Fournissez des valeurs pour les groupes de sécurité et les sous-réseaux à configurer pour la fonction Lambda. Consultez [Identifiants de groupes de sécurité et de sous-réseaux](#) pour plus de détails.
4. Enregistrez vos modifications.

Paramètres dans les modèles importés pour un modèle externe VPC avec Infrastructure Composer

Lorsque vous importez un modèle existant avec des paramètres définis pour les groupes de sécurité et les sous-réseaux d'un modèle externeVPC, Infrastructure Composer fournit une liste déroulante dans laquelle vous pouvez sélectionner vos paramètres.

Voici un exemple de Parameters section d'un modèle importé :

```

...
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
...

```

Lorsque vous configurez une fonction externe VPC pour une nouvelle fonction Lambda sur le canevas, ces paramètres seront disponibles dans une liste déroulante. Voici un exemple :

Subnet IDs

List of VPC subnet identifiers

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

Limitations lors de l'importation de types de paramètres de liste

Normalement, vous pouvez spécifier plusieurs identificateurs de groupes de sécurité et de sous-réseaux pour chaque fonction Lambda. Si votre modèle existant contient des types de paramètres de liste, tels que `List<AWS::EC2::SecurityGroup::Id>` ou `List<AWS::EC2::Subnet::Id>`, vous ne pouvez spécifier qu'un seul identifiant.

Pour plus d'informations sur le type de listes de paramètres, consultez la section [Types de paramètres AWS spécifiques pris en charge](#) dans le Guide de AWS CloudFormation l'utilisateur.

Voici un exemple de modèle défini `VPCSecurityGroups` comme un type de paramètre de liste :

```

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

```

...

Dans Infrastructure Composer, si vous sélectionnez la `VPCSecurityGroups` valeur comme identifiant de groupe de sécurité pour une fonction Lambda, le message suivant s'affiche :

Security group IDs

List of VPC security group identifiers

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

Add new item

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

Cette limitation est due au fait que les `SubnetIds` propriétés `SecurityGroupIds` et d'un `AWS::Lambda::Function VpcConfig` objet n'acceptent toutes deux qu'une liste de valeurs de chaîne. Étant donné qu'un seul type de paramètre de liste contient une liste de chaînes, il peut être le seul objet fourni lorsqu'il est spécifié.

Pour les types de paramètres de liste, voici un exemple de la façon dont ils sont définis dans le modèle lorsqu'ils sont configurés avec une fonction 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
```

Ajout de nouveaux paramètres aux modèles importés avec Infrastructure Composer

Lorsque vous importez un modèle existant avec des paramètres définis, vous pouvez également créer de nouveaux paramètres. Au lieu de sélectionner un paramètre existant dans la liste déroulante, saisissez un nouveau type et une nouvelle valeur. Voici un exemple de création d'un nouveau paramètre nommé `MySecurityGroup` :

Security group IDs

List of VPC security group identifiers

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

Pour toutes les nouvelles valeurs que vous fournissez dans le panneau des propriétés des ressources pour la fonction Lambda, Infrastructure Composer les définit dans une liste sous les `SubnetIds` propriétés `SecurityGroupIds` ou d'une fonction Lambda. Voici un exemple :

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

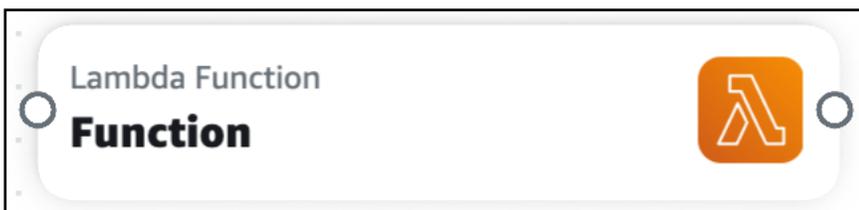
Si vous souhaitez référencer l'ID logique d'un type de paramètre de liste à partir d'un modèle externe, nous vous recommandons d'utiliser la vue Modèle et de modifier directement votre modèle. L'ID logique d'un type de paramètre de liste doit toujours être fourni sous forme de valeur unique et de valeur unique.

```
...
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
```

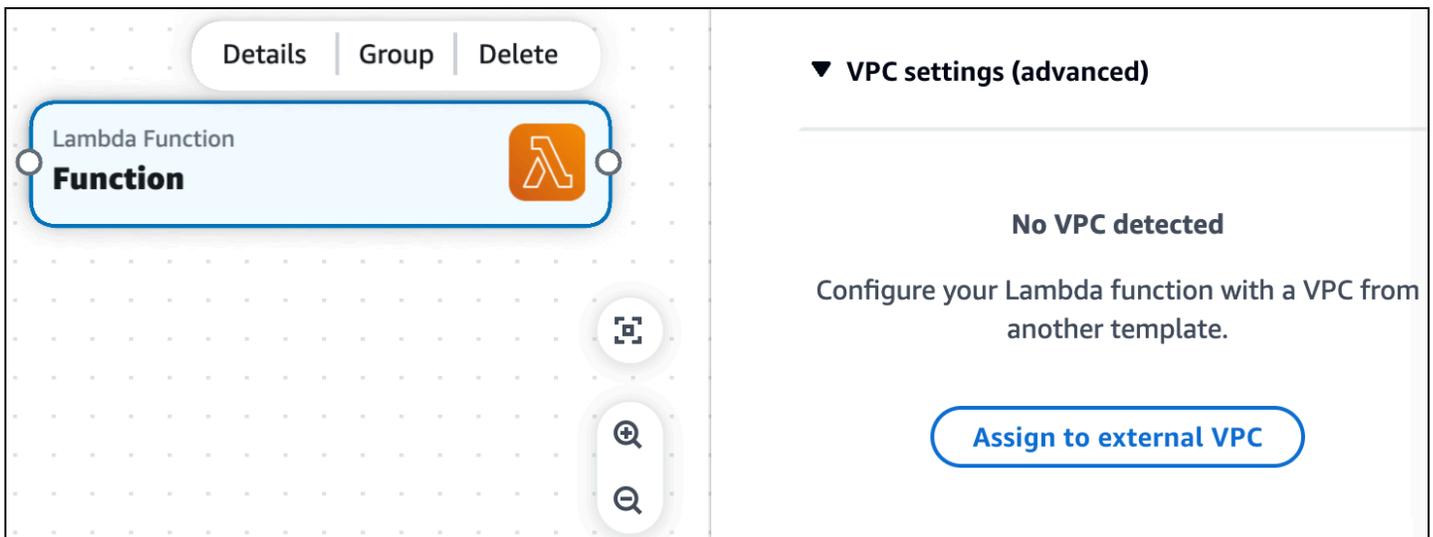
Configurer une fonction Lambda et une fonction VPC définie dans un autre modèle avec Infrastructure Composer

Dans cet exemple, nous configurons une fonction Lambda dans Infrastructure Composer avec une VPC définition sur un autre modèle.

Nous commençons par faire glisser une carte de composant améliorée de la fonction Lambda sur le canevas.



Ensuite, nous ouvrons le panneau des propriétés des ressources de la carte et élargissons la section déroulante des VPCparamètres (avancés).

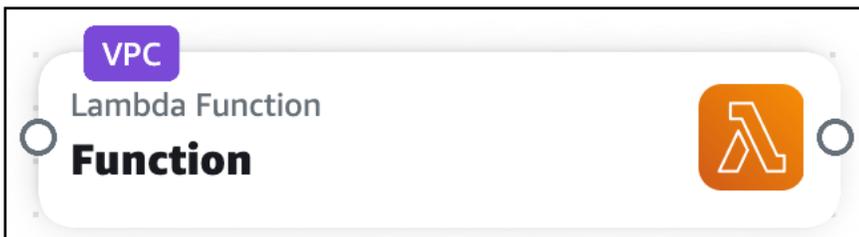


Ensuite, nous sélectionnons Attribuer à un modèle externe VPC pour commencer à configurer un modèle VPC à partir d'un modèle externe.

Dans cet exemple, nous faisons référence à un ID de groupe de sécurité et à un ID de sous-réseau. Ces valeurs sont créées lorsque le modèle qui les définit VPC est déployé. Nous choisissons le type de valeur statique et saisissons la valeur de notre IDs. Nous sélectionnons Enregistrer lorsque vous avez terminé.

The screenshot shows the configuration interface for a Lambda Function. On the left, a card labeled 'Lambda Function' with the AWS Lambda icon is visible, with buttons for 'Details', 'Group', and 'Delete'. On the right, the 'Security group IDs' section is active, showing a list of VPC security group identifiers. The 'Value' field contains 'sg-10f35d07e1be09e15' and the 'Type' is set to 'Static value'. Below this, the 'Subnet IDs' section is also active, showing a list of VPC subnet identifiers. The 'Value' field contains 'subnet-0d80727ca90325716' and the 'Type' is set to 'Static value'. At the bottom right, there are 'Cancel' and 'Save' buttons, and a 'Remove from VPC' button.

Maintenant que notre fonction Lambda est configurée avec notre VPC, le VPC tag est affiché sur notre carte.



Infrastructure Composer a créé le code d'infrastructure pour configurer notre fonction Lambda avec le groupe de sécurité et le sous-réseau de l'externe. VPC

```

Transform: AWS::Serverless-2016-10-31
Resources:
  Function:
    Type: AWS::Serverless::Function
    Properties:
      Description: !Sub
        - Stack ${AWS::StackName} Function ${ResourceName}
        - ResourceName: Function
  
```

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

Déployez votre application sans serveur Infrastructure Composer dans le cloud AWS

AWS Infrastructure Composer À utiliser pour concevoir des applications sans serveur prêtes à être déployées. Pour le déploiement, utilisez n'importe quel service AWS CloudFormation compatible. Nous vous recommandons d'utiliser le [AWS Serverless Application Model \(AWS SAM\)](#).

AWS SAM est un framework open source qui fournit des outils de développement pour créer et exécuter des applications sans serveur sur. AWS Avec AWS SAM sa syntaxe abrégée, les développeurs déclarent les AWS CloudFormation ressources et les ressources sans serveur spécialisées qui sont transformées en infrastructure lors du déploiement.

AWS SAM Concepts importants

Avant de l'utiliser AWS SAM, il est important que vous vous familiarisiez avec certains de ses concepts fondamentaux.

- [AWS SAM Fonctionnement](#) : Cette rubrique, qui figure dans le guide du AWS Serverless Application Model développeur, fournit des informations importantes sur les principaux composants que vous utilisez pour créer votre application sans service : AWS SAM CLI, le AWS SAM projet et le AWS SAM modèle.
- [Comment utiliser AWS Serverless Application Model \(AWS SAM\)](#) : cette rubrique, qui figure dans le guide du AWS Serverless Application Model développeur, fournit une présentation détaillée des étapes à suivre AWS SAM pour déployer votre application AWS dans le cloud.

Lorsque vous concevez votre application dans Infrastructure Composer, vous pouvez utiliser la `sam sync` commande pour avoir AWS SAM CLI détectez automatiquement les modifications locales et déployez ces modifications sur AWS CloudFormation. Pour en savoir plus, consultez la section [Utilisation de sam sync](#) dans le manuel du AWS Serverless Application Model développeur.

Étapes suivantes

Reportez-vous [Configuré pour le déploiement à l'aide du AWS SAM CLI et Compositeur d'infrastructure](#) à pour préparer le déploiement de votre application.

Configuré pour le déploiement à l'aide du AWS SAM CLI et Compositeur d'infrastructure

Pour déployer votre application avec AWS SAM, vous devez d'abord installer et accéder au AWS CLI et le AWS SAM CLI. Les rubriques de cette section fournissent des informations détaillées à ce sujet.

Installez le AWS CLI

Nous vous recommandons d'installer et de configurer AWS CLI avant d'installer AWS SAM CLI. Pour obtenir des instructions, voir [Installer ou mettre à jour la dernière version du AWS CLI dans le](#) guide de AWS Command Line Interface l'utilisateur.

Note

Après avoir installé AWS CLI, vous devez configurer les AWS informations d'identification. Pour en savoir plus, consultez la section [Configuration rapide](#) dans le guide de AWS Command Line Interface l'utilisateur.

Installez le AWS SAM CLI

Pour installer AWS SAM CLI, voir [Installation du AWS SAM CLI](#) dans le Manuel du développeur AWS Serverless Application Model .

Accédez au AWS SAM CLI

Si vous utilisez Infrastructure Composer depuis le AWS Management Console, vous disposez des options suivantes pour utiliser le AWS SAM CLI.

Activer le mode de synchronisation local

Avec le mode de synchronisation locale, le dossier de votre projet, y compris le AWS SAM modèle, est automatiquement enregistré sur votre ordinateur local. Infrastructure Composer structure le répertoire de votre projet de manière à AWS SAM reconnaître. Vous pouvez exécuter le AWS SAM CLI depuis le répertoire racine de votre projet.

Pour plus d'informations sur le mode de synchronisation locale, consultez [Synchronisez et enregistrez localement votre projet dans la console Infrastructure Composer](#).

Exportez votre modèle

Vous pouvez exporter votre modèle vers votre machine locale. Ensuite, lancez le AWS SAM CLI depuis le dossier parent qui contient le modèle. Vous pouvez également utiliser l'option `--template-file` avec n'importe quel AWS SAM CLI commande et indiquez le chemin d'accès à votre modèle.

Utilisez Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code

Vous pouvez utiliser Infrastructure Composer depuis le Toolkit for VS Code pour intégrer Infrastructure Composer sur votre machine locale. Ensuite, utilisez Infrastructure Composer et le AWS SAM CLI de VS Code.

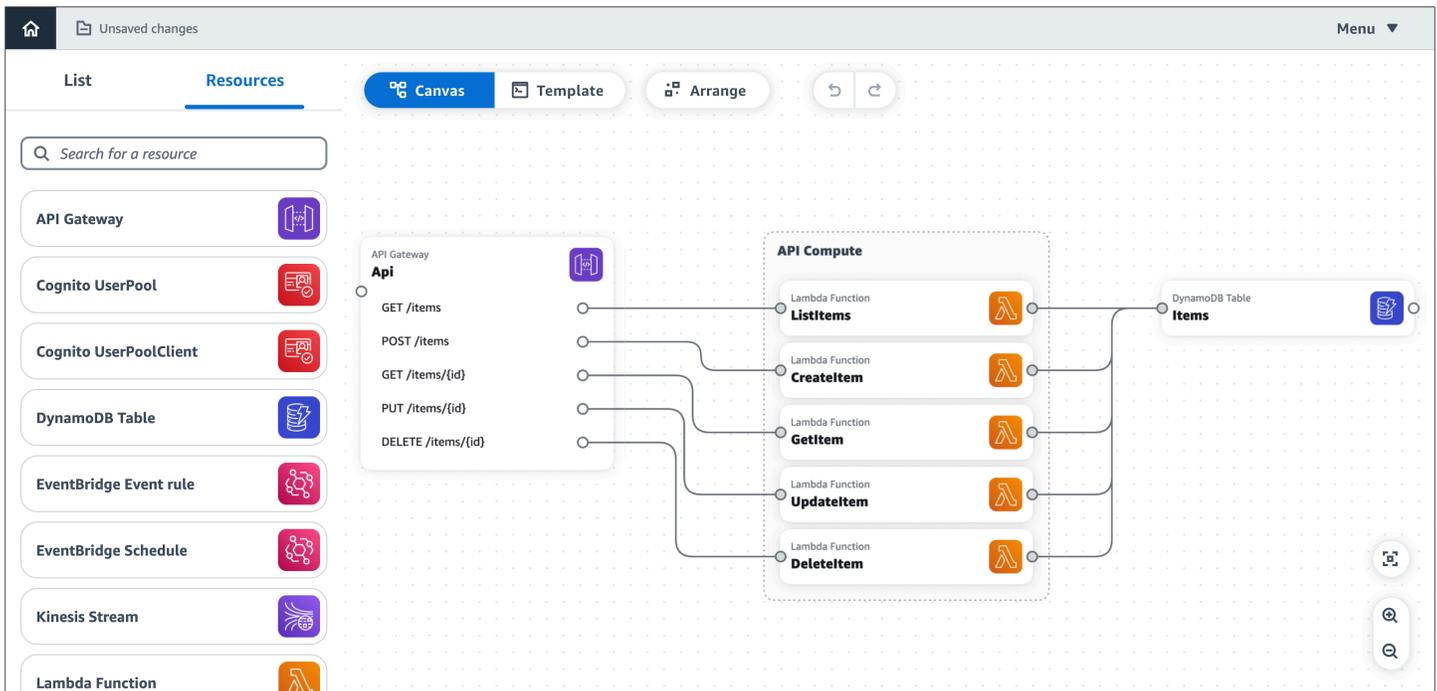
Étapes suivantes

Pour déployer votre application, reportez-vous à [Utilisez Infrastructure Composer AWS SAM pour créer et déployer](#).

Utilisez Infrastructure Composer AWS SAM pour créer et déployer

Maintenant que vous avez terminé [Configuré pour le déploiement à l'aide du AWS SAM CLI et Compositeur d'infrastructure](#), vous pouvez déployer votre application avec AWS SAM Infrastructure Composer. Cette section fournit un exemple expliquant comment procéder. Vous pouvez également consulter la section [Déployer votre application et vos ressources AWS SAM](#) dans le Guide du AWS Serverless Application Model développeur pour obtenir des instructions sur le déploiement de votre application avec AWS SAM.

Cet exemple montre comment créer et déployer l'application de démonstration Infrastructure Composer. L'application de démonstration contient les ressources suivantes :



Note

- Pour en savoir plus sur l'application de démonstration, consultez [Chargez et modifiez le projet de démonstration d'Infrastructure Composer](#).
- Dans cet exemple, nous utilisons Infrastructure Composer avec la synchronisation locale activée.

1. Utilisez la `sam build` commande pour créer l'application.

```
$ 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
```

Le AWS SAM CLI crée le `./aws-sam` répertoire dans le dossier du projet. Ce répertoire contient des artefacts de construction pour les fonctions Lambda de l'application. Voici un résultat du répertoire du projet :

```
.
### 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. L'application est maintenant prête à être déployée. Nous utiliserons `sam deploy --guided`. Cela prépare votre application au déploiement par le biais d'une série d'instructions.

```
$ 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]:

```

Le AWS SAM CLI affiche un résumé de ce qui sera déployé :

```

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

```

Le AWS SAM CLI déploie l'application, d'abord en créant un ensemble de AWS CloudFormation modifications :

```

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	ApiDeploymentcc153d135b 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
	Prod

```

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

```

```

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

```

Ensuite, le AWS SAM CLI déploie l'application :

```

CloudFormation events from stack operations (refresh every 0.5 seconds)
-----

```

ResourceStatus	LogicalResourceId	ResourceType	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	
	DeleteItemRole	-	
CREATE_IN_PROGRESS	-	AWS::IAM::Role	
	ListItemsRole	-	
CREATE_IN_PROGRESS	-	AWS::IAM::Role	
	UpdateItemRole	-	
CREATE_IN_PROGRESS	-	AWS::IAM::Role	GetItemRole
CREATE_IN_PROGRESS	-	AWS::IAM::Role	
	CreateItemRole	-	
CREATE_IN_PROGRESS	-	AWS::IAM::Role	Resource creation Initiated
	DeleteItemRole		
CREATE_IN_PROGRESS	-	AWS::IAM::Role	Resource creation Initiated
	ListItemsRole		
CREATE_IN_PROGRESS	-	AWS::IAM::Role	GetItemRole
	CreateItemRole		
CREATE_IN_PROGRESS	-	AWS::IAM::Role	Resource creation Initiated
	DeleteItemRole		
CREATE_IN_PROGRESS	-	AWS::IAM::Role	Resource creation Initiated
	CreateItemRole		

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

CREATE_COMPLETE	-	AWS::ApiGateway::RestApi	Api
CREATE_IN_PROGRESS	GetItemApiGETitemsidPermissionPro	AWS::Lambda::Permission	d
CREATE_IN_PROGRESS	ListItemsApiGETitemsPermissionPro	AWS::Lambda::Permission	d
CREATE_IN_PROGRESS	DeleteItemApiDELETEitemsidPermiss	AWS::Lambda::Permission	ionProd
CREATE_IN_PROGRESS	ApiDeploymentccc153d135b	AWS::ApiGateway::Deployment	
CREATE_IN_PROGRESS	UpdateItemApiPUTitemsidPermission	AWS::Lambda::Permission	Prod
CREATE_IN_PROGRESS	CreateItemApiPOSTitemsPermissionP	AWS::Lambda::Permission	rod
CREATE_IN_PROGRESS	GetItemApiGETitemsidPermissionPro	AWS::Lambda::Permission	d
CREATE_IN_PROGRESS	UpdateItemApiPUTitemsidPermission	AWS::Lambda::Permission	Prod
CREATE_IN_PROGRESS	CreateItemApiPOSTitemsPermissionP	AWS::Lambda::Permission	rod
CREATE_IN_PROGRESS	ListItemsApiGETitemsPermissionPro	AWS::Lambda::Permission	d
CREATE_IN_PROGRESS	DeleteItemApiDELETEitemsidPermiss	AWS::Lambda::Permission	ionProd
CREATE_IN_PROGRESS	ApiDeploymentccc153d135b	AWS::ApiGateway::Deployment	
CREATE_COMPLETE	ApiDeploymentccc153d135b	AWS::ApiGateway::Deployment	
CREATE_IN_PROGRESS	ApiProdStage	AWS::ApiGateway::Stage	
CREATE_IN_PROGRESS	ApiProdStage	AWS::ApiGateway::Stage	
CREATE_COMPLETE	ApiProdStage	AWS::ApiGateway::Stage	

```

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

```

Enfin, un message s'affiche pour vous informer que le déploiement a été effectué avec succès :

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

Utiliser Infrastructure Composer AWS SAM pour supprimer une pile

Cet exemple montre comment supprimer une AWS CloudFormation pile à l'aide de la `aws sam delete` commande.

Entrez la commande `aws sam delete` dans AWS SAM CLI et confirmez si vous souhaitez supprimer la pile et le modèle :

```

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

```

- Deleting S3 object with key e8b8984f881c3732bfb34257cdd58f1e
- Deleting S3 object with key 3185c59b550594ee7fca7f8c36686119.template
- Deleting S3 object with key 30439348c0be6e1b85043b7a935b34ab.template
- Deleting Cloudformation stack aws-app-composer-basic-api

Deleted successfully

AWS Infrastructure Composer résolution des problèmes

Les rubriques de cette section fournissent des conseils sur la résolution des messages d'erreur lors de l'utilisation AWS Infrastructure Composer.

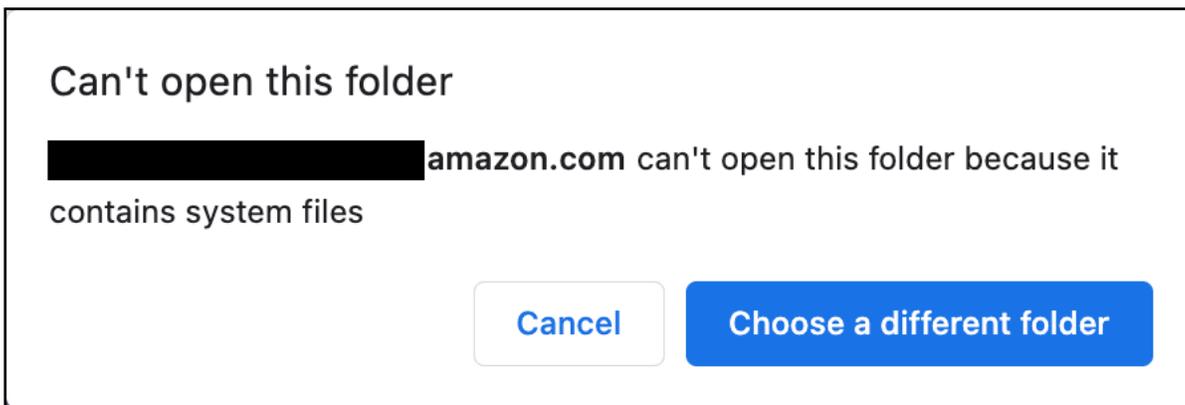
Rubriques

- [Messages d'erreur](#)

Messages d'erreur

« Impossible d'ouvrir ce dossier »

Exemple d'erreur :



Cause possible : Infrastructure Composer ne parvient pas à accéder à un répertoire sensible en mode de synchronisation local.

Pour en savoir plus sur cette erreur, consultez [Data Infrastructure Composer a accès à](#).

Essayez de vous connecter à un autre répertoire local ou d'utiliser Infrastructure Composer avec la synchronisation locale désactivée.

« Modèle incompatible »

Exemple d'erreur : lors du chargement d'un nouveau projet dans Infrastructure Composer, le message suivant s'affiche :

Cause possible : votre projet contient un fichier référencé en externe qui n'est pas pris en charge dans Infrastructure Composer.

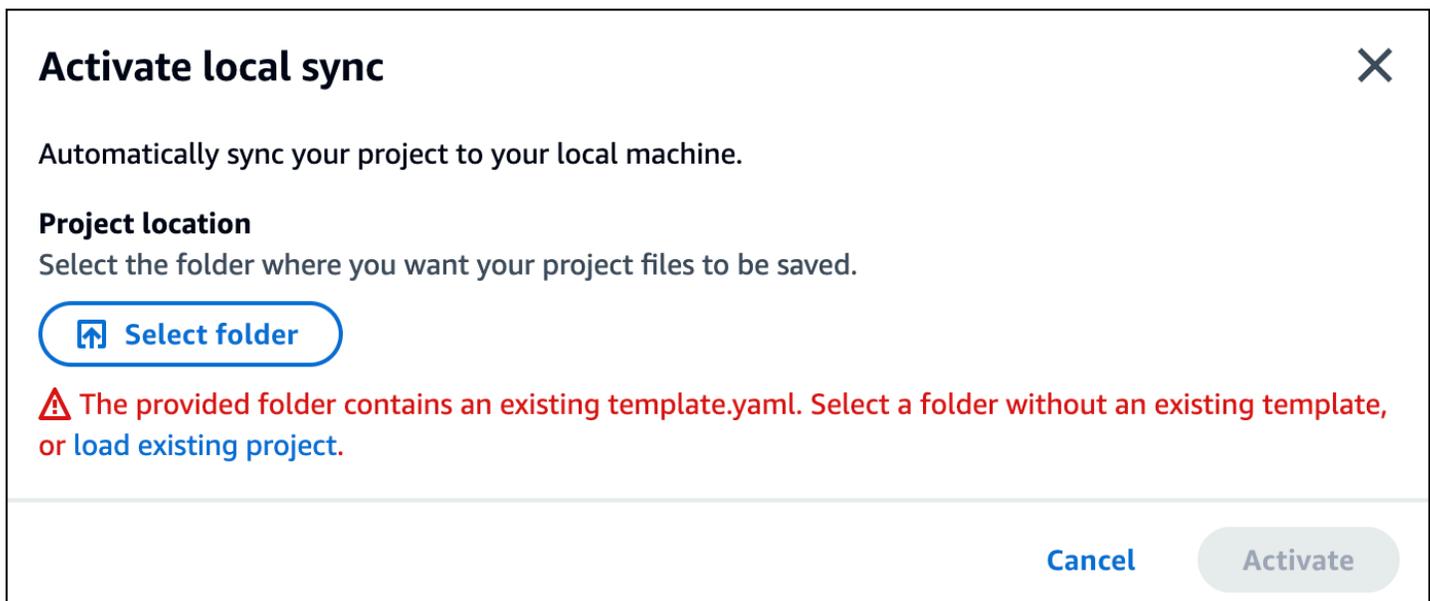
Pour en savoir plus sur les fichiers externes pris en charge dans Infrastructure Composer, consultez [Fichiers externes de référence](#).

Cause possible : votre projet renvoie vers un fichier externe situé dans un autre répertoire local.

Déplacez votre fichier référencé en externe vers un sous-répertoire du répertoire que vous sélectionnez pour utiliser avec le mode de synchronisation local d'Infrastructure Composer.

« Le dossier fourni contient un modèle .yaml existant »

Lorsque vous essayez d'activer la synchronisation locale, le message d'erreur suivant s'affiche :



Cause possible : le dossier sélectionné contient déjà un fichier template.yaml.

Sélectionnez un autre répertoire qui ne contient pas de modèle d'application ou créez un nouveau répertoire.

« Votre navigateur n'est pas autorisé à enregistrer votre projet dans ce dossier... »

Cause possible : Infrastructure Composer ne parvient pas à accéder à un répertoire sensible en mode de synchronisation local.

Pour en savoir plus sur cette erreur, consultez [Data Infrastructure Composer a accès à](#).

Essayez de vous connecter à un autre répertoire local ou utilisez Infrastructure Composer avec la synchronisation locale désactivée.

Sécurité dans AWS Infrastructure Composer

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Infrastructure Composer, voir [AWS Services concernés par programme de conformitéAWS](#) .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Infrastructure Composer. Les rubriques suivantes expliquent comment configurer Infrastructure Composer pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources Infrastructure Composer.

Rubriques

- [Protection des données dans AWS Infrastructure Composer](#)
- [AWS Identity and Access Management pour AWS Infrastructure Composer](#)
- [Validation de conformité pour AWS Infrastructure Composer](#)
- [Résilience dans AWS Infrastructure Composer](#)

Protection des données dans AWS Infrastructure Composer

Le [modèle de responsabilité AWS partagée](#) s'applique à la protection des données dans AWS Infrastructure Composer. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure

mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez la section [Confidentialité des données FAQ](#). Pour plus d'informations sur la protection des données en Europe, consultez le [modèle de responsabilité AWS partagée](#) et le billet de GDPR blog sur le blog sur la AWS sécurité.

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) pour chaque compte.
- Utilisez SSL/TLS pour communiquer avec les AWS ressources. Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Configuration API et journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section [Utilisation des CloudTrail sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de FIPS 140 à 3 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou un API, utilisez un point de terminaison. FIPS Pour plus d'informations sur les FIPS points de terminaison disponibles, voir [Federal Information Processing Standard \(FIPS\) 140-3](#).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec Infrastructure Composer ou autre Services AWS à l'aide de la console API, AWS CLI, ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez un URL à un serveur externe, nous vous recommandons vivement de ne pas inclure d'informations d'identification dans le URL afin de valider votre demande auprès de ce serveur.

Note

Toutes les données que vous entrez dans Infrastructure Composer sont utilisées dans le seul but de fournir des fonctionnalités dans Infrastructure Composer et de générer des fichiers de projet et des répertoires qui sont enregistrés localement sur votre machine. Infrastructure Composer n'enregistre, ne stocke ni ne transmet aucune de ces données.

Chiffrement des données

Infrastructure Composer ne chiffre pas le contenu des clients car les données ne sont ni enregistrées, ni stockées, ni transmises.

Chiffrement au repos

Infrastructure Composer ne chiffre pas le contenu des clients car les données ne sont ni enregistrées, ni stockées, ni transmises.

Chiffrement en transit

Infrastructure Composer ne chiffre pas le contenu des clients car les données ne sont ni enregistrées, ni stockées, ni transmises.

Gestion des clés

Infrastructure Composer ne prend pas en charge la gestion des clés car le contenu client n'est ni enregistré, ni stocké, ni transmis.

Confidentialité du trafic inter-réseaux

Infrastructure Composer ne génère pas de trafic avec les clients et applications sur site.

AWS Identity and Access Management pour AWS Infrastructure Composer

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. IAM les administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources

d'Infrastructure Composer. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion des accès à l'aide de politiques](#)
- [Comment AWS Infrastructure Composer fonctionne avec IAM](#)

Public ciblé

Infrastructure Composer nécessite, au minimum, un accès en lecture seule au AWS Management Console. Tout utilisateur disposant de cette autorisation peut utiliser toutes les fonctionnalités d'Infrastructure Composer. L'accès granulaire à des fonctionnalités spécifiques d'Infrastructure Composer n'est pas pris en charge.

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS à l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant que Utilisateur racine d'un compte AWS, en tant qu'IAM utilisateur ou en assumant un IAM rôle.

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez en tant qu'identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS à l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section [Comment vous connecter à votre compte Compte AWS dans](#) le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide

de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la [version 4 de AWS Signature pour les API demandes](#) dans le guide de IAM l'utilisateur.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, voir [Authentification multifactorielle](#) dans le guide de l'AWS IAM Identity Center utilisateur et [Authentification AWS multifactorielle IAM dans](#) le guide de l'IAMutilisateur.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui nécessitent que vous vous connectiez en tant qu'utilisateur root, consultez la section [Tâches nécessitant des informations d'identification utilisateur root](#) dans le guide de IAM l'utilisateur.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour

plus d'informations sur IAM Identity Center, consultez [Qu'est-ce qu'IAM Identity Center ?](#) dans le guide de AWS IAM Identity Center l'utilisateur.

Utilisateurs et groupes IAM

Un [IAM utilisateur](#) est une identité au sein de votre Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des IAM utilisateurs dotés d'informations d'identification à long terme, telles que des mots de passe et des clés d'accès. Toutefois, si vous avez des cas d'utilisation spécifiques qui nécessitent des informations d'identification à long terme auprès des IAM utilisateurs, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme](#) dans le Guide de IAM l'utilisateur.

Un [groupe IAM](#) est une identité qui spécifie un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAM Adminset lui donner les autorisations nécessaires pour administrer IAM des ressources.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez la section [Cas d'utilisation pour IAM les utilisateurs](#) dans le Guide de IAM l'utilisateur.

Rôles IAM

Un [IAM rôle](#) est une identité au sein de votre Compte AWS vous dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais un rôle n'est pas associé à une personne en particulier. Pour assumer temporairement un IAM rôle dans le AWS Management Console, vous pouvez [passer d'un utilisateur à un IAM rôle \(console\)](#). Vous pouvez assumer un rôle en appelant une AWS API opération AWS CLI or ou en utilisant une option personnalisée URL. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section [Méthodes pour assumer un rôle](#) dans le Guide de IAM l'utilisateur.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- **Accès utilisateur fédéré** : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour plus d'informations sur les rôles pour la fédération, voir [Créer un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un ensemble d'autorisations. Pour contrôler les accès auxquels vos identités peuvent accéder après leur authentification, IAM Identity Center met en corrélation l'ensemble d'autorisations avec un rôle dans IAM. Pour plus d'informations sur les jeux d'autorisations, consultez [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .
- **Autorisations IAM utilisateur temporaires** : un IAM utilisateur ou un rôle peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- **Accès intercompte** : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (un principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, voir [Accès aux ressources entre comptes IAM dans le guide de l'IAMutilisateur](#).
- **Accès multiservices** — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
- **Sessions d'accès transmises (FAS)** — Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche ensuite une autre action dans un autre service. FAS utilise les autorisations du principal appelant au Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FAS les demandes ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, voir [Transférer les sessions d'accès](#).
- **Rôle de service** — Un rôle de service est un [IAMrôle](#) qu'un service assume pour effectuer des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction

du service à partir de IAM. Pour plus d'informations, consultez la section [Créer un rôle pour déléguer des autorisations à un Service AWS](#) dans le guide de IAM l'utilisateur.

- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier les autorisations concernant les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui AWS CLI soumettent des AWS API demandes. Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section [Utiliser un IAM rôle pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon](#) dans le Guide de IAM l'utilisateur.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de JSON documents. Pour plus d'informations sur la structure et le contenu des documents de JSON politique, voir [Présentation des JSON politiques](#) dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

Les stratégies IAM définissent les autorisations d'une action quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle auprès du AWS Management Console AWS CLI, ou du AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir [Définir des IAM autorisations personnalisées avec des politiques gérées par le client](#) dans le Guide de l'IAMutilisateur.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour savoir comment choisir entre une politique gérée ou une politique intégrée, voir [Choisir entre les politiques gérées et les politiques intégrées dans le Guide](#) de l'IAMutilisateur.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser de politiques AWS gérées depuis une IAM stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Amazon S3 et Amazon VPC sont des exemples de services compatibles ACLs. AWS WAF Pour en savoir plus ACLs, consultez la [présentation de la liste de contrôle d'accès \(ACL\)](#) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- **Limites d'autorisations** — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (IAM utilisateur ou rôle). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez la section Limites d'[autorisations pour les IAM entités](#) dans le Guide de IAM l'utilisateur.
- **Politiques de contrôle des services (SCPs)** : SCPs JSON politiques qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Les SCP limites d'autorisations pour les entités présentes dans les comptes des membres, y compris chacune d'entre elles Utilisateur racine d'un compte AWS. Pour plus d'informations sur les Organizations et consultez SCPs les [politiques de contrôle des services](#) dans le Guide de AWS Organizations l'utilisateur.
- **Politiques de contrôle des ressources (RCPs)** : RCPs JSON politiques que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les IAM politiques associées à chaque ressource que vous possédez. Cela RCP limite les autorisations pour les ressources dans les comptes des membres et peut avoir un

impact sur les autorisations effectives pour les identités Utilisateur racine d'un compte AWS, y compris, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les OrganizationsRCPs, y compris une liste de ces Services AWS supportsRCPs, consultez la section [Resource control policies \(RCPs\)](#) dans le guide de AWS Organizations l'utilisateur.

- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour en savoir plus, consultez [Politiques de session](#) dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de IAM l'utilisateur.

Comment AWS Infrastructure Composer fonctionne avec IAM

AWS Infrastructure Composer nécessite, au minimum, un accès en lecture seule au. AWS Management Console Tout utilisateur disposant de cette autorisation peut utiliser toutes les fonctionnalités d'Infrastructure Composer. L'accès granulaire à des fonctionnalités spécifiques d'Infrastructure Composer n'est pas pris en charge.

Lorsque vous déployez votre modèle de projet et vos fichiers sur AWS CloudFormation, vous devez disposer des autorisations nécessaires. Pour en savoir plus, veuillez consulter la rubrique [Contrôle de l'accès avec AWS Identity and Access Management](#) dans le Guide de l'utilisateur AWS CloudFormation .

Le tableau suivant indique les IAM fonctionnalités qui peuvent être utilisées AWS Infrastructure Composer.

Fonctionnalité IAM	Support pour Infrastructure Composer
Politiques basées sur l'identité	Non

Fonctionnalité IAM	Support pour Infrastructure Composer
Politiques basées sur les ressources	Non
Actions de politique	Non
Ressources de politique	Non
Clés de condition d'une politique	Non
ACLs	Non
ABAC(balises dans les politiques)	Non
Informations d'identification temporaires	Oui
Autorisations de principaux	Non
Fonctions du service	Non
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont Infrastructure Composer et les autres AWS services fonctionnent avec la plupart des IAM fonctionnalités, consultez la section [AWS Services compatibles IAM](#) dans le Guide de IAM l'utilisateur.

Politiques basées sur l'identité pour Infrastructure Composer

Prend en charge les politiques basées sur l'identité : Non

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir [Définir des IAM autorisations personnalisées avec des politiques gérées par le client](#) dans le Guide de l'IAMutilisateur.

Avec les stratégies IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité, car

celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour en savoir plus sur tous les éléments que vous pouvez utiliser dans une JSON politique, consultez la [référence aux éléments de IAM JSON politique](#) dans le Guide de IAM l'utilisateur.

Politiques basées sur les ressources dans Infrastructure Composer

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès entre comptes, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que mandataire dans une stratégie basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un IAM administrateur du compte de confiance doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, voir [Accès aux ressources entre comptes IAM dans](#) le Guide de IAM l'utilisateur.

Actions politiques pour Infrastructure Composer

Soutient les actions politiques : Non

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'Actionélément d'une JSON politique décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès dans une politique. Les actions de stratégie portent généralement le même nom que l' AWS APIopération associée. Il existe certaines exceptions, telles que les actions avec autorisation

uniquement qui n'ont pas d'opération correspondante. API Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions d'Infrastructure Composer, consultez la section [Actions définies par AWS Infrastructure Composer](#) dans la référence d'autorisation de service.

Ressources relatives aux politiques pour Infrastructure Composer

Prend en charge les ressources politiques : Non

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Resource` JSON de stratégie indique le ou les objets auxquels s'applique l'action. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de spécifier une ressource en utilisant son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de ressources Infrastructure Composer et leurs caractéristiques ARNs, consultez la section [Ressources définies par AWS Infrastructure Composer](#) dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier pour chaque ressource, voir [Actions définies par AWS Infrastructure Composer](#). ARN

Clés de conditions de politique pour Infrastructure Composer

Prend en charge les clés de condition de politique spécifiques au service : Non

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#), tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une OR opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM . Pour plus d'informations, consultez [IAM la section Éléments de politique : variables et balises](#) dans le Guide de IAM l'utilisateur.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les [clés contextuelles de condition AWS globales](#) dans le guide de IAM l'utilisateur.

Pour consulter la liste des clés de condition d'Infrastructure Composer, consultez la section [Clés de condition pour AWS Infrastructure Composer](#) dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, voir [Actions définies par AWS Infrastructure Composer](#).

ACLs dans Infrastructure Composer

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

ABAC avec Infrastructure Composer

Supports ABAC (balises dans les politiques) : Non

Le contrôle d'accès basé sur les attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises à IAM des entités (utilisateurs ou rôles) et à de nombreuses AWS ressources. Le balisage des entités et des ressources est la première étape de ABAC. Vous concevez ensuite des ABAC politiques pour autoriser les opérations lorsque le tag du principal correspond à celui de la ressource à laquelle il essaie d'accéder.

ABAC est utile dans les environnements qui se développent rapidement et aide dans les situations où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'[élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations ABAC, voir [Définir des autorisations avec ABAC autorisation](#) dans le Guide de IAM l'utilisateur. Pour consulter un didacticiel présentant les étapes de configuration ABAC, voir [Utiliser le contrôle d'accès basé sur les attributs \(ABAC\)](#) dans le guide de l'IAM utilisateur.

Utilisation d'informations d'identification temporaires avec Infrastructure Composer

Prend en charge les informations d'identification temporaires : oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section [relative à l'utilisation IAM](#) dans le Guide de IAM l'utilisateur.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, voir [Passer d'un utilisateur à un IAM rôle \(console\)](#) dans le guide de IAM l'utilisateur.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide du AWS CLI ou AWS API. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez la section Informations [d'identification de sécurité temporaires dans IAM](#).

Vous pouvez utiliser des informations d'identification temporaires pour accéder à Infrastructure Composer via le AWS Management Console. Par exemple, consultez la section [Activation de l'accès à la AWS console par un courtier d'identité personnalisé](#) dans le Guide de IAM l'utilisateur.

Autorisations principales interservices pour Infrastructure Composer

Prend en charge les sessions d'accès transféré (FAS) : Non

Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche ensuite une autre action dans un autre service. FAS utilise les autorisations du principal appelant au Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FAS les demandes ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, voir [Transférer les sessions d'accès](#).

Rôles de service pour Infrastructure Composer

Prend en charge les rôles de service : Non

Un rôle de service est un [IAM rôle](#) qu'un service assume pour effectuer des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir de IAM. Pour plus d'informations, consultez la section [Créer un rôle pour déléguer des autorisations à un Service AWS](#) dans le guide de IAM l'utilisateur.

Warning

La modification des autorisations pour un rôle de service peut perturber les fonctionnalités d'Infrastructure Composer. Modifiez les rôles de service uniquement lorsque Infrastructure Composer fournit des instructions à cet effet.

Rôles liés aux services pour Infrastructure Composer

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier les autorisations concernant les rôles liés à un service.

Pour plus de détails sur la création ou la gestion des rôles liés à un service, consultez la section [AWS Services compatibles avec](#). IAM Recherchez un service dans le tableau qui inclut un Yes dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Validation de conformité pour AWS Infrastructure Composer

Pour savoir si un [programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité](#) spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- [Conformité et gouvernance de la sécurité](#) : ces guides de mise en œuvre de solutions traitent des considérations architecturales et fournissent les étapes à suivre afin de déployer des fonctionnalités de sécurité et de conformité.
- [Référence des services éligibles à la HIPAA — Répertoire les services éligibles](#) à la HIPAA. Tous ne Services AWS sont pas éligibles à la loi HIPAA.
- AWS Ressources de <https://aws.amazon.com/compliance/resources/> de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.

- [AWS Guides de conformité destinés aux clients](#) — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- [Évaluation des ressources à l'aide des règles](#) du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#)— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez [Référence des contrôles Security Hub](#).
- [Amazon GuardDuty](#) — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- [AWS Audit Manager](#)— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dans AWS Infrastructure Composer

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section [Infrastructure AWS globale](#).

Toutes les données que vous entrez dans Infrastructure Composer sont utilisées dans le seul but de fournir des fonctionnalités dans Infrastructure Composer et de générer des fichiers de projet et des répertoires qui sont enregistrés localement sur votre machine. Infrastructure Composer n'enregistre ni ne stocke aucune de ces données.

Historique du document pour Infrastructure Composer

Le tableau suivant décrit les publications de documentation importantes pour Infrastructure Composer. Pour recevoir des notifications concernant les mises à jour de cette documentation, vous pouvez vous abonner à un RSS flux.

- Dernière mise à jour de la documentation : 30 novembre 2023

Modification	Description	Date
Contenu restructuré et mis à jour dans l'ensemble du guide du développeur	Réorganisation et restructuration du guide pour améliorer la découvrabilité et la convivialité. Titres mis à jour et améliorés. A fourni des détails supplémentaires lors de l'introduction des sujets et des concepts.	1er août 2024
Ajout de documentation sur l'utilisation d'Infrastructure Composer en mode CloudFormation console et restructuration du guide du développeur d'Infrastructure Composer.	AWS Infrastructure Composer peut désormais être utilisé en mode AWS CloudFormation console. Pour en savoir plus, consultez la section Utilisation d'Infrastructure Composer en mode CloudFormation console . En outre, une grande partie du contenu du guide de l'utilisateur a été réorganisée afin de créer une expérience rationalisée.	28 mars 2024
Documentation ajoutée pour l'intégration d'Infrastructure Composer avec CodeWhisperer	AWS Infrastructure Composer du Toolkit for VS Code Code fournit une intégration avec Amazon CodeWhisperer.	30 novembre 2023

	<p>Pour en savoir plus, consultez Utiliser AWS Infrastructure Composer avec Amazon CodeWhisperer.</p>	
<p>Ajout de documentation pour le déploiement de votre application avec Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code</p>	<p>Utilisez le bouton de synchronisation depuis le canevas d'Infrastructure Composer pour déployer votre application sur le AWS Cloud. Pour en savoir plus, voir Déployer votre application avec Sam Sync.</p>	30 novembre 2023
<p>Ajout de la documentation pour Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code</p>	<p>Vous pouvez désormais utiliser Infrastructure Composer depuis VS Code avec le AWS Toolkit for Visual Studio Code. Pour en savoir plus, consultez la section Utilisation AWS Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code.</p>	30 novembre 2023
<p>Ajout de l'intégration de Step Functions Workflow Studio</p>	<p>Lancez Step Functions Workflow Studio depuis le canevas d'Infrastructure Composer. Pour en savoir plus, consultez la section Utilisation AWS Infrastructure Composer avec AWS Step Functions.</p>	27 novembre 2023

Ajout de la console Lambda et de l'intégration d'Infrastructure Composer	Lancez le canevas Infrastructure Composer depuis la console Lambda. Pour en savoir plus, consultez la section Utilisation AWS Infrastructure Composer avec la AWS Lambda console .	14 novembre 2023
Ajout d'Amazon VPC en tant que service vedette avec Infrastructure Composer	Infrastructure Composer introduit une VPC balise pour visualiser les ressources configurées avec un VPC. Vous pouvez également configurer les fonctions Lambda VPCs définies sur un modèle externe. Pour en savoir plus, consultez Utilisation d'Infrastructure Composer avec Amazon VPC .	17 octobre 2023
Ajout d'Amazon RDS en tant que service vedette avec Infrastructure Composer	Connectez votre application Infrastructure Composer à un RDS cluster ou à une instance de base de données Amazon défini sur un modèle externe. Pour en savoir plus, consultez Utilisation d'Infrastructure Composer avec Amazon RDS .	17 octobre 2023
Ajout du support Infrastructure Composer pour concevoir avec toutes les AWS CloudFormation ressources	Sélectionnez n'importe quelle AWS CloudFormation ressource dans la palette Ressources pour concevoir vos applications. Pour en savoir plus, voir Utiliser n'importe quelle AWS CloudFormation ressource .	26 septembre 2023

Ajout de documentation pour les cartes dans Infrastructure Composer	Infrastructure Composer prend en charge plusieurs types de cartes que vous pouvez utiliser pour concevoir et développer votre application. Pour en savoir plus, consultez la section Conception à l'aide de cartes dans Infrastructure Composer .	20 septembre 2023
Ajout de documentation pour la fonction d'annulation et de restauration	Utilisez les boutons Annuler et Rétablir sur le canevas d'Infrastructure Composer. Pour en savoir plus, consultez la section Annuler et rétablir .	1er août 2023
Documentation ajoutée pour le mode de synchronisation locale	Utilisez le mode de synchronisation locale pour synchroniser et enregistrer automatiquement votre projet sur votre machine locale. Pour en savoir plus, consultez la section Mode de synchronisation locale .	1er août 2023
Ajout de documentation pour la fonctionnalité d'exportation du canevas	Utilisez la fonction d'exportation du canevas pour exporter le canevas de votre application sous forme d'image vers votre machine locale. Pour en savoir plus, consultez la section Exporter le canevas .	1er août 2023

Support d'Infrastructure Composer pour les références de fichiers externes	Référez les fichiers externes pour les ressources prises en charge dans Infrastructure Composer. Pour en savoir plus, consultez la section Utilisation de modèles faisant référence à des fichiers externes .	17 mai 2023
Nouvelle documentation sur la connexion des ressources	Connectez les ressources entre elles pour définir des relations basées sur les événements entre les ressources de votre application. Pour en savoir plus, voir Connecter des ressources entre elles à l'aide du canevas visuel Infrastructure Composer .	7 mars 2023
Nouvelle fonctionnalité Change Inspector	Utilisez le Change Inspector pour consulter les mises à jour du code de votre modèle et découvrir ce qu'Infrastructure Composer est en train de créer pour vous. Pour en savoir plus, voir Afficher les mises à jour du code avec le Change Inspector .	7 mars 2023

[Infrastructure Composer est désormais disponible pour tous](#)

AWS Infrastructure Composer est désormais disponible pour tous. Pour en savoir plus, consultez [AWS Infrastructure Composer désormais la rubrique « Disponible en général » - Créez rapidement et visuellement des applications sans serveur.](#)

7 mars 2023

[Plus d'informations sur les avantages de l'utilisation du mode connecté](#)

Utilisez Infrastructure Composer en mode connecté avec votre local IDE pour accélérer le développement. Pour en savoir plus, consultez la section [Utilisation d'Infrastructure Composer avec votre local IDE.](#)

7 mars 2023

[Rubrique mise à jour sur l'utilisation d'autres AWS services pour déployer votre application](#)

Utilisez Infrastructure Composer pour concevoir des applications sans serveur prêtes à être déployées. AWS SAM À utiliser pour déployer votre application sans serveur. Pour en savoir plus, consultez la section [Utilisation d'Infrastructure Composer avec AWS CloudFormation et AWS SAM.](#)

3 mars 2023

[Ajout d'une section sur les concepts sans serveur](#)

Découvrez les concepts de base du mode sans serveur avant d'utiliser Infrastructure Composer. Pour en savoir plus, consultez la section [Concepts sans serveur.](#)

2 mars 2023

[Publication publique](#)

Première version publique
d'Infrastructure Composer.

1er décembre 2022

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.