



Entwicklerhandbuch

# AWS-Infrastruktur-Composer



# AWS-Infrastruktur-Composer: Entwicklerhandbuch

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

---

# Table of Contents

Was ist Infrastructure Composer? .....	1
Stellen Sie Ihre Architektur zusammen .....	2
Definieren Sie Ihre Vorlagen .....	4
Integrieren Sie es in Ihre Workflows .....	5
Möglichkeiten, auf Infrastructure Composer zuzugreifen .....	6
Weitere Informationen .....	8
Nächste Schritte .....	8
Serverlose Konzepte .....	8
Serverlose Konzepte .....	9
Karten .....	10
Verbesserte Komponentenkarten .....	11
Beispiel .....	12
Standardkomponentenkarten .....	13
Kartenverbindungen .....	17
Verbindungen zwischen Karten .....	17
Verbindungen zwischen erweiterten Komponentenkarten .....	18
Verbindungen zu und von Standard-IaC-Ressourcenkarten .....	20
Erste Schritte .....	21
Machen Sie einen Rundgang durch die Konsole .....	21
Nächste Schritte .....	22
Laden und ändern .....	22
Schritt 1: Öffnen Sie die Demo .....	23
Schritt 2: Erkunden Sie die visuelle Leinwand .....	23
Schritt 3: Erweitern Sie Ihre Architektur .....	27
Schritt 4: Speichern Sie Ihre Bewerbung .....	28
Nächste Schritte .....	29
Entwicklung .....	29
ResourceProperties .....	30
Schritt 1: Erstelle dein Projekt .....	30
Fügen Sie Karten hinzu .....	33
Schritt 3: Konfigurieren Sie Ihre REST-API .....	34
Schritt 4: Konfigurieren Sie Ihre Funktionen .....	35
Schritt 5: Connect deine Karten .....	36
Schritt 6: Organisieren Sie die Leinwand .....	37

Eine DynamoDB-Tabelle hinzufügen .....	38
Schritt 8: Überprüfe deine Vorlage .....	39
Schritt 9: Integrieren Sie es in Ihre Workflows .....	40
Nächste Schritte .....	40
Wo kann Infrastructure Composer verwendet werden .....	41
Infrastructure Composer-Konsole .....	41
Visueller Überblick .....	42
Verwalten Sie Ihr Projekt .....	45
Connect zu Ihrer lokalen IDE her .....	49
Zugriff auf Webseiten zulassen .....	51
Lokal synchronisieren und speichern .....	53
Aus der Lambda-Konsole importieren .....	57
Leinwand exportieren .....	57
CloudFormation Konsolenmodus .....	59
Warum sollte dieser Modus verwendet werden? .....	59
Greifen Sie auf diesen Modus zu .....	60
Visualisieren Sie eine Bereitstellung .....	61
Erstellen Sie eine neue Vorlage .....	61
Aktualisieren Sie einen vorhandenen Stack .....	63
AWS Toolkit for Visual Studio Code .....	65
Visueller Überblick .....	65
Zugriff über VS Code .....	67
Synchronisieren mit AWS Cloud .....	68
Infrastructure Composer mit Amazon Q .....	70
Wie komponiert man .....	73
Platzieren Sie Karten auf der Leinwand .....	73
Gruppieren Sie Karten zusammen .....	74
Gruppieren erweiterter Komponentenkarten .....	75
Gruppieren einer Standardkomponentenkarte in eine andere .....	75
Karten Connect .....	77
Anschließen erweiterter Komponentenkarten .....	77
Standardkarten anschließen .....	78
Beispiele .....	80
Trennen Sie die Karten .....	82
Verbesserte Komponentenkarten .....	82
Standardkomponentenkarten .....	83

Karten anordnen .....	85
Karten konfigurieren und ändern .....	85
Verbesserte Karten .....	86
Standardkarten .....	102
Karten löschen .....	103
Verbesserte Komponentenkarten .....	104
Standardkomponentenkarten .....	104
Codeaktualisierungen anzeigen .....	104
Vorteile des Change Inspector .....	105
Verfahren .....	106
Weitere Informationen .....	107
Verweisen Sie auf externe Dateien .....	108
Bewährte Methoden .....	109
Erstellen Sie eine externe Dateireferenz .....	109
Laden Sie ein Projekt .....	110
Erstellen Sie eine Anwendung mit AWS SAM CLI .....	111
Verweisen Sie auf OpenAPI Spezifikation .....	115
Integrieren Sie mit Amazon VPC .....	118
Identifizieren Sie Ressourcen und Informationen .....	119
Funktionen konfigurieren .....	125
Parameter in importierten Vorlagen .....	125
Hinzufügen neuer Parameter zu importierten Vorlagen .....	128
Konfigurieren Sie eine Lambda-Funktion mit einer VPC in einer anderen Vorlage .....	129
Stellen Sie sie in der Cloud AWS bereit .....	133
Wichtige Konzepte AWS SAM .....	133
Nächste Schritte .....	133
Richten Sie das ein AWS SAM CLI .....	134
Installieren Sie das AWS CLI .....	134
Installieren Sie den AWS SAM CLI .....	134
Greifen Sie auf AWS SAM CLI .....	134
Nächste Schritte .....	135
Erstellen und Bereitstellen .....	135
Löschen eines Stacks .....	143
Fehlerbehebung .....	145
Fehlermeldungen .....	145
„Dieser Ordner kann nicht geöffnet werden“ .....	145

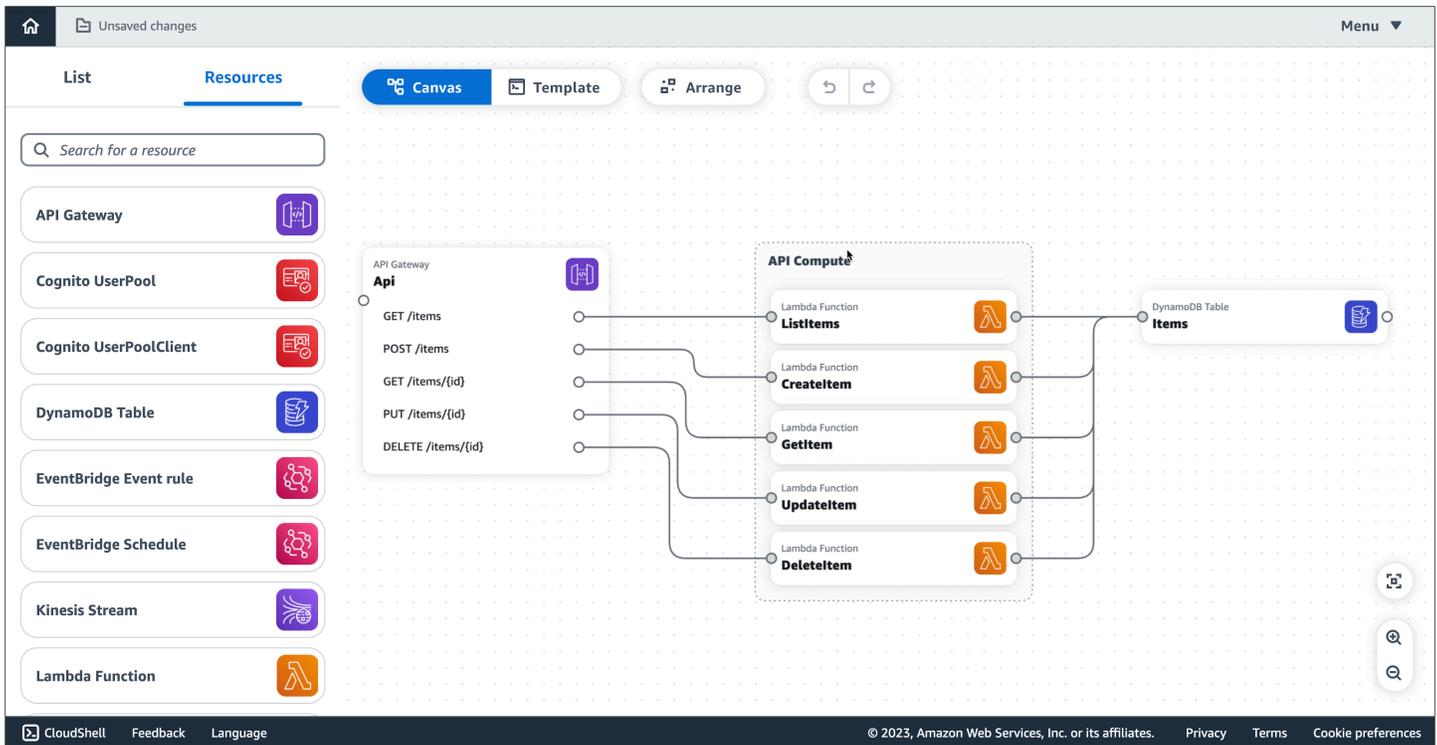
---

„Inkompatible Vorlage“ .....	145
„Der bereitgestellte Ordner enthält eine vorhandene Datei template.yaml“ .....	146
„Ihr Browser ist nicht berechtigt, Ihr Projekt in diesem Ordner zu speichern...“ .....	147
Sicherheit .....	148
Datenschutz .....	148
Datenverschlüsselung .....	150
Verschlüsselung während der Übertragung .....	150
Schlüsselverwaltung .....	150
Datenschutz für den Datenverkehr zwischen Netzwerken .....	150
AWS Identity and Access Management .....	150
Zielgruppe .....	151
Authentifizierung mit Identitäten .....	151
Verwalten des Zugriffs mit Richtlinien .....	155
Wie AWS-Infrastruktur-Composer funktioniert mit IAM .....	158
Compliance-Validierung .....	165
Ausfallsicherheit .....	166
Dokumentverlauf .....	167
.....	clxxiv

# Was ist AWS-Infrastruktur-Composer?

AWS-Infrastruktur-Composer ermöglicht es Ihnen, moderne Anwendungen visuell zu erstellen. AWS Insbesondere können Sie Infrastructure Composer verwenden, um moderne Anwendungen aus allen AWS Diensten, die von unterstützt werden, zu visualisieren, zu erstellen und bereitzustellen, AWS CloudFormation ohne dass Sie dafür Experte sein müssen. AWS CloudFormation

Bei der Zusammenstellung Ihrer AWS CloudFormation Infrastruktur erstellt Infrastructure Composer über eine ansprechende drag-and-drop Oberfläche Ihre IaC-Vorlagen (Infrastructure as Code) und befolgt dabei die AWS bewährten Methoden. Die folgende Abbildung zeigt, wie einfach es ist, Ressourcen auf der visuellen Leinwand von Infrastructure Composer per Drag-and-Drop zu konfigurieren und zu verbinden.



Infrastructure Composer kann über die Infrastructure Composer-Konsole AWS Toolkit for Visual Studio Code, den und im CloudFormation Konsolenmodus verwendet werden.

## Themen

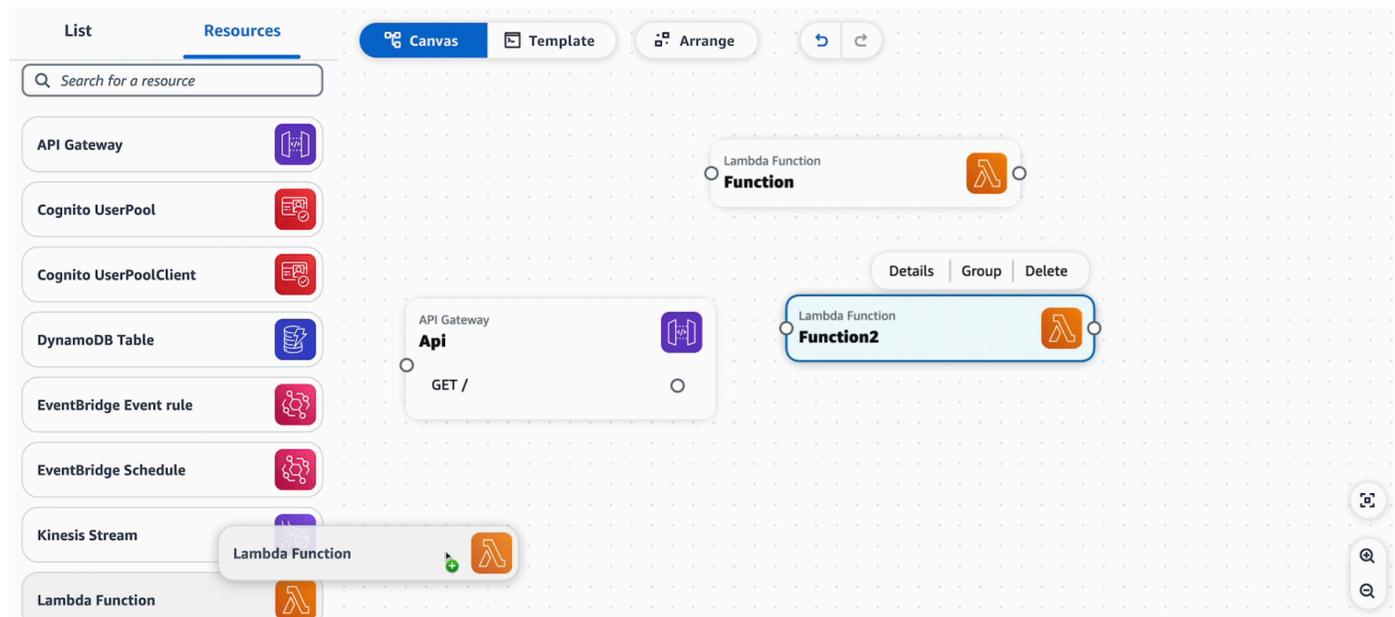
- [Stellen Sie Ihre Anwendungsarchitektur zusammen](#)
- [Definieren Sie Ihre Infrastruktur als Codevorlagen \(IaC\)](#)
- [Integrieren Sie es in Ihre bestehenden Workflows](#)

- [Möglichkeiten, auf Infrastructure Composer zuzugreifen](#)
- [Weitere Informationen](#)
- [Nächste Schritte](#)
- [Serverlose Konzepte für AWS-Infrastruktur-Composer](#)

## Stellen Sie Ihre Anwendungsarchitektur zusammen

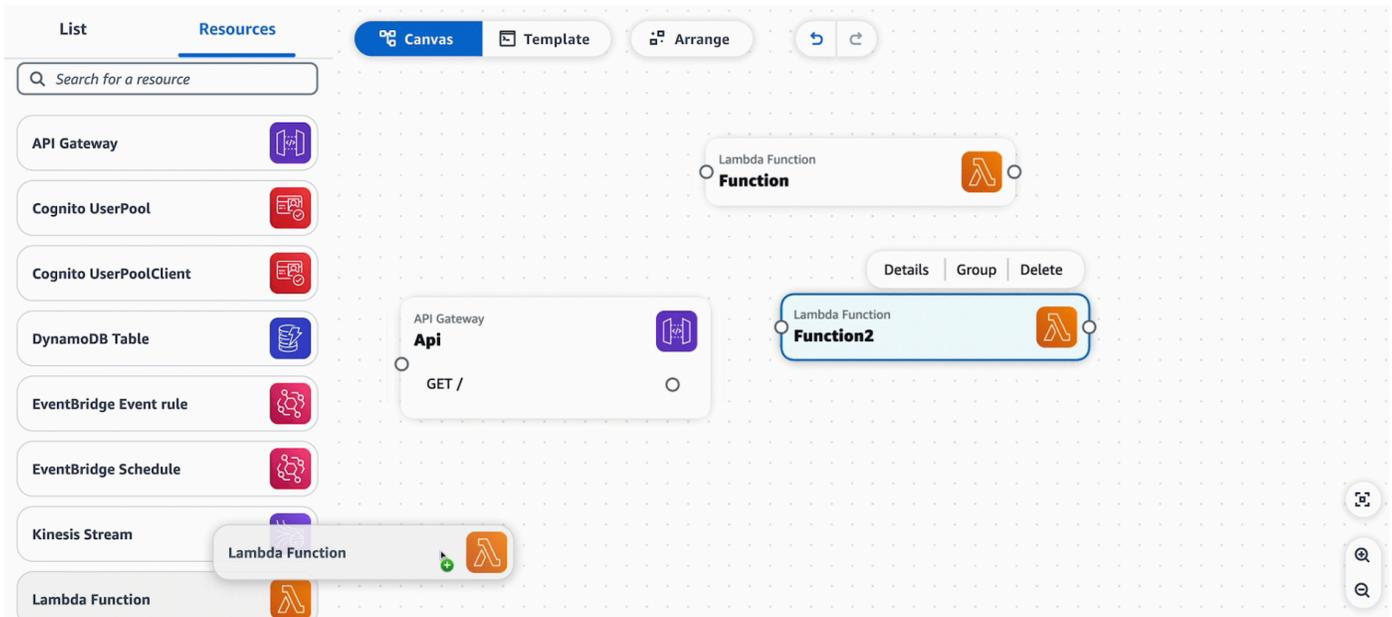
### Baue mit Karten

Platzieren Sie Karten auf der Infrastructure Composer-Arbeitsfläche, um Ihre Anwendungsarchitektur zu visualisieren und zu erstellen.



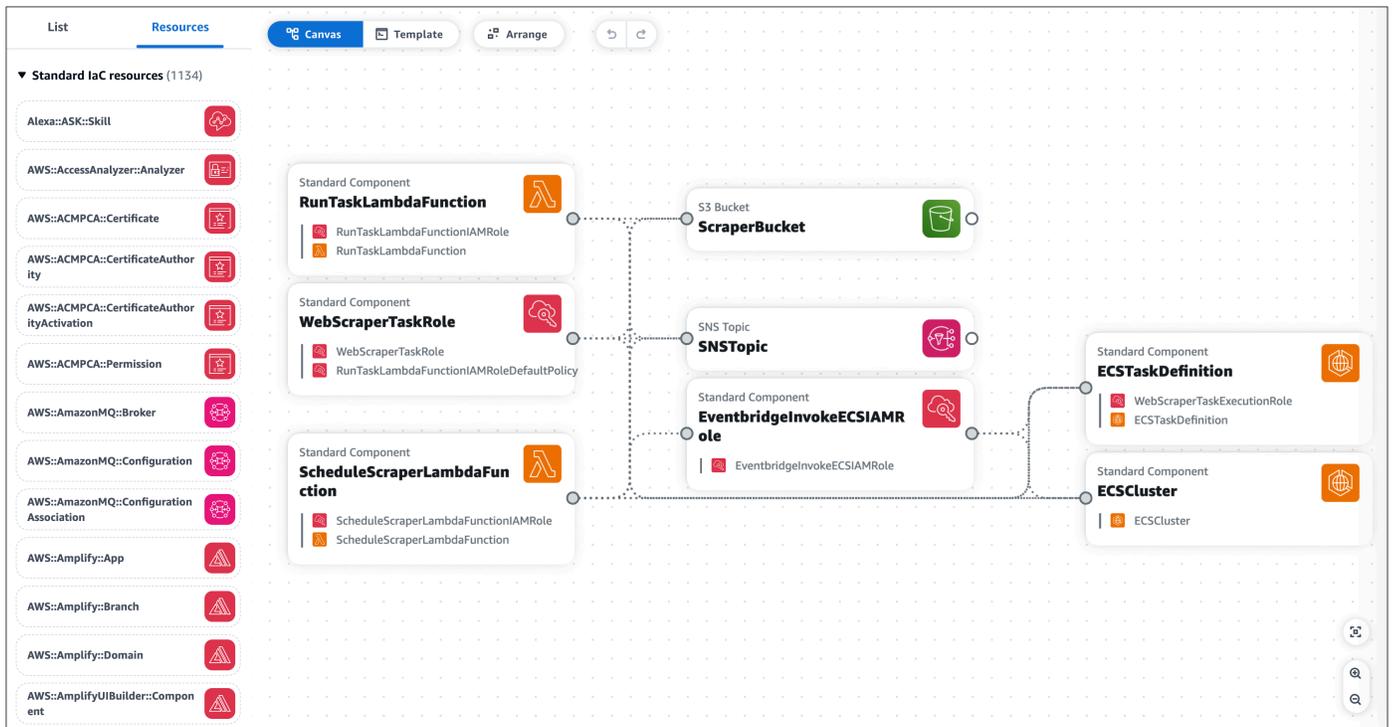
### Karten miteinander verbinden

Konfigurieren Sie, wie Ihre Ressourcen miteinander interagieren, indem Sie sie visuell miteinander verbinden. Spezifizieren Sie ihre Eigenschaften in einem kuratierten Eigenschaftenfenster weiter.



Arbeiten Sie mit einer beliebigen AWS CloudFormation Ressource

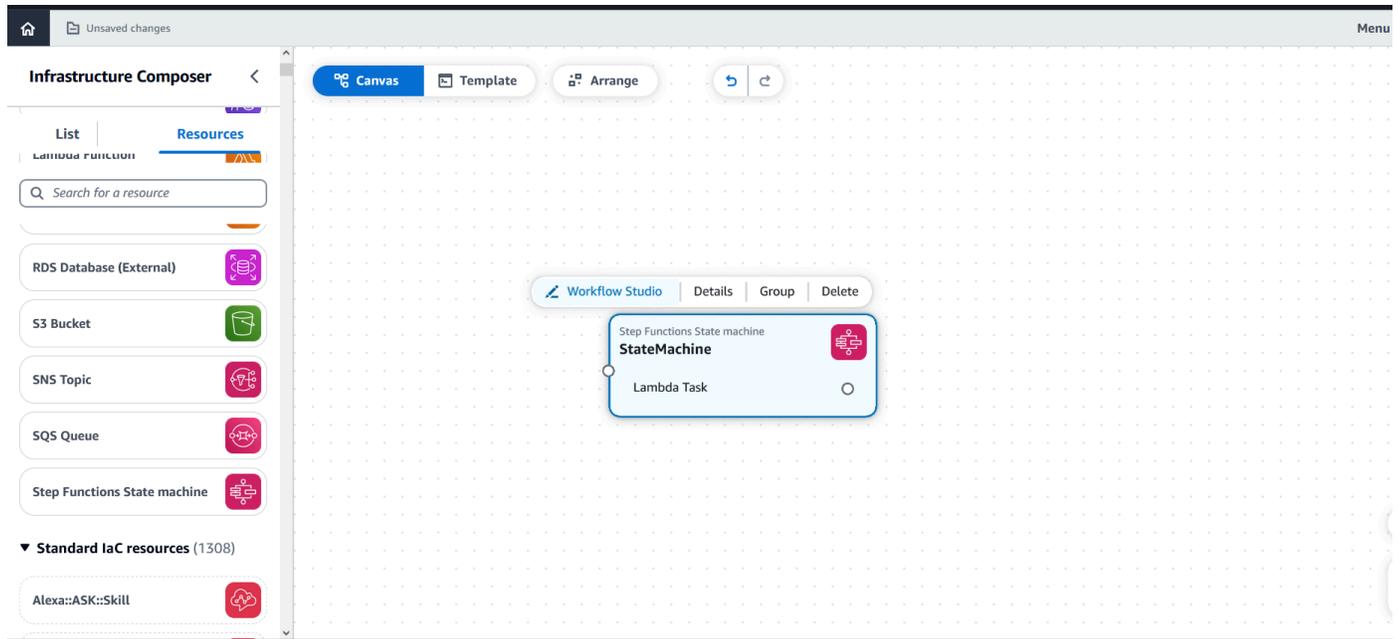
Ziehen Sie eine beliebige AWS CloudFormation Ressource auf die Arbeitsfläche, um Ihre Anwendungsarchitektur zusammenzustellen. Infrastructure Composer bietet eine IaC-Startvorlage, mit der Sie die Eigenschaften Ihrer Ressource angeben können. Weitere Informationen hierzu finden Sie unter [Karten in Infrastructure Composer konfigurieren und ändern](#).



## Greifen Sie auf zusätzliche Funktionen mit folgenden Funktionen zu AWS-Services

Infrastructure Composer-Funktionen AWS-Services , die bei der Erstellung von Anwendungen häufig verwendet oder zusammen konfiguriert werden. Weitere Informationen hierzu finden Sie unter [Integrieren Sie mit Amazon VPC](#).

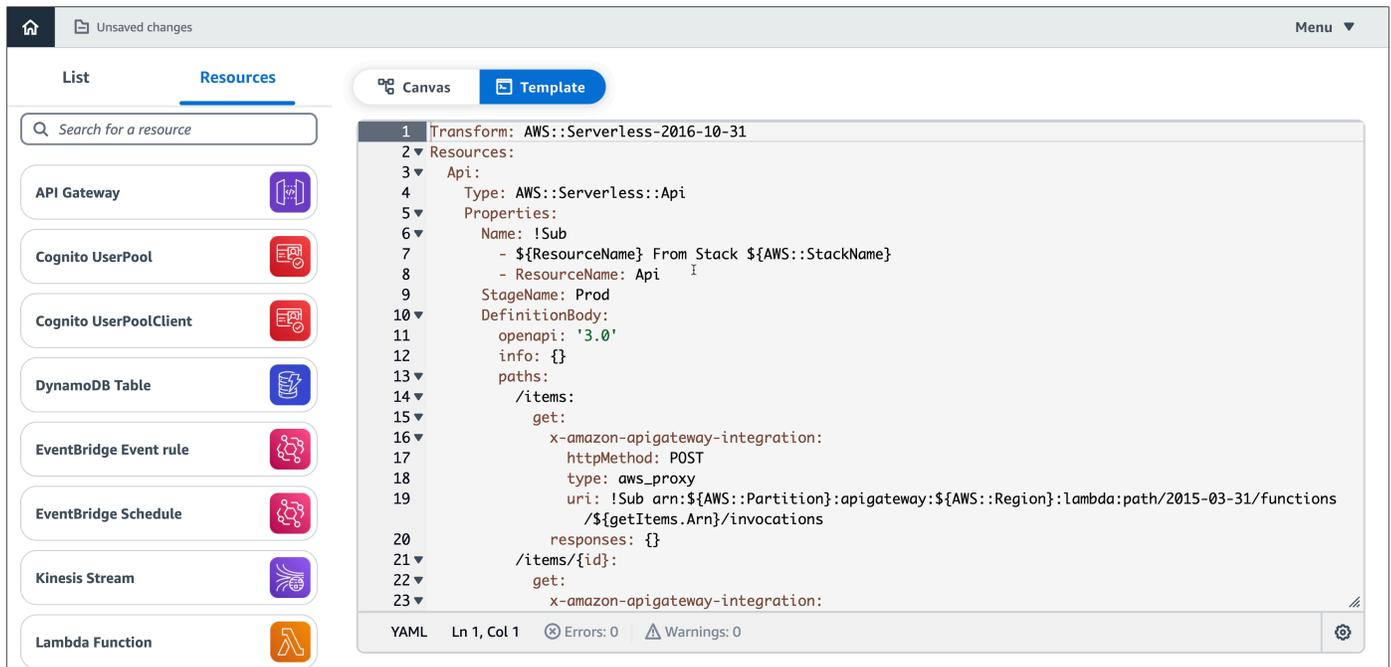
Das Folgende ist ein Beispiel für die AWS Step Functions Funktion, die eine Integration zum Starten von Step Functions bietet Workflow Studio direkt in der Infrastructure Composer-Arbeitsfläche.



## Definieren Sie Ihre Infrastruktur als Codevorlagen (IaC)

Infrastructure Composer erstellt Ihren Infrastrukturcode

Beim Verfassen erstellt Infrastructure Composer automatisch Ihre AWS CloudFormation und AWS Serverless Application Model (AWS SAM) -Vorlagen und befolgt AWS dabei bewährte Methoden. Sie können Ihre Vorlagen direkt in Infrastructure Composer anzeigen und ändern. Infrastructure Composer synchronisiert automatisch Änderungen zwischen dem Visual Canvas und Ihrem Vorlagencode.



The screenshot displays the AWS Infrastructure Composer interface. On the left, there is a 'List' view showing various AWS resources with their respective icons: API Gateway, Cognito UserPool, Cognito UserPoolClient, DynamoDB Table, EventBridge Event rule, EventBridge Schedule, Kinesis Stream, and Lambda Function. The main area is titled 'Resources' and shows a 'Canvas' view of a template. The template is a YAML file for an API Gateway resource, with the following content:

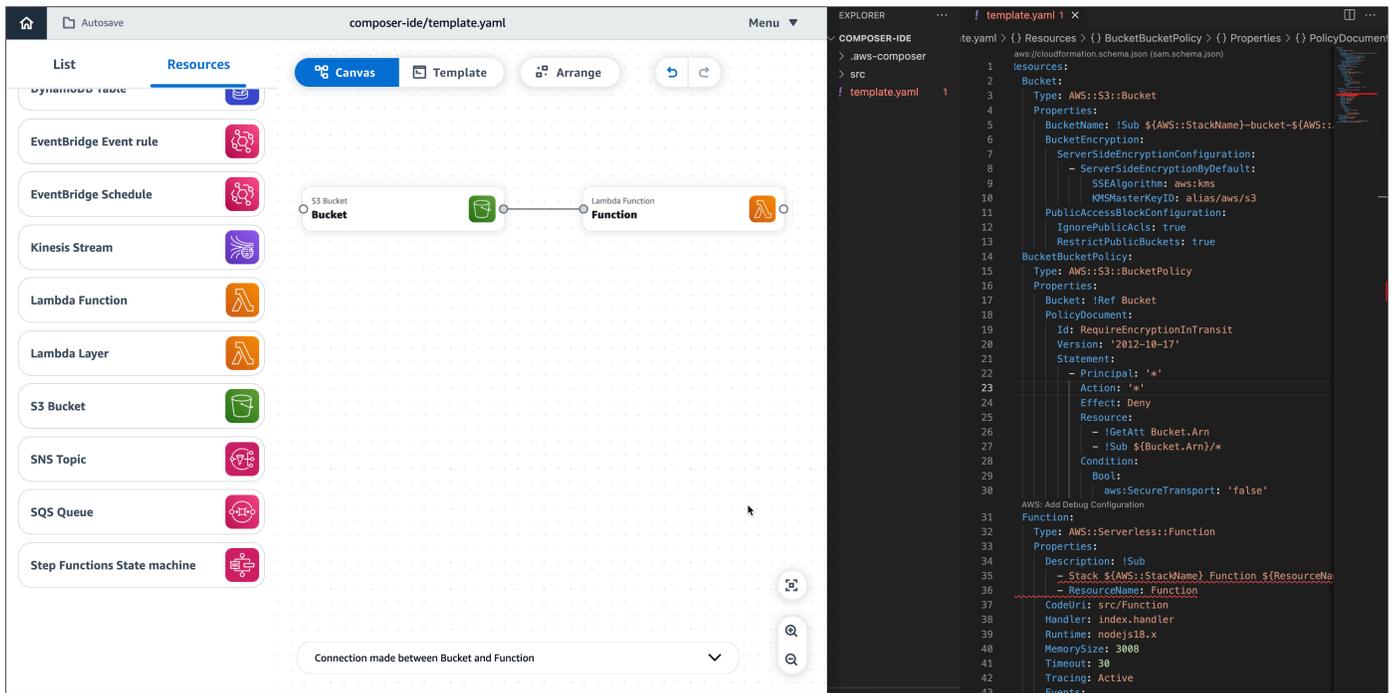
```
1 Transform: AWS::Serverless-2016-10-31
2 Resources:
3   Api:
4     Type: AWS::Serverless::Api
5     Properties:
6       Name: !Sub
7         - ${ResourceName} From Stack ${AWS::StackName}
8         - ResourceName: Api
9     StageName: Prod
10    DefinitionBody:
11      openapi: '3.0'
12      info: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

The status bar at the bottom indicates 'YAML Ln 1, Col 1', 'Errors: 0', and 'Warnings: 0'.

## Integrieren Sie es in Ihre bestehenden Workflows

Importieren Sie bestehende Vorlagen und Projekte

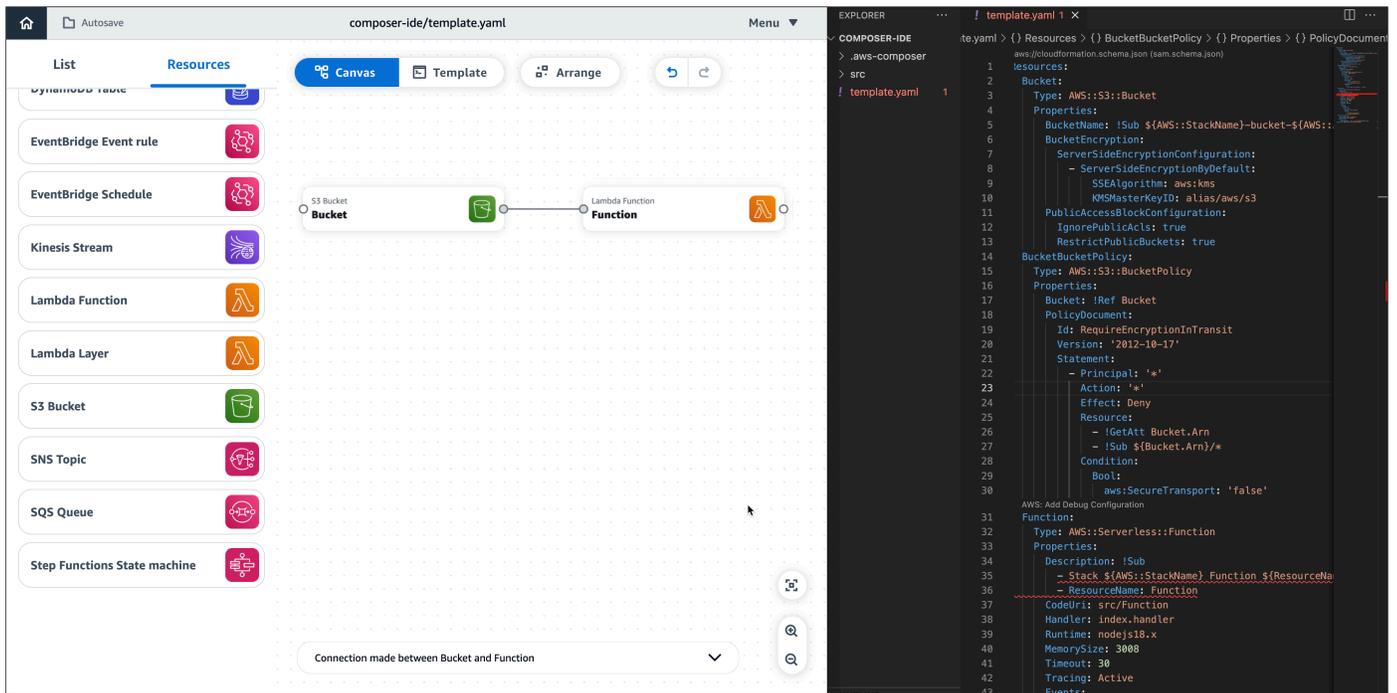
Importieren Sie vorhandene AWS CloudFormation AWS SAM Vorlagen, um sie zum besseren Verständnis zu visualisieren und ihr Design zu ändern. Exportieren Sie die Vorlagen, die Sie in Infrastructure Composer erstellen, und integrieren Sie sie bis zur Bereitstellung in Ihre bestehenden Workflows.



## Möglichkeiten, auf Infrastructure Composer zuzugreifen

### Über die Infrastructure Composer-Konsole

Greifen Sie über die Infrastructure Composer-Konsole auf Infrastructure Composer zu, um schnell loszulegen. Darüber hinaus können Sie den lokalen Synchronisierungsmodus verwenden, um Infrastructure Composer automatisch mit Ihrem lokalen Computer zu synchronisieren und zu speichern.



## Von der Konsole aus AWS CloudFormation

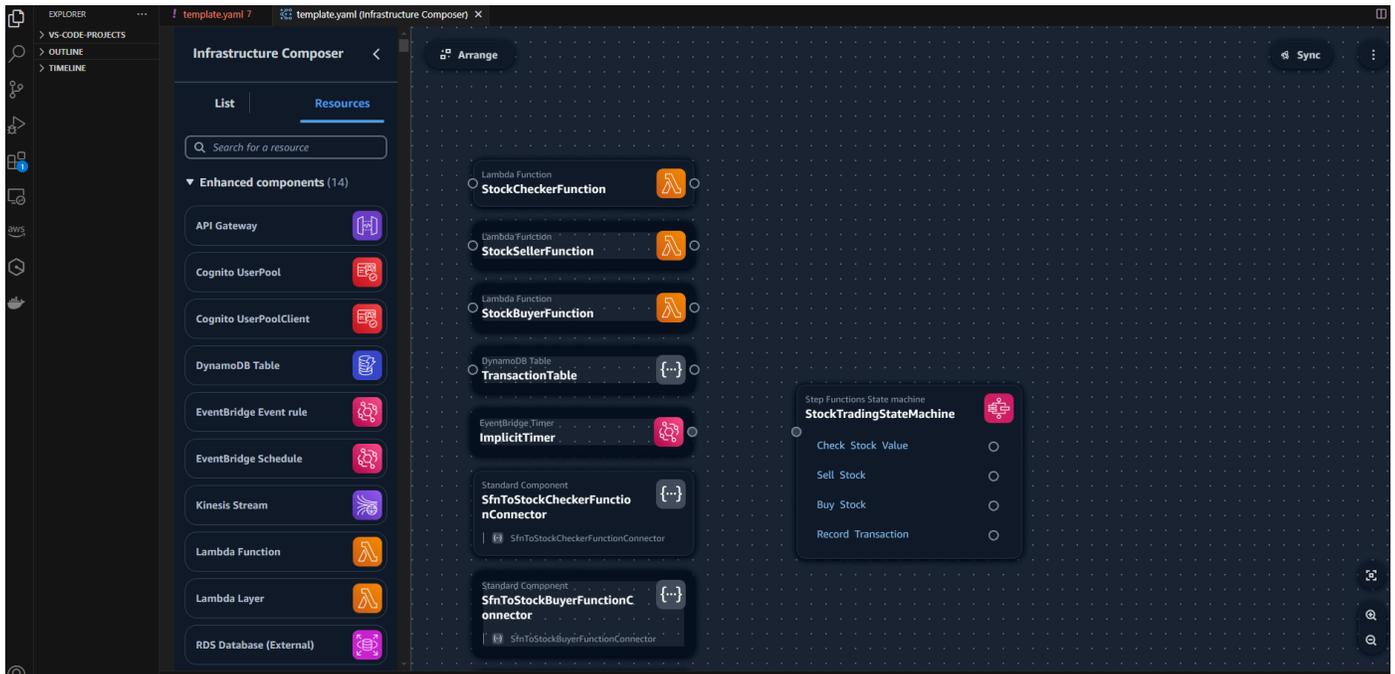
Die Infrastructure [CloudFormation Composer-Konsole unterstützt auch den Konsolenmodus](#), eine Verbesserung gegenüber CloudFormation Designer, die in den AWS CloudFormation Stack-Workflow integriert ist. Dieses neue Tool ist jetzt das empfohlene Tool zur Visualisierung Ihrer CloudFormation Vorlagen.

## Von der Lambda-Konsole

Mit Infrastructure Composer können Sie auch Lambda-Funktionen aus der Lambda-Konsole importieren. Weitere Informationen hierzu finden Sie unter [Funktionen von der Lambda-Konsole in Infrastructure Composer importieren](#).

## Aus dem AWS Toolkit for Visual Studio Code

Greifen Sie über die Erweiterung Toolkit for VS Code auf Infrastructure Composer zu, um Infrastructure Composer in Ihre lokale Entwicklungsumgebung zu integrieren.



## Weitere Informationen

Weitere Informationen zu Infrastructure Composer finden Sie in den folgenden Ressourcen:

- [Infrastructure Composer-Karten](#)
- [Serverlose Anwendungen visuell zusammenstellen und erstellen | Serverless Office Hours](#) — Überblick und Demo von Infrastructure Composer.

## Nächste Schritte

Informationen zur Einrichtung von Infrastructure Composer finden Sie unter [Erste Schritte mit der Infrastructure Composer-Konsole](#)

## Serverlose Konzepte für AWS-Infrastruktur-Composer

Informieren Sie sich vor der Verwendung AWS-Infrastruktur-Composer über grundlegende serverlose Konzepte.

# Serverlose Konzepte

## Ereignisgesteuerte Architektur

Eine serverlose Anwendung besteht aus einzelnen AWS Diensten, z. B. AWS Lambda für Datenverarbeitung und Amazon DynamoDB für die Datenbankverwaltung, die jeweils eine spezielle Rolle erfüllen. Diese Dienste sind dann über eine ereignisgesteuerte Architektur lose miteinander integriert. Weitere Informationen zur ereignisgesteuerten Architektur finden Sie unter [Was ist eine ereignisgesteuerte Architektur?](#) .

## Infrastruktur als Code (IaC)

Infrastructure as Code (IaC) ist eine Methode, Infrastruktur auf die gleiche Weise zu behandeln, wie Entwickler Code behandeln, wobei die gleiche Strenge bei der Entwicklung von Anwendungscode auf die Infrastrukturbereitstellung angewendet wird. Sie definieren Ihre Infrastruktur in einer Vorlagendatei, stellen sie bereit AWS und erstellen die Ressourcen AWS für Sie. Mit IAC definieren Sie im Code, was Sie bereitstellen AWS möchten. Weitere Informationen finden Sie unter [Infrastruktur als Code](#) in der Einführung in ein AWS AWS Whitepaper. DevOps

## Serverlose Technologien

Mit AWS serverlosen Technologien können Sie Anwendungen erstellen und ausführen, ohne Ihre eigenen Server verwalten zu müssen. Die gesamte Serververwaltung erfolgt AWSüber und bietet viele Vorteile wie automatische Skalierung und integrierte Hochverfügbarkeit, sodass Sie Ihre Idee schnell in die Produktion umsetzen können. Durch den Einsatz serverloser Technologien können Sie sich auf den Kern Ihres Produkts konzentrieren, ohne sich um die Verwaltung und den Betrieb von Servern kümmern zu müssen. [Weitere Informationen zu Serverless finden Sie unter Serverless on. AWS](#)

Eine grundlegende Einführung in die wichtigsten AWS serverlosen Dienste finden Sie unter Serverless [101: Understanding the serverless services at Serverless Land](#).

# Infrastructure Composer-Karten

Infrastructure Composer vereinfacht das Schreiben von Infrastructure as Code (IaC) für AWS CloudFormation Ressourcen. Um Infrastructure Composer effektiv nutzen zu können, müssen Sie zunächst zwei grundlegende Konzepte verstehen: Infrastructure [Composer-Karten](#) und [Kartenverbindungen](#).

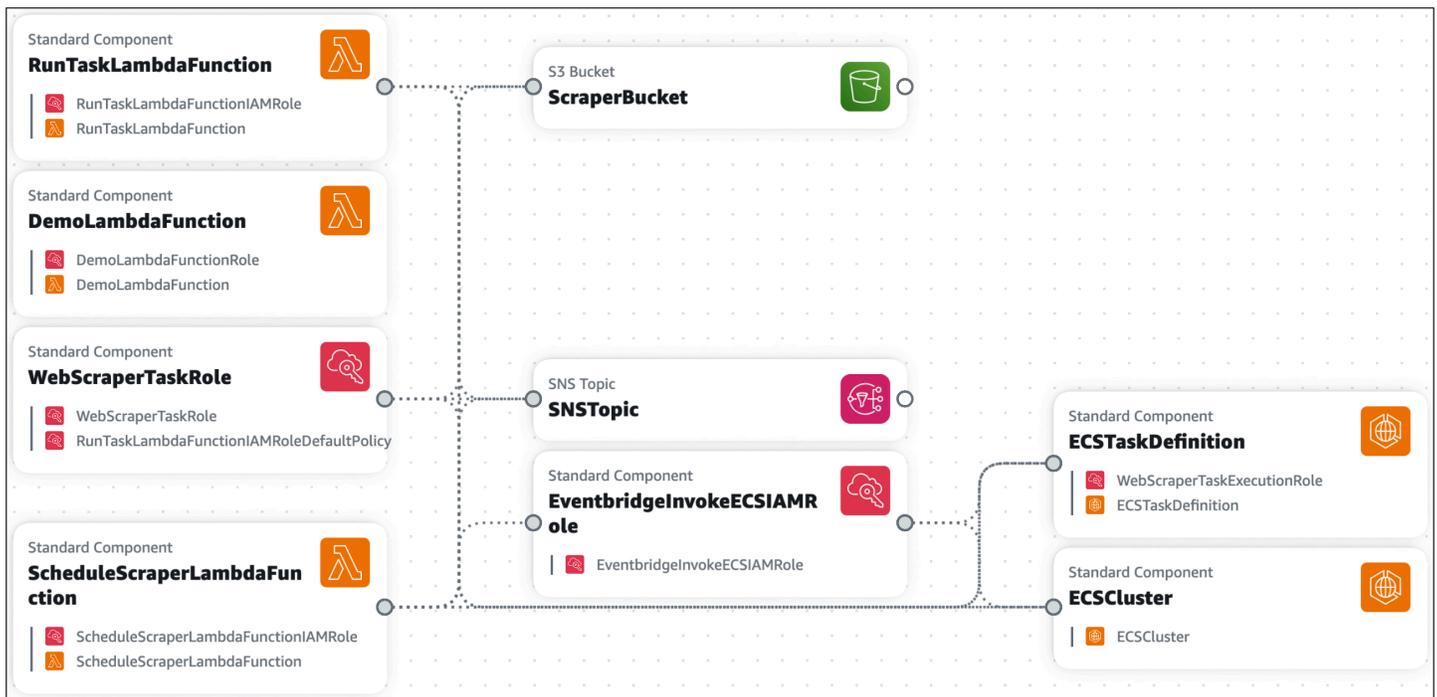
In Infrastructure Composer stehen Karten für AWS CloudFormation Ressourcen. Es gibt zwei allgemeine Kategorien von Karten:

- [Erweiterte Komponentenkarte](#) — Eine Sammlung von AWS CloudFormation Ressourcen, die zu einer einzigen kuratierten Karte zusammengefasst wurden. Sie verbessert die Benutzerfreundlichkeit und Funktionalität und ist für eine Vielzahl von Anwendungsfällen konzipiert. Karten mit erweiterten Komponenten sind die ersten Karten, die in der Ressourcenpalette von Infrastructure Composer aufgeführt werden.
- [Standard-IaC-Ressourcenkarte](#) — Eine einzelne AWS CloudFormation Ressource. Jede Standard-IaC-Ressourcenkarte trägt nach dem Ziehen auf die Arbeitsfläche die Bezeichnung Standardkomponente und kann zu mehreren Ressourcen kombiniert werden.

## Note

Je nach Karte kann eine Standard-IaC-Ressourcenkarte als Standard-Komponentenkarte bezeichnet werden, nachdem sie auf die visuelle Arbeitsfläche gezogen wurde. Das bedeutet einfach, dass es sich bei der Karte um eine Sammlung von einer oder mehreren Standard-IaC-Ressourcenkarten handelt.

Einige Kartentypen sind zwar in der Ressourcenpalette verfügbar, Karten können jedoch auch auf der Arbeitsfläche angezeigt werden, wenn Sie eine vorhandene AWS CloudFormation oder AWS Serverless Application Model (AWS SAM) -Vorlage in Infrastructure Composer importieren. Die folgende Abbildung zeigt ein Beispiel für eine importierte Anwendung, die verschiedene Kartentypen enthält:



## Themen

- [Verbesserte Komponentenkarten in Infrastructure Composer](#)
- [Standardkomponentenkarten in Infrastructure Composer](#)
- [Kartenverbindungen in Infrastructure Composer](#)

## Verbesserte Komponentenkarten in Infrastructure Composer

Karten mit erweiterten Komponenten werden von Infrastructure Composer erstellt und verwaltet. Jede Karte enthält AWS CloudFormation Ressourcen, die häufig zusammen verwendet werden, wenn Anwendungen darauf erstellt werden AWS. Ihr Infrastrukturcode wird von Infrastructure Composer nach AWS bewährten Methoden erstellt. Verbesserte Komponentenkarten sind eine hervorragende Möglichkeit, mit dem Entwerfen Ihrer Anwendung zu beginnen.

Karten mit erweiterten Komponenten sind in der Palette Ressourcen im Abschnitt Erweiterte Komponenten verfügbar.

Erweiterte Komponentenkarten können in Infrastructure Composer vollständig konfiguriert und verwendet werden, um Ihre serverlosen Anwendungen zu entwerfen und zu erstellen. Wir empfehlen, erweiterte Komponentenkarten zu verwenden, wenn Sie Ihre Anwendungen ohne vorhandenen Code entwerfen.

In dieser Tabelle sind unsere erweiterten Komponenten mit Links zur AWS CloudFormation oder AWS Serverless Application Model (AWS SAM) -Vorlagenspezifikation der ausgewählten Ressource der Karte aufgeführt:

Karte	Referenz
Amazon API Gateway	<a href="#">AWS::Serverless::API</a>
Amazon Cognito UserPool	<a href="#">AWS::Cognito::UserPool</a>
Amazon Cognito UserPoolClient	<a href="#">AWS::Cognito::UserPoolClient</a>
Amazon DynamoDB-Tabelle	<a href="#">AWS::DynamoDB::Table</a>
EventBridge Amazon-Event-Regel	<a href="#">AWS::Events::Rule</a>
EventBridge Zeitplan	<a href="#">AWS::Scheduler::Schedule</a>
Amazon Kinesis Stream	<a href="#">AWS::Kinesis::Stream</a>
AWS Lambda Funktion	<a href="#">AWS::Serverless::Function</a>
Lambda-Schicht	<a href="#">AWS::Serverless::LayerVersion</a>
Bucket für Amazon Simple Storage Service (Amazon S3)	<a href="#">AWS::S3::Bucket</a>
Thema Amazon Simple Notification Service (Amazon SNS)	<a href="#">AWS::SNS::Topic</a>
Amazon Simple Queue Service (Amazon SQS) -Warteschlange	<a href="#">AWS::SQS::Queue</a>
AWS Step Functions Zustandsmaschine	<a href="#">AWS::Serverless::StateMachine</a>

## Beispiel

Im Folgenden finden Sie ein Beispiel für eine erweiterte S3-Bucket-Komponente:



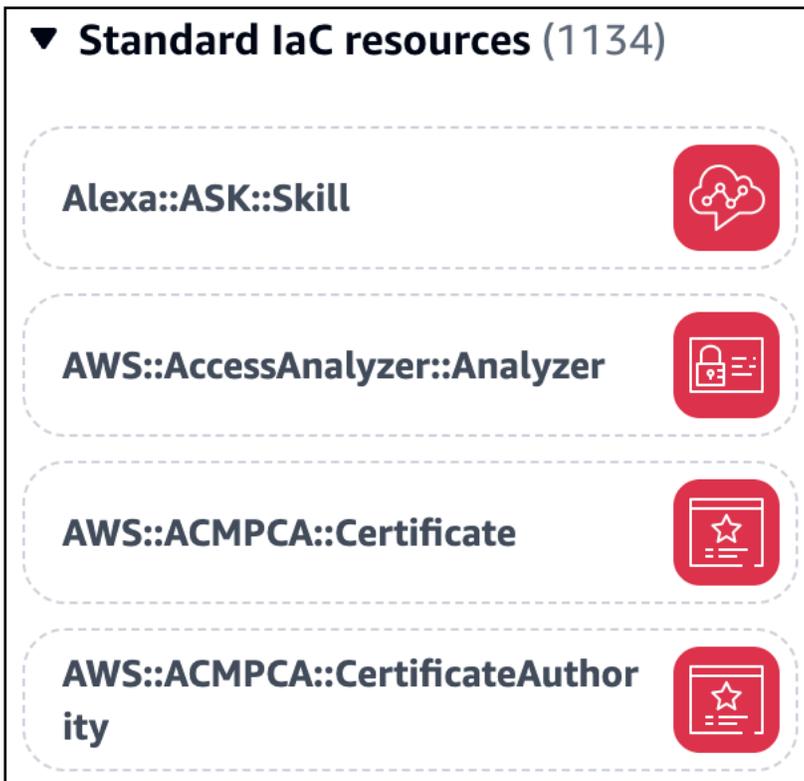
Wenn Sie eine S3-Bucket-Komponentenkarte auf die Arbeitsfläche ziehen und Ihre Vorlage aufrufen, werden Ihrer Vorlage die folgenden beiden AWS CloudFormation Ressourcen hinzugefügt:

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

Die erweiterte Komponentenkarte S3 Bucket stellt zwei AWS CloudFormation Ressourcen dar, die beide für die Interaktion eines Amazon Simple Storage Service (Amazon S3) -Buckets mit anderen Services in Ihrer Anwendung erforderlich sind.

## Standardkomponentenkarten in Infrastructure Composer

Bevor eine Standardkomponentenkarte auf der visuellen Arbeitsfläche von Infrastructure Composer platziert wird, wird sie in Infrastructure Composer auf der Ressourcenpalette als Standard-Ressourcenkarte (IaC) aufgeführt. Eine Standard-Ressourcenkarte (IaC) steht für eine einzelne AWS CloudFormation Ressource. Jede Standard-IaC-Ressourcenkarte wird, sobald sie auf der visuellen Leinwand platziert wurde, zu einer Karte mit der Bezeichnung Standardkomponente. Sie kann kombiniert werden, um mehrere AWS CloudFormation Ressourcen darzustellen.



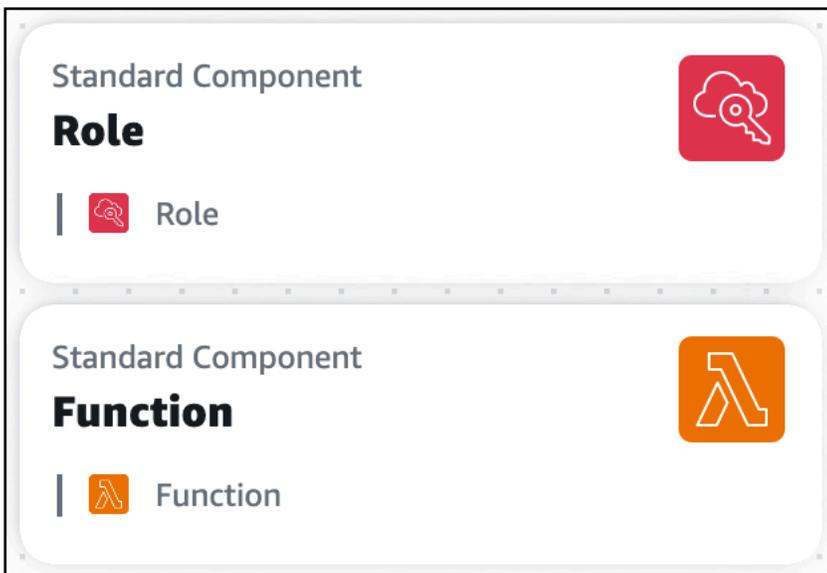
Jede Standard-IaC-Ressourcenkarte kann anhand ihres AWS CloudFormation Ressourcentyps identifiziert werden. Im Folgenden finden Sie ein Beispiel für eine Standard-IaC-Ressourcenkarte, die einen `AWS::ECS::Cluster` AWS CloudFormation Ressourcentyp darstellt:



Jede Standardkomponentenkarte visualisiert die AWS CloudFormation Ressourcen, die sie enthält. Im Folgenden finden Sie ein Beispiel für eine Standard-Komponentenkarte, die zwei Standard-IaC-Ressourcen enthält:



Bei der Konfiguration der Eigenschaften Ihrer Standardkomponentenkarten kann Infrastructure Composer verwandte Karten miteinander kombinieren. Hier sind zum Beispiel zwei Standardkomponentenkarten:



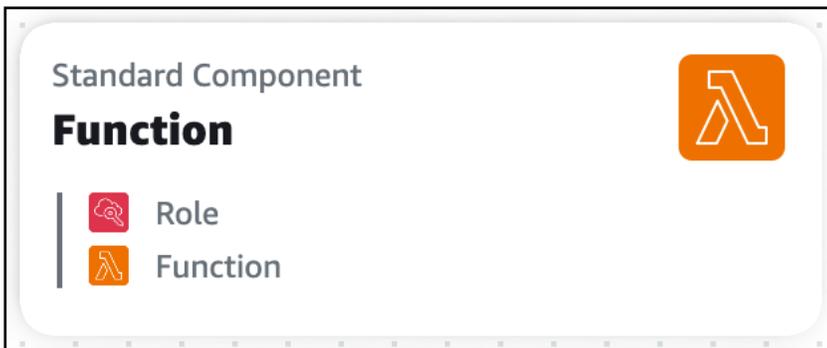
Im Bereich mit den Ressourceneigenschaften der Standardkomponentenkarte, die eine `AWS::Lambda::Function` Ressource darstellt, verweisen wir anhand ihrer logischen ID auf die Rolle AWS Identity and Access Management (IAM):

The screenshot displays the AWS-Infrastruktur-Composer interface. On the left, a grid contains two standard component cards: a red 'Role' card and a blue 'Function' card. The 'Function' card is highlighted with a blue border. On the right, the 'Resource properties' panel is open, showing details 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 panel, there is a 'Resource reference' button with an external link icon.

Nach dem Speichern unserer Vorlage werden die beiden Standardkomponentenkarten zu einer einzigen Standardkomponentenkarte zusammengefasst.



## Kartenverbindungen in Infrastructure Composer

AWS-Infrastruktur-Composer In wird eine Verbindung zwischen zwei Karten visuell durch eine Linie dargestellt. Diese Linien stellen ereignisgesteuerte Beziehungen innerhalb Ihrer Anwendung dar.

Themen

- [Verbindungen zwischen Karten](#)
- [Verbindungen zwischen erweiterten Komponentenkarten](#)
- [Verbindungen zu und von Standard-IaC-Ressourcenkarten](#)

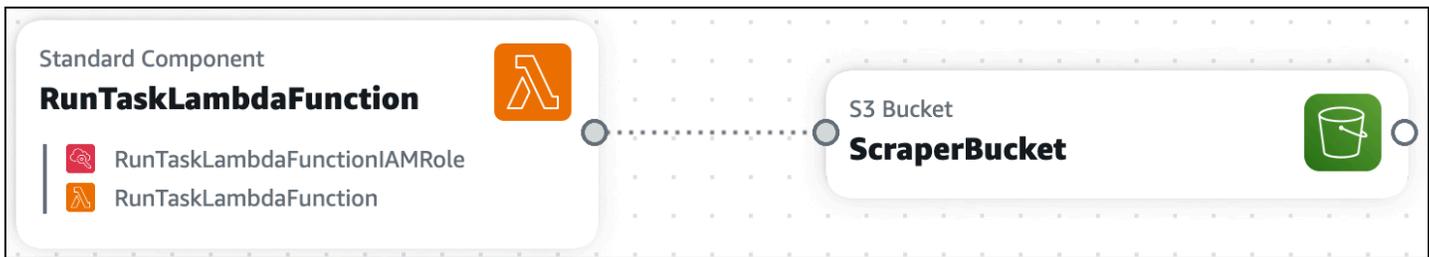
### Verbindungen zwischen Karten

Wie Sie Karten miteinander verbinden, hängt vom Kartentyp ab. Jede erweiterte Karte hat mindestens einen Anschluss. Um sie zu verbinden, wählen Sie einfach einen Anschluss aus und ziehen ihn auf den Anschluss einer anderen Karte. Infrastructure Composer verbindet dann die beiden Ressourcen oder zeigt eine Meldung an, dass diese Konfiguration nicht unterstützt wird.



Wie oben zu sehen ist, sind die Linien zwischen Karten mit erweiterten Komponenten durchgehend. Umgekehrt verfügen Standard-IaC-Ressourcenkarten (auch als Standardkomponentenkarten bezeichnet) nicht über Anschlüsse. Für diese Karten müssen Sie diese ereignisgesteuerten Beziehungen in der Vorlage Ihrer Anwendung angeben. Infrastructure Composer erkennt dann

automatisch deren Verbindungen und visualisiert sie mit einer gepunkteten Linie zwischen Ihren Karten.

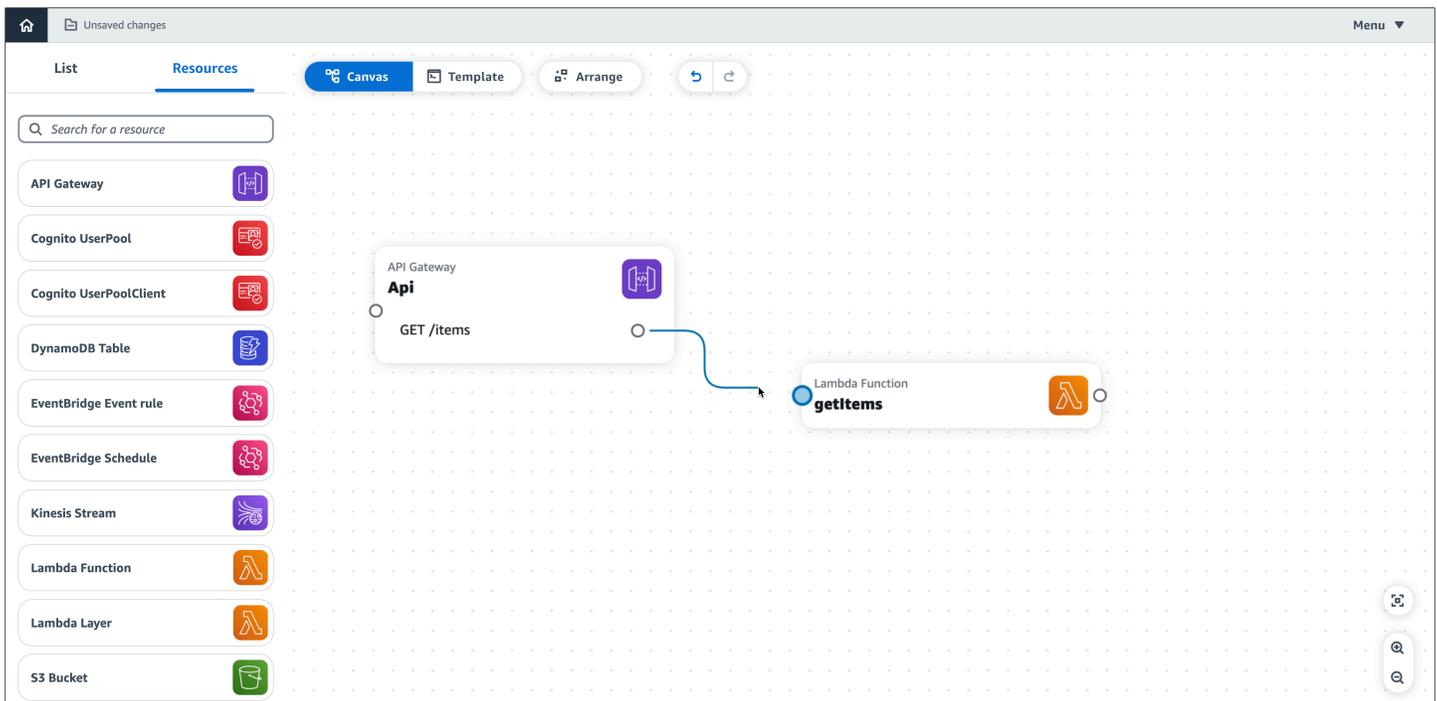


Weitere Informationen finden Sie in den folgenden Abschnitten.

## Verbindungen zwischen erweiterten Komponentenkarten

In Infrastructure Composer wird eine Verbindung zwischen zwei erweiterten Komponentenkarten visuell durch eine durchgezogene Linie dargestellt. Diese Linien stellen ereignisgesteuerte Beziehungen innerhalb Ihrer Anwendung dar.

Um zwei Karten zu verbinden, klicken Sie auf einen Anschluss von einer Karte und ziehen Sie ihn auf einen Anschluss auf einer anderen Karte.



**Note**

Standard-IaC-Ressourcenkarten haben keine Connector-Ports. Für diese Karten müssen Sie ihre ereignisgesteuerten Beziehungen in der Vorlage Ihrer Anwendung angeben. Infrastructure Composer erkennt dann automatisch ihre Verbindungen und visualisiert sie mit einer gepunkteten Linie zwischen Ihren Karten.

Weitere Informationen finden Sie unter [Karten auf der visuellen Leinwand von Infrastructure Composer Connect](#).

## Welche erweiterten Komponenten bieten Karten

Verbindungen zwischen zwei Karten, die visuell durch eine Linie gekennzeichnet sind, bieten bei Bedarf Folgendes:

- AWS Identity and Access Management (IAM-) Richtlinien
- Umgebungsvariablen
- --Ereignisse

### IAM-Richtlinien

Wenn eine Ressource die Erlaubnis benötigt, eine andere Ressource aufzurufen, stellt Infrastructure Composer ressourcenbasierte Richtlinien mithilfe von AWS Serverless Application Model (AWS SAM) -Richtlinienvorlagen bereit.

- Weitere Informationen zu IAM-Berechtigungen und -Richtlinien finden Sie im [IAM-Benutzerhandbuch unter Überblick über die Zugriffsverwaltung: Berechtigungen und Richtlinien](#).
- Weitere Informationen zu AWS SAM Richtlinienvorlagen finden Sie unter [AWS SAM Richtlinienvorlagen](#) im AWS Serverless Application Model Entwicklerhandbuch.

### Umgebungsvariablen

Umgebungsvariablen sind temporäre Werte, die geändert werden können, um das Verhalten Ihrer Ressourcen zu beeinflussen. Falls erforderlich, definiert Infrastructure Composer den Infrastrukturcode, um Umgebungsvariablen zwischen Ressourcen zu verwenden.

## --Ereignisse

Ressourcen können durch verschiedene Arten von Ereignissen eine andere Ressource aufrufen. Falls erforderlich, definiert Infrastructure Composer den Infrastrukturcode, der für die Interaktion von Ressourcen über Ereignistypen erforderlich ist.

## Verbindungen zu und von Standard-IaC-Ressourcenkarten

Alle AWS CloudFormation Ressourcen können in der Ressourcenpalette als Standard-IaC-Ressourcenkarten verwendet werden. Wenn Sie eine Standard-IaC-Ressourcenkarte auf die Arbeitsfläche ziehen, wird eine Standard-IaC-Ressourcenkarte zu einer Standard-Komponentenkarte. Dadurch wird Infrastructure Composer aufgefordert, eine Startvorlage für Ihre Ressource in Ihrer Anwendung zu erstellen.

Weitere Informationen finden Sie unter [Standardkarten in Infrastructure Composer](#).

# Erste Schritte mit der Infrastructure Composer-Konsole

Verwenden Sie die Themen in diesem Abschnitt, um eine Anwendung mithilfe ihrer visuellen Arbeitsfläche einzurichten AWS-Infrastruktur-Composer und zu lernen, wie Sie sie entwerfen. Die Tour und die Tutorials in diesem Abschnitt werden in der Infrastructure Composer-Konsole angezeigt, der Standardbenutzeroberfläche. Die Themen in diesem Abschnitt zeigen Ihnen, wie Sie die Voraussetzungen für die Verwendung von Infrastructure Composer erfüllen, die Infrastructure Composer-Konsole verwenden, ein Projekt laden und ändern und Ihre erste Anwendung erstellen.

Infrastructure Composer ist auch im AWS Toolkit for Visual Studio Code und im CloudFormation Konsolenmodus verfügbar. Die Erfahrungen zwischen den Tools sind im Allgemeinen dieselben, es gibt jedoch einige Unterschiede zwischen den einzelnen Tools. Einzelheiten zur Verwendung von Infrastructure Composer in jedem dieser Tools finden Sie unter [Wo Sie Infrastructure Composer verwenden können](#).

## Themen

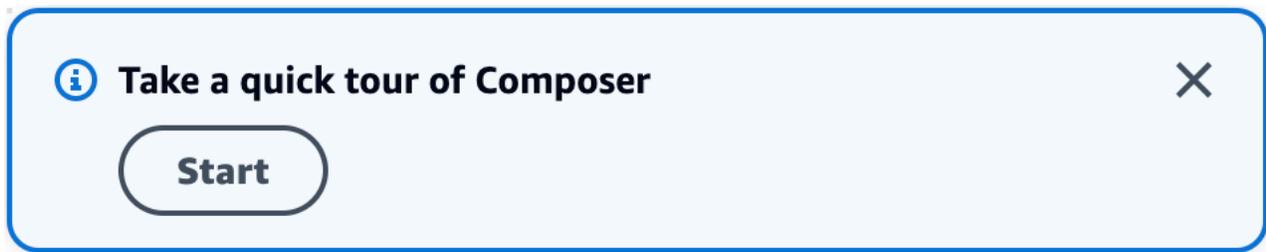
- [Machen Sie einen Rundgang in der Infrastructure Composer-Konsole](#)
- [Laden und ändern Sie das Infrastructure Composer-Demoprojekt](#)
- [Erstellen Sie Ihre erste Anwendung mit Infrastructure Composer](#)

## Machen Sie einen Rundgang in der Infrastructure Composer-Konsole

Um sich einen Überblick über die AWS-Infrastruktur-Composer Funktionsweise zu verschaffen, nehmen Sie an der Tour teil, die in die Infrastructure Composer-Konsole integriert ist. Einen Überblick über die Infrastructure Composer-Konsole finden Sie unter [Machen Sie einen Rundgang in der Infrastructure Composer-Konsole](#). Ausführliche Anleitungen zur Verwendung von Infrastructure Composer finden Sie unter [Wie komponiert man in AWS-Infrastruktur-Composer](#).

Um einen Rundgang durch Infrastructure Composer zu machen

1. Melden Sie sich bei der [Infrastructure Composer-Konsole](#) an.
2. Wählen Sie auf der Startseite die Option Demo öffnen aus.
3. Wählen Sie in der oberen rechten Ecke im Fenster Kurzer Überblick über Composer die Option Start aus.



4. Gehen Sie im Composer-Tourfenster wie folgt vor:

- Um mit dem nächsten Schritt fortzufahren, wählen Sie Weiter.
- Um zum vorherigen Schritt zurückzukehren, wählen Sie Zurück.
- Wählen Sie im letzten Schritt, um die Tour zu beenden, „Ende“.

Die Tour bietet einen kurzen Überblick über die grundlegenden Funktionen von Infrastructure Composer, z. B. die Verwendung, Konfiguration und Verbindung von Karten. Weitere Informationen finden Sie unter [Wie komponiert man in AWS-Infrastruktur-Composer](#).

## Nächste Schritte

Informationen zum Laden und Ändern eines Projekts in Infrastructure Composer finden Sie unter [Laden und ändern Sie das Infrastructure Composer-Demoprojekt](#)

## Laden und ändern Sie das Infrastructure Composer-Demoprojekt

Machen Sie sich in diesem Tutorial mit der Benutzeroberfläche von Infrastructure Composer vertraut und erfahren Sie, wie Sie das Infrastructure Composer-Demoprojekt laden, ändern und speichern.

Dieses Tutorial wird in der Infrastructure Composer-Konsole erstellt. Sobald Sie fertig sind, können Sie loslegen [Erstellen Sie Ihre erste Anwendung mit Infrastructure Composer](#).

### Themen

- [Schritt 1: Öffnen Sie die Demo](#)
- [Schritt 2: Erkunden Sie die visuelle Oberfläche von Infrastructure Composer](#)
- [Schritt 3: Erweitern Sie Ihre Anwendungsarchitektur](#)
- [Schritt 4: Speichern Sie Ihre Anwendung](#)
- [Nächste Schritte](#)

## Schritt 1: Öffnen Sie die Demo

Beginnen Sie mit der Verwendung von Infrastructure Composer, indem Sie ein Demo-Projekt erstellen.

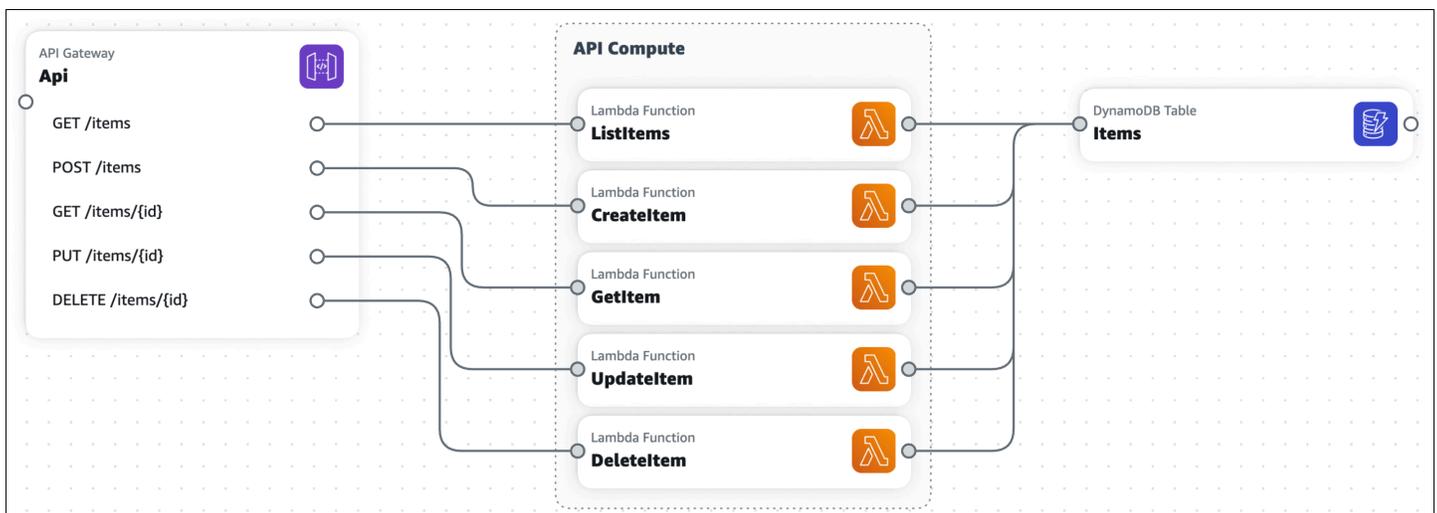
Um ein Demo-Projekt zu erstellen

1. Melden Sie sich bei der [Infrastructure Composer-Konsole](#) an.
2. Wählen Sie auf der Startseite die Option Demo öffnen aus.

Die Demo-Anwendung ist eine einfache serverlose Anwendung zum Erstellen, Lesen, Löschen und Aktualisieren (CRUD), die Folgendes umfasst:

- Eine Amazon API Gateway Gateway-Ressource mit fünf Routen.
- Fünf AWS Lambda Funktionen.
- Eine Amazon DynamoDB-Tabelle.

Das folgende Bild zeigt die Demo:

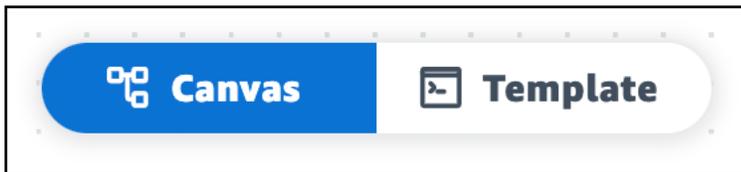


## Schritt 2: Erkunden Sie die visuelle Oberfläche von Infrastructure Composer

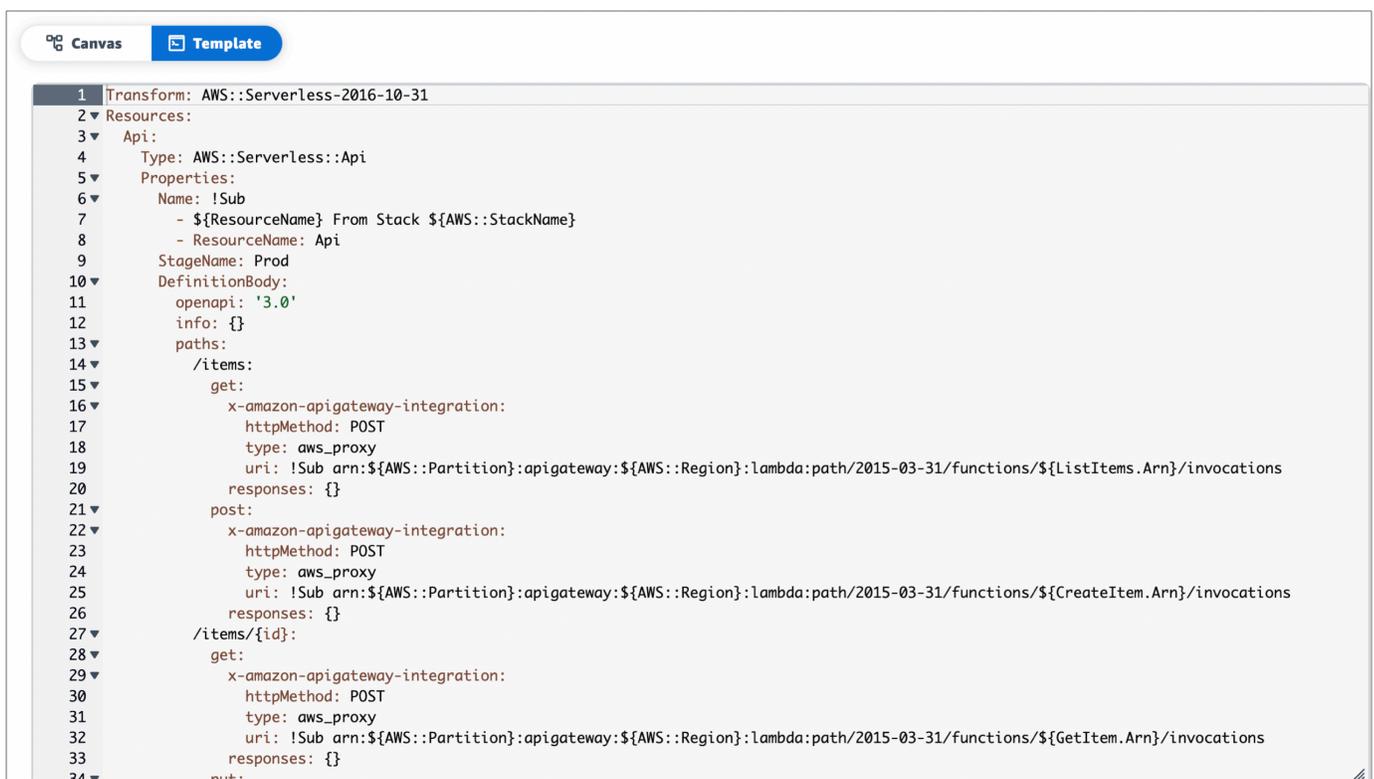
Lernen Sie die Funktionen des Visual Canvas kennen, um Ihr Infrastructure Composer-Demoprojekt zu erstellen. Einen Überblick über das Visual Canvas-Layout finden Sie unter [Visueller Überblick](#).

## Erfahren Sie mehr über die Funktionen der visuellen Leinwand

1. Wenn Sie ein neues oder vorhandenes Anwendungsprojekt öffnen, lädt Infrastructure Composer die Leinwandansicht, wie über dem Hauptansichtsbereich angezeigt.

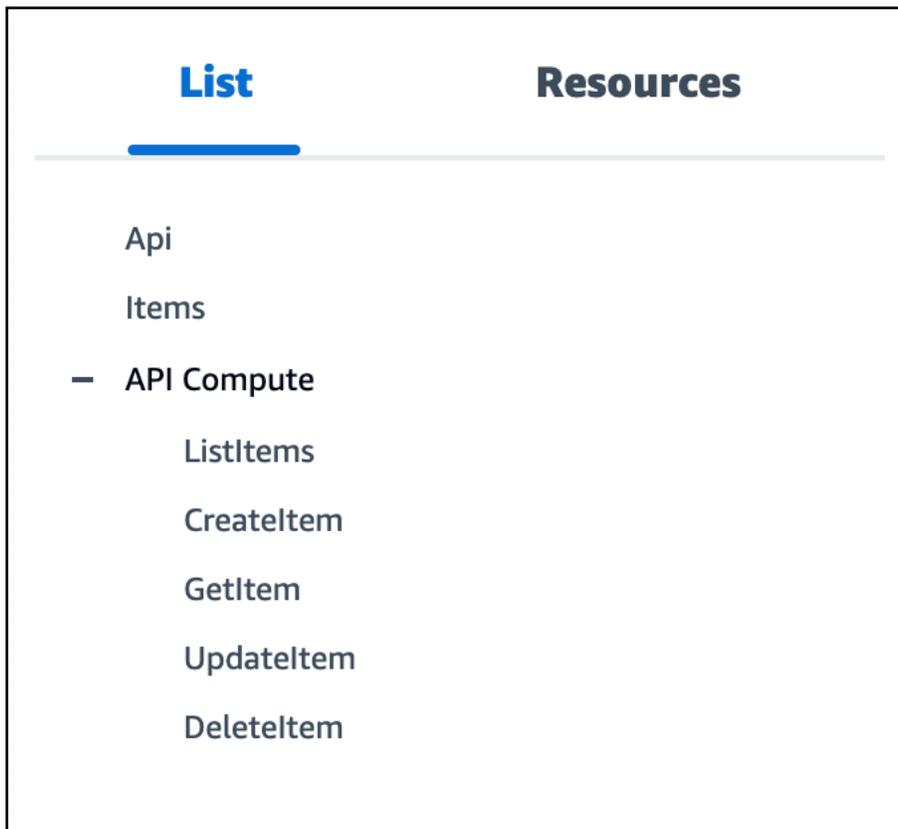


Um den Infrastrukturcode Ihrer Anwendung im Hauptansichtsbereich anzuzeigen, wählen Sie Vorlage. Hier ist zum Beispiel die Vorlagenansicht AWS Serverless Application Model (AWS SAM) des Infrastructure Composer-Demoprojekts.

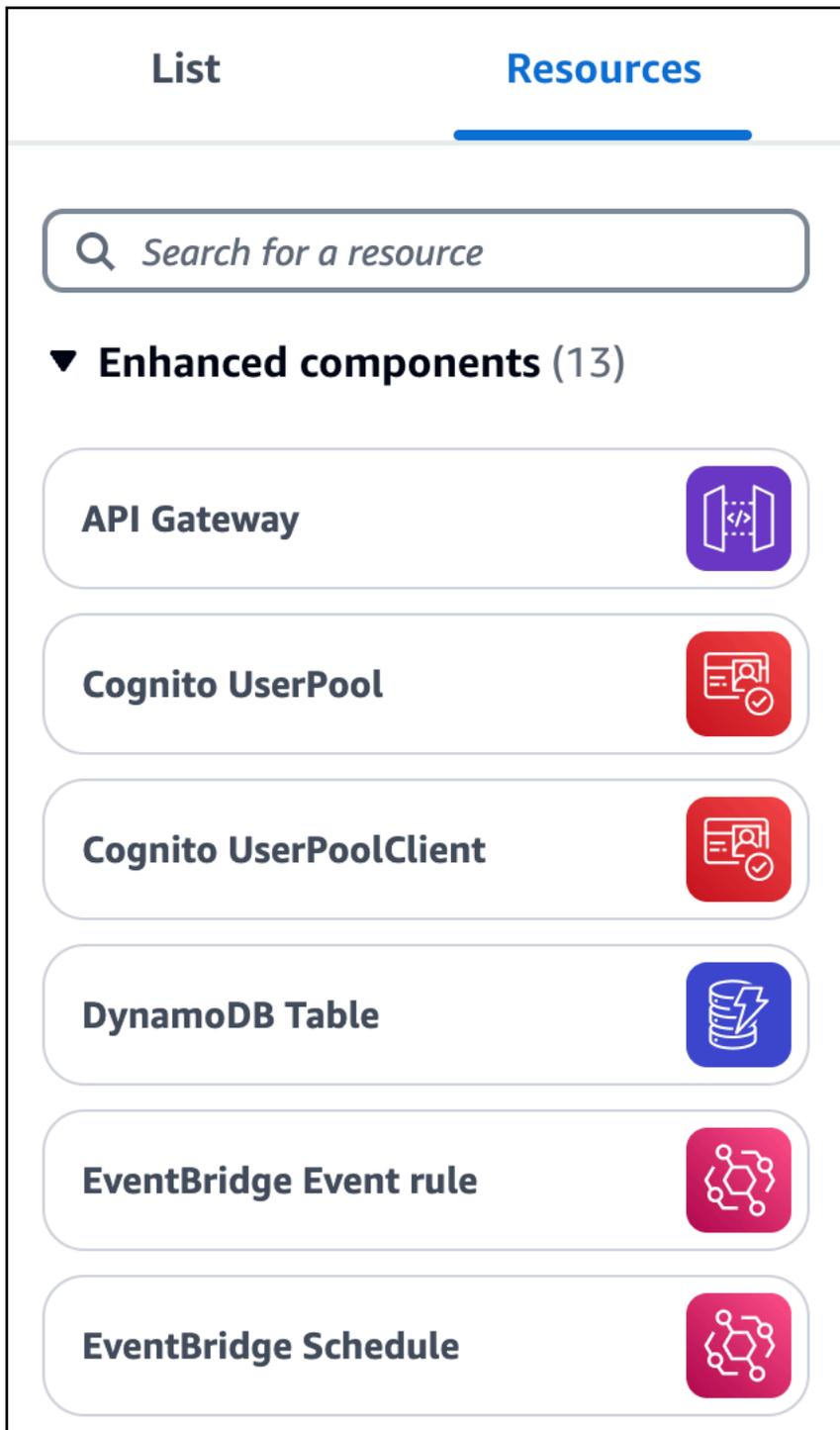
The screenshot shows the Infrastructure Composer interface with the 'Template' view selected. The main area displays a JSON configuration for an AWS SAM application. The configuration includes a transform, resources, an API definition, and several API gateway integrations for different endpoints.

```
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. Um die Leinwandansicht Ihrer Anwendung erneut anzuzeigen, wählen Sie Canvas.
3. Um die Ressourcen Ihrer Anwendung in einer Strukturansicht anzuzeigen, wählen Sie Liste.



- Um die Ressourcenpalette anzuzeigen, wählen Sie Ressourcen. Diese Palette enthält Karten, mit denen Sie Ihre Anwendungsarchitektur erweitern können. Sie können nach Karten suchen oder durch die Liste blättern.



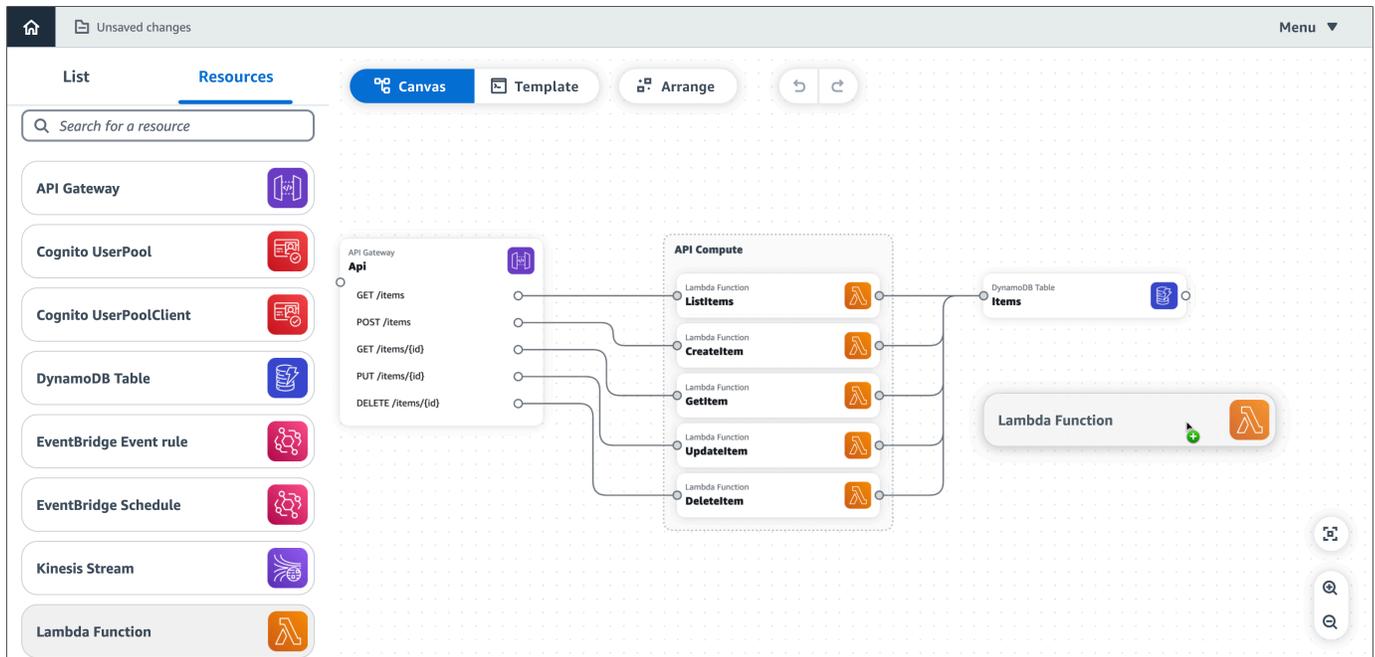
5. Verwenden Sie einfache Gesten, um sich auf der visuellen Leinwand zu bewegen. Weitere Informationen finden Sie unter [Platzieren Sie Karten auf der Leinwand](#).

## Schritt 3: Erweitern Sie Ihre Anwendungsarchitektur

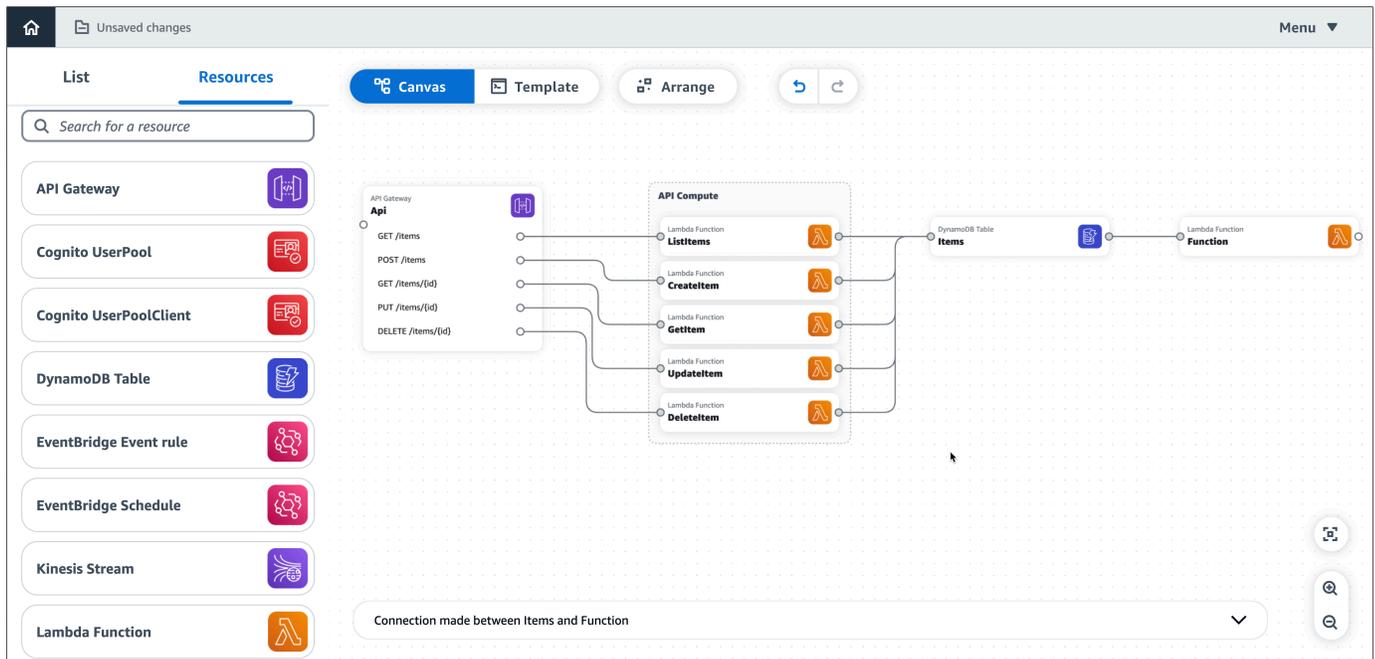
In diesem Schritt erweitern Sie Ihre Anwendungsarchitektur, indem Sie Ihrer DynamoDB-Tabelle eine Lambda-Funktion hinzufügen.

So fügen Sie Ihrer DynamoDB-Tabelle eine Lambda-Funktion hinzu

1. Ziehen Sie die erweiterte Komponentenkarte mit Lambda-Funktionen aus der Ressourcenpalette (Ressourcen) auf die Arbeitsfläche, rechts neben der DynamoDB-Tabellenkarte.



2. Connect die DynamoDB-Tabelle mit der Lambda-Funktion. Um sie zu verbinden, klicken Sie auf den rechten Anschluss der DynamoDB-Tabellenkarte und ziehen Sie ihn auf den linken Anschluss der Lambda-Funktionskarte.
3. Wählen Sie „Anordnen“, um die Karten in der Leinwandansicht zu organisieren.



4. Konfigurieren Sie Ihre Lambda-Funktion. Gehen Sie wie folgt vor, um sie zu konfigurieren:
- Ändern Sie in der Leinwandansicht die Eigenschaften der Funktion im Bereich Ressourceneigenschaften. Doppelklicken Sie auf die Lambda-Funktionskarte, um das Panel zu öffnen. Oder wählen Sie die Karte und dann Details aus. Weitere Informationen zu den konfigurierbaren Lambda-Funktionseigenschaften, die im Bereich mit den Ressourceneigenschaften aufgeführt sind, finden Sie im [AWS Lambda Entwicklerhandbuch](#).
  - Ändern Sie in der Vorlagenansicht den Code für Ihre Funktion (`AWS::Serverless::Function`). Infrastructure Composer synchronisiert Ihre Änderungen automatisch mit der Arbeitsfläche. Weitere Informationen zur Funktionsressource in einer AWS SAM Vorlage finden Sie [AWS::Serverless::Function](#) in der AWS SAM Ressourcen- und Eigenschaftsreferenz.

## Schritt 4: Speichern Sie Ihre Anwendung

Speichern Sie Ihre Anwendung, indem Sie Ihre Anwendungsvorlage manuell auf Ihrem lokalen Computer speichern oder die lokale Synchronisierung aktivieren.

Um Ihre Anwendungsvorlage manuell zu speichern

1. Wählen Sie im Menü Speichern > Vorlagendatei speichern.
2. Geben Sie einen Namen für Ihre Vorlage ein und wählen Sie einen Speicherort auf Ihrem lokalen Computer aus, um Ihre Vorlage zu speichern. Drücken Sie auf Speichern.

Anweisungen zur Aktivierung der lokalen Synchronisierung finden Sie unter [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#).

## Nächste Schritte

Informationen zu den ersten Schritten beim Erstellen Ihrer ersten Anwendung finden Sie unter [Erstellen Sie Ihre erste Anwendung mit Infrastructure Composer](#).

## Erstellen Sie Ihre erste Anwendung mit Infrastructure Composer

In diesem Tutorial erstellen Sie eine serverlose Anwendung AWS-Infrastruktur-Composer zum Erstellen, Lesen, Aktualisieren und Löschen (CRUD), die Benutzer in einer Datenbank verwaltet.

Für dieses Tutorial verwenden wir Infrastructure Composer in der AWS Management Console. Wir empfehlen die Verwendung Google Chrome or Microsoft Edge und ein Browserfenster im Vollbildmodus.

### Sind Sie neu im Bereich Serverless?

Wir empfehlen ein grundlegendes Verständnis der folgenden Themen:

- [Ereignisgesteuerte Architektur](#)
- [Infrastruktur als Code \(IaC\)](#)
- [Serverlose Technologien](#)

Weitere Informationen hierzu finden Sie unter [Serverlose Konzepte für AWS-Infrastruktur-Composer](#).

### Themen

- [Referenz zu den Ressourceneigenschaften](#)
- [Schritt 1: Erstelle dein Projekt](#)
- [Schritt 2: Fügen Sie der Leinwand Karten hinzu](#)
- [Schritt 3: Konfigurieren Sie Ihre API-Gateway-REST-API](#)
- [Schritt 4: Konfigurieren Sie Ihre Lambda-Funktionen](#)

- [Schritt 5: Connect deine Karten](#)
- [Schritt 6: Organisieren Sie die Leinwand](#)
- [Schritt 7: DynamoDB-Tabelle hinzufügen und verbinden](#)
- [Schritt 8: Überprüfen Sie Ihre Vorlage AWS CloudFormation](#)
- [Schritt 9: Integrieren Sie es in Ihre Entwicklungsworkflows](#)
- [Nächste Schritte](#)

## Referenz zu den Ressourceneigenschaften

Verwenden Sie beim Erstellen Ihrer Anwendung diese Tabelle als Referenz, um die Eigenschaften Ihres Amazon API Gateway und der AWS Lambda Ressourcen zu konfigurieren.

Methode	Pfad	Funktionsname
GET	/items	Artikel abrufen
GET	/items/ {id}	getItem
PUT	/artikel/ {id}	Artikel aktualisieren
POST	/artikel	Artikel hinzufügen
DELETE	/artikel/ {id}	deleteItem

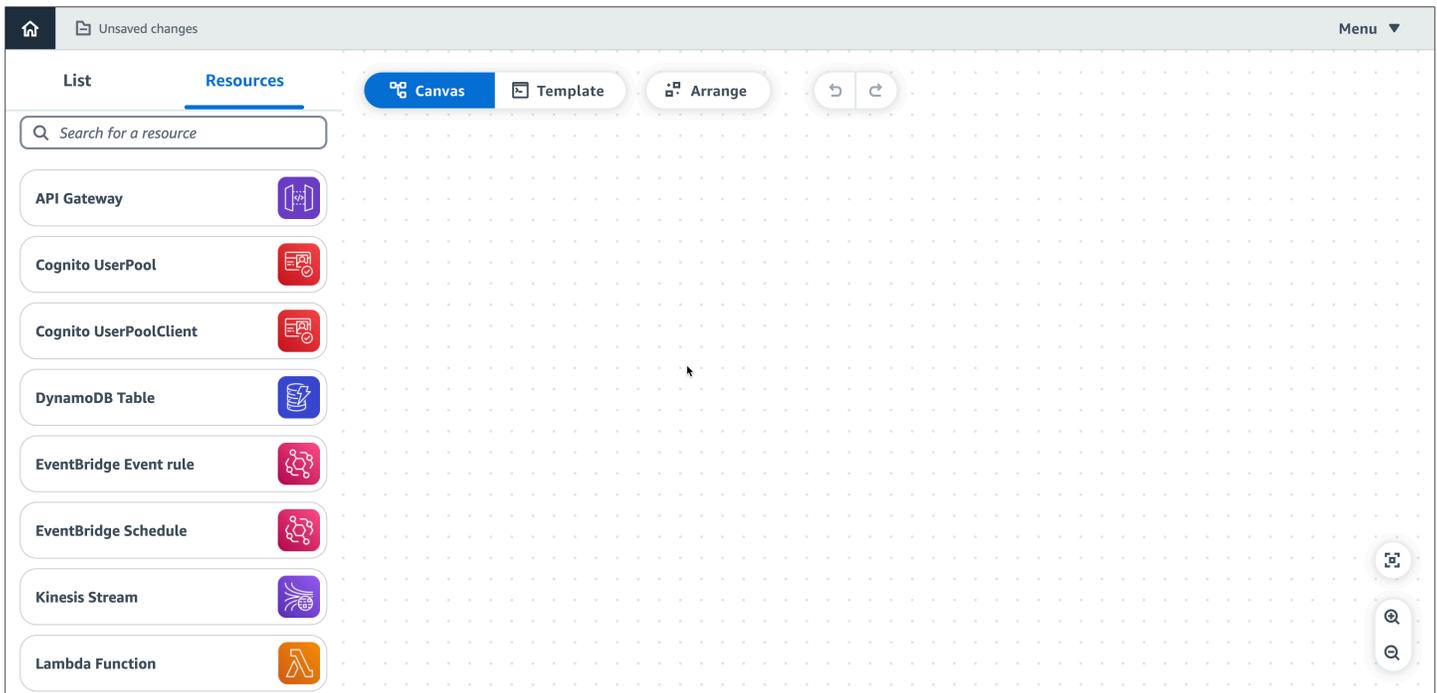
## Schritt 1: Erstelle dein Projekt

Um mit Ihrer serverlosen CRUD-Anwendung zu beginnen, erstellen Sie ein neues Projekt in Infrastructure Composer und aktivieren Sie die lokale Synchronisierung.

Um ein neues leeres Projekt zu erstellen

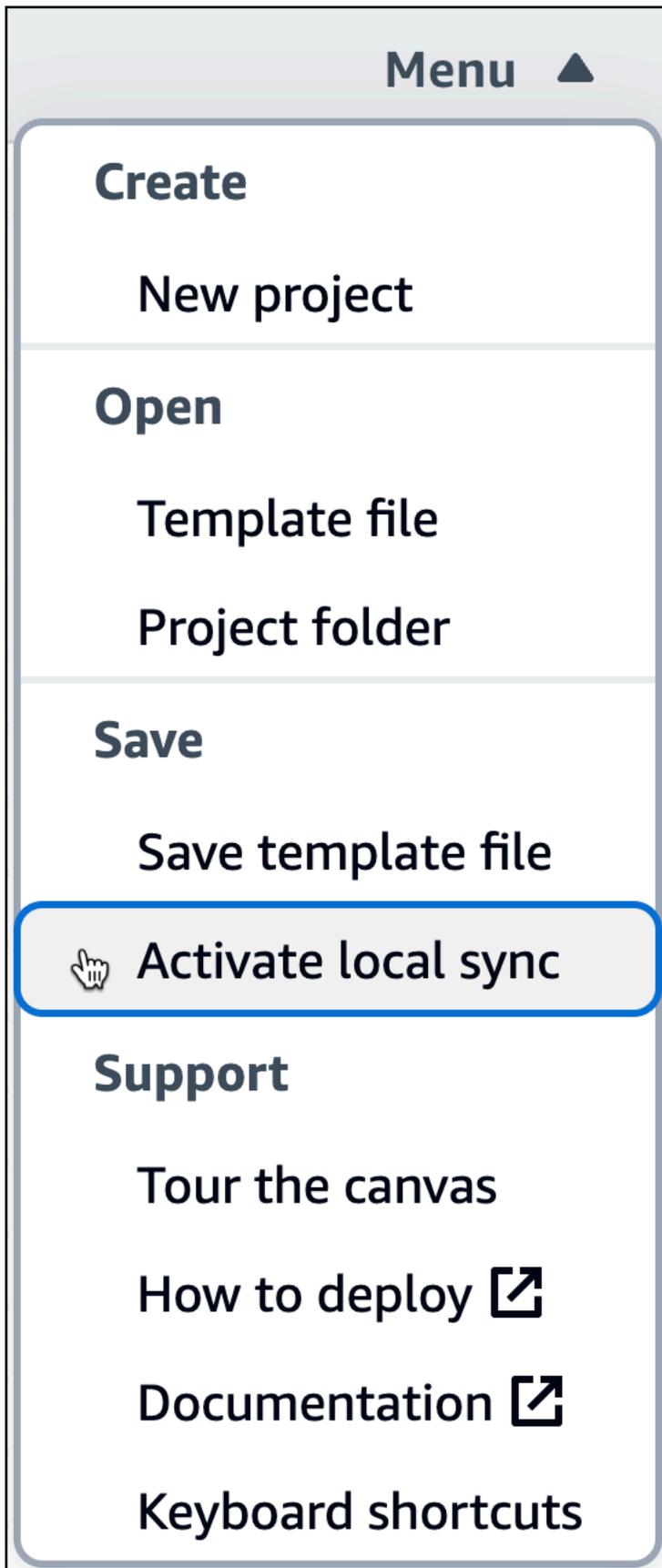
1. Melden Sie sich bei der [Infrastructure Composer-Konsole](#) an.
2. Wählen Sie auf der Startseite die Option Projekt erstellen aus.

Wie in der folgenden Abbildung gezeigt, öffnet Infrastructure Composer die visuelle Arbeitsfläche und lädt eine (leere) Startanwendungsvorlage.



Um die lokale Synchronisierung zu aktivieren

1. Wählen Sie im Infrastructure Composer-Menü Speichern > Lokale Synchronisierung aktivieren aus.



2. Drücken Sie unter Projektstandort auf Ordner auswählen und wählen Sie ein Verzeichnis aus. Hier speichert und synchronisiert Infrastructure Composer Ihre Vorlagendateien und Ordner, während Sie entwerfen.

Der Projektstandort darf keine vorhandene Anwendungsvorlage enthalten.

 Note

Für die lokale Synchronisierung ist ein Browser erforderlich, der die File System Access API unterstützt. Weitere Informationen finden Sie unter [Data Infrastructure Composer erhält Zugriff auf](#).

3. Wenn Sie aufgefordert werden, den Zugriff zuzulassen, wählen Sie Dateien anzeigen aus.
4. Drücken Sie auf Aktivieren, um die lokale Synchronisierung zu aktivieren. Wenn Sie aufgefordert werden, die Änderungen zu speichern, wählen Sie Änderungen speichern.

Wenn diese Option aktiviert ist, wird die Anzeige für automatisches Speichern im oberen linken Bereich Ihrer Leinwand angezeigt.

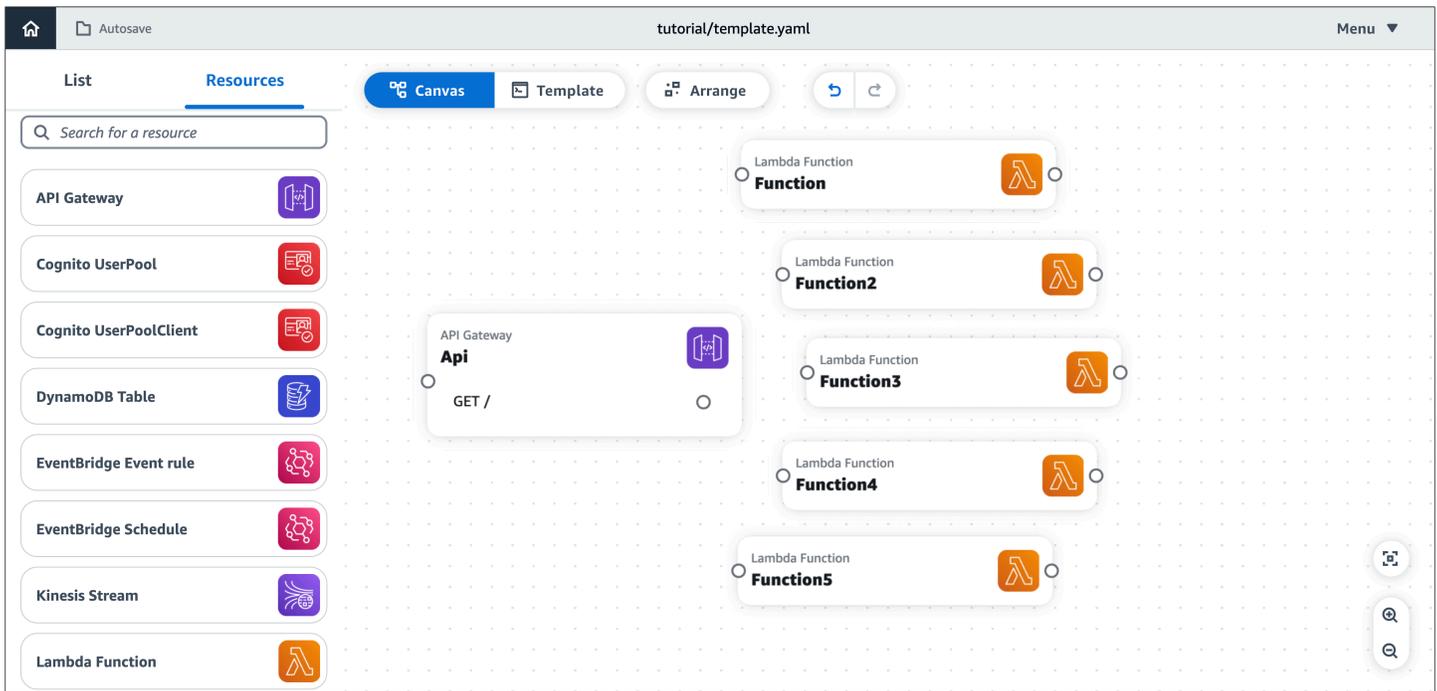
## Schritt 2: Fügen Sie der Leinwand Karten hinzu

Beginnen Sie mit dem Entwurf Ihrer Anwendungsarchitektur mithilfe erweiterter Komponentenkarten, beginnend mit einer API-Gateway-REST-API und fünf Lambda-Funktionen.

So fügen Sie API Gateway- und Lambda-Karten zur Arbeitsfläche hinzu

Gehen Sie in der Palette Ressourcen im Abschnitt Erweiterte Komponenten wie folgt vor:

1. Ziehen Sie eine API-Gateway-Karte auf die Leinwand.
2. Ziehen Sie eine Lambda-Funktionskarte auf die Leinwand. Wiederholen Sie den Vorgang, bis Sie der Leinwand fünf Lambda-Funktionskarten hinzugefügt haben.



## Schritt 3: Konfigurieren Sie Ihre API-Gateway-REST-API

Fügen Sie als Nächstes fünf Routen zu Ihrer API-Gateway-Karte hinzu.

So fügen Sie Routen zur API-Gateway-Karte hinzu

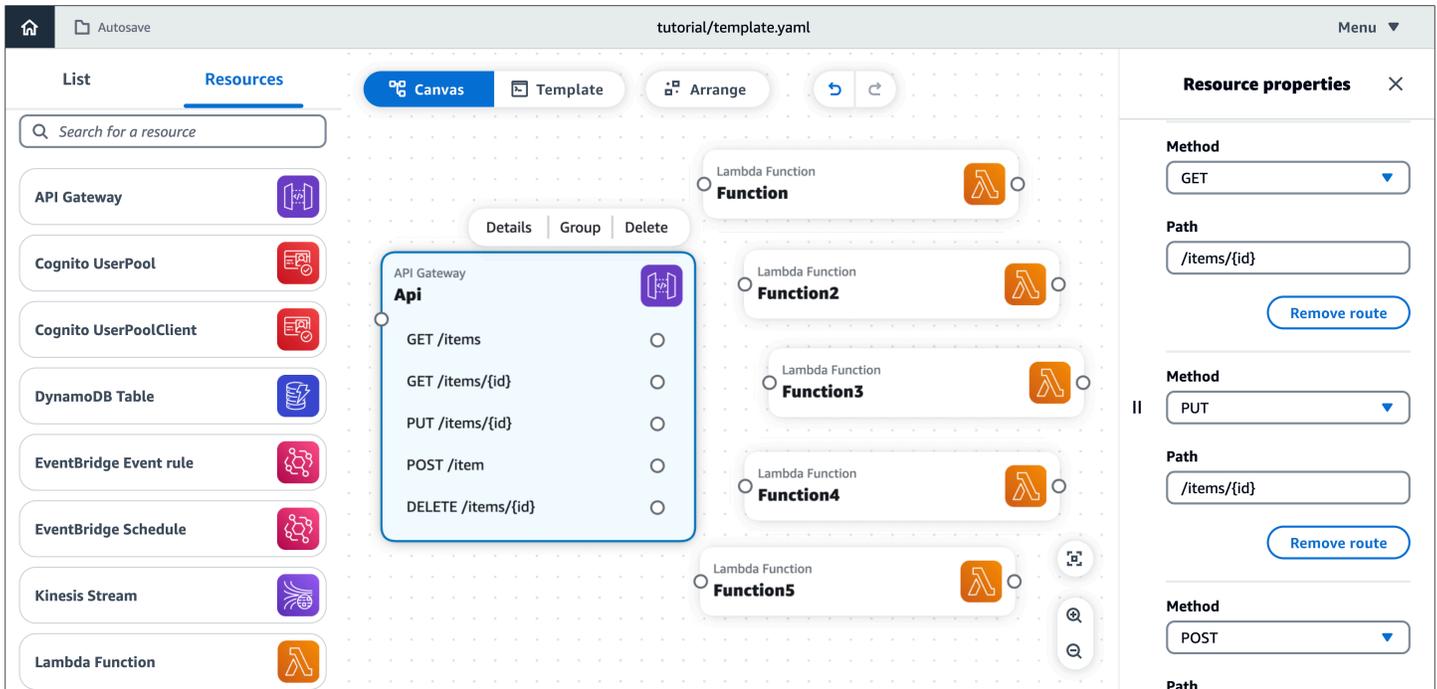
1. Öffnen Sie den Bereich mit den Ressourceneigenschaften für die API-Gateway-Karte. Doppelklicken Sie auf die Karte, um das Panel zu öffnen. Oder wählen Sie die Karte und dann Details aus.
2. Gehen Sie im Bereich Ressourceneigenschaften unter Routen wie folgt vor:

### Note

Verwenden Sie für jede der folgenden Routen die HTTP-Methode und die Pfadwerte, die in der [Referenztable mit den Ressourceneigenschaften](#) angegeben sind.

- a. Wählen Sie unter Methode die angegebene HTTP-Methode aus. Zum Beispiel GET.
- b. Geben Sie für Pfad den angegebenen Pfad ein. Beispiel, **/items**.
- c. Wählen Sie Route hinzufügen aus.

- d. Wiederholen Sie die vorherigen Schritte, bis Sie alle fünf angegebenen Routen hinzugefügt haben.
3. Wählen Sie Save (Speichern) aus.



## Schritt 4: Konfigurieren Sie Ihre Lambda-Funktionen

Benennen Sie jede der fünf Lambda-Funktionen so, wie sie in der [Referenztable mit den Ressourceneigenschaften](#) angegeben sind.

Um die Lambda-Funktionen zu benennen

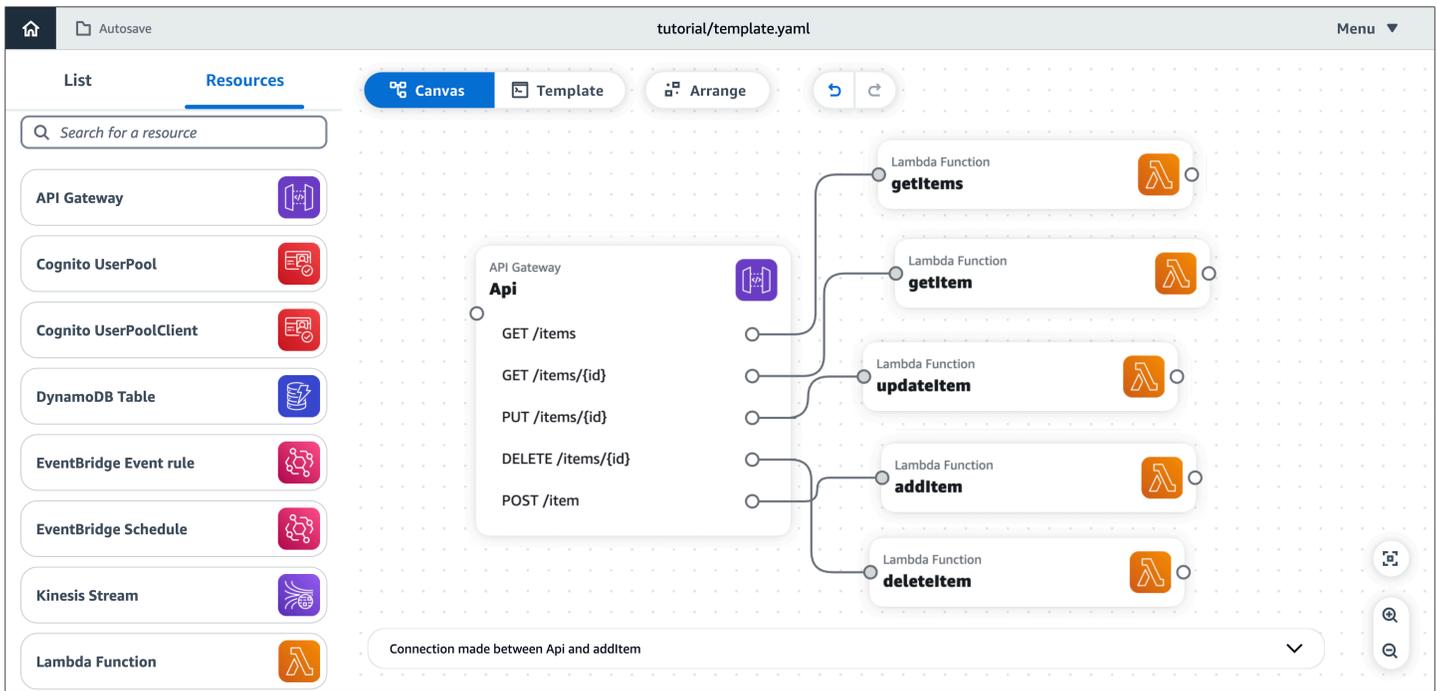
1. Öffnen Sie den Bereich mit den Ressourceneigenschaften einer Lambda-Funktionskarte. Doppelklicken Sie auf die Karte, um das Panel zu öffnen. Oder wählen Sie die Karte und dann Details aus.
2. Geben Sie im Eigenschaftenbereich der Ressource für Logische ID einen bestimmten Funktionsnamen ein. Beispiel, **getItems**.
3. Wählen Sie Save (Speichern) aus.
4. Wiederholen Sie die vorherigen Schritte, bis Sie alle fünf Funktionen benannt haben.

## Schritt 5: Connect deine Karten

Connect jede Route auf Ihrer API-Gateway-Karte mit der zugehörigen Lambda-Funktionskarte, wie in der [Referenztable mit den Ressourceneigenschaften](#) angegeben.

Um Ihre Karten zu verbinden

1. Klicken Sie auf der API-Gateway-Karte auf einen rechten Port und ziehen Sie ihn auf den linken Port der angegebenen Lambda-Funktionskarte. Klicken Sie beispielsweise auf den Anschluss GET /items und ziehen Sie ihn auf den linken Anschluss von GetItems.
2. Wiederholen Sie den vorherigen Schritt, bis Sie alle fünf Routen auf der API-Gateway-Karte mit den entsprechenden Lambda-Funktionskarten verbunden haben.



## Schritt 6: Organisieren Sie die Leinwand

Organisieren Sie die visuelle Leinwand, indem Sie Ihre Lambda-Funktionen gruppieren und alle Karten anordnen.

Um Ihre Funktionen zu gruppieren

1. Halten Sie die Umschalttaste gedrückt und wählen Sie dann jede Lambda-Funktionskarte auf der Leinwand aus.
2. Wählen Sie Gruppe.

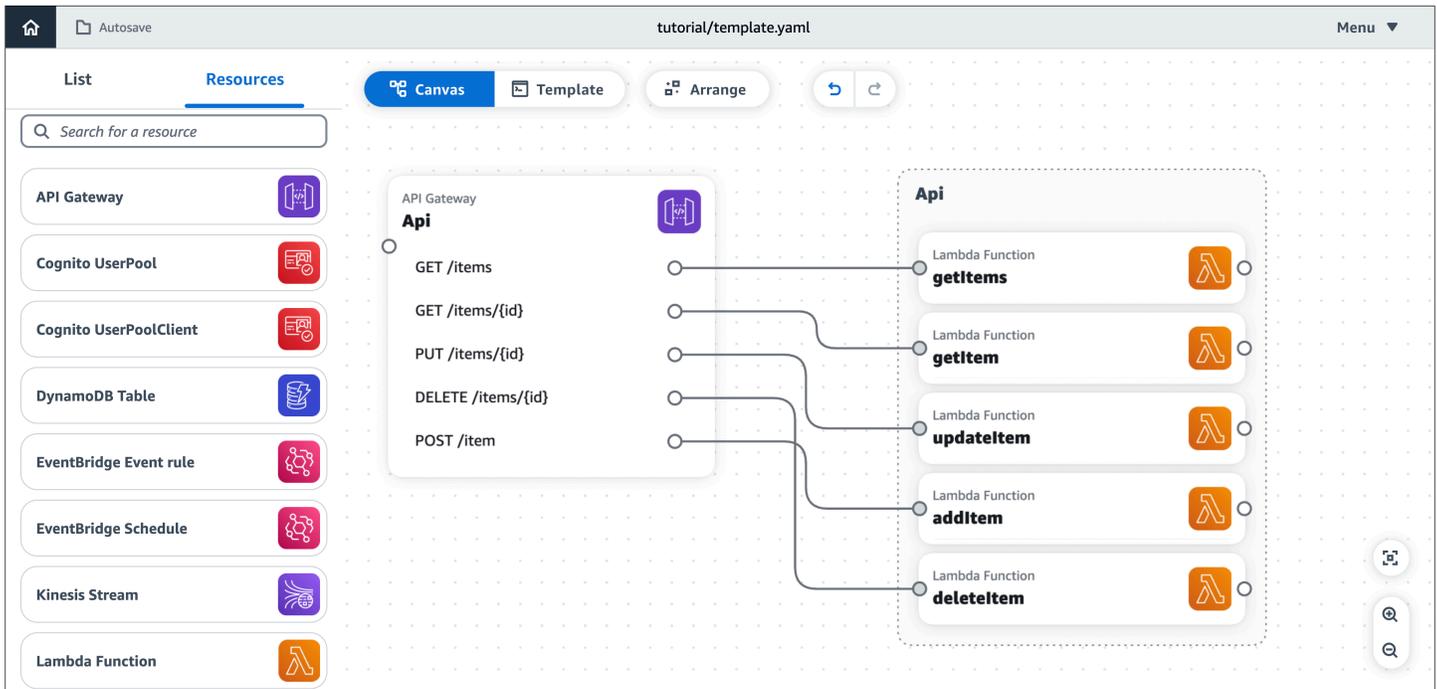
Um deiner Gruppe einen Namen zu geben

1. Doppelklicken Sie auf den oberen Rand der Gruppe, neben dem Gruppennamen (Gruppe).  
Das Fenster mit den Gruppeneigenschaften wird geöffnet.
2. Geben Sie im Bereich Gruppeneigenschaften als Gruppenname den Text ein **API**.
3. Wählen Sie Save (Speichern) aus.

Um Ihre Karten anzuordnen

Wählen Sie auf der Leinwand über dem Hauptansichtsbereich die Option „Anordnen“.

Infrastructure Composer ordnet alle Karten auf der visuellen Leinwand an und richtet sie aus, einschließlich Ihrer neuen Gruppe (API), wie hier gezeigt:

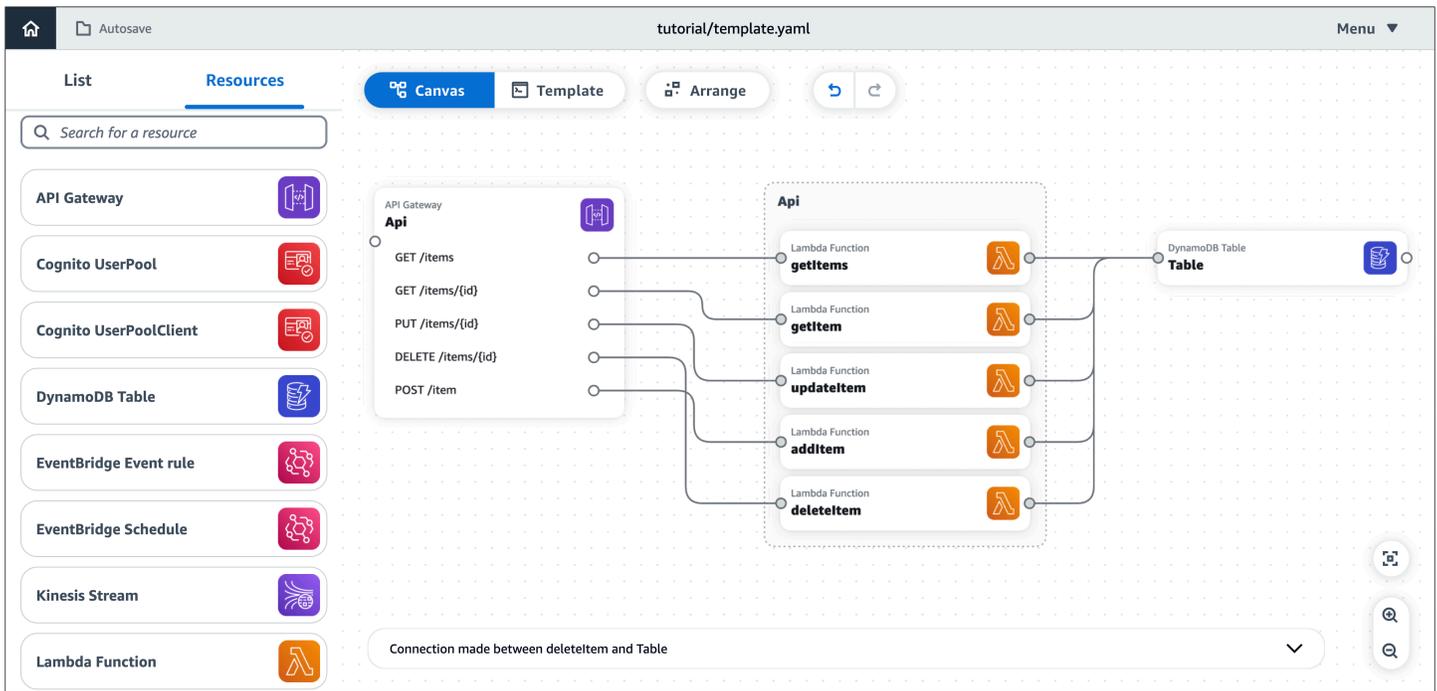


## Schritt 7: DynamoDB-Tabelle hinzufügen und verbinden

Fügen Sie nun Ihrer Anwendungsarchitektur eine DynamoDB-Tabelle hinzu und verbinden Sie sie mit Ihren Lambda-Funktionen.

So fügen Sie eine DynamoDB-Tabelle hinzu und verbinden sie

1. Ziehen Sie aus der Ressourcenpalette (Ressourcen) im Bereich Erweiterte Komponenten eine DynamoDB-Tabellenkarte auf die Arbeitsfläche.
2. Klicken Sie auf den rechten Port auf einer Lambda-Funktionskarte und ziehen Sie ihn auf den linken Port der DynamoDB-Tabellenkarte.
3. Wiederholen Sie den vorherigen Schritt, bis Sie alle fünf Lambda-Funktionskarten mit der DynamoDB-Tabellenkarte verbunden haben.
4. (Optional) Um die Karten auf der Leinwand neu zu organisieren und neu auszurichten, wählen Sie „Anordnen“.

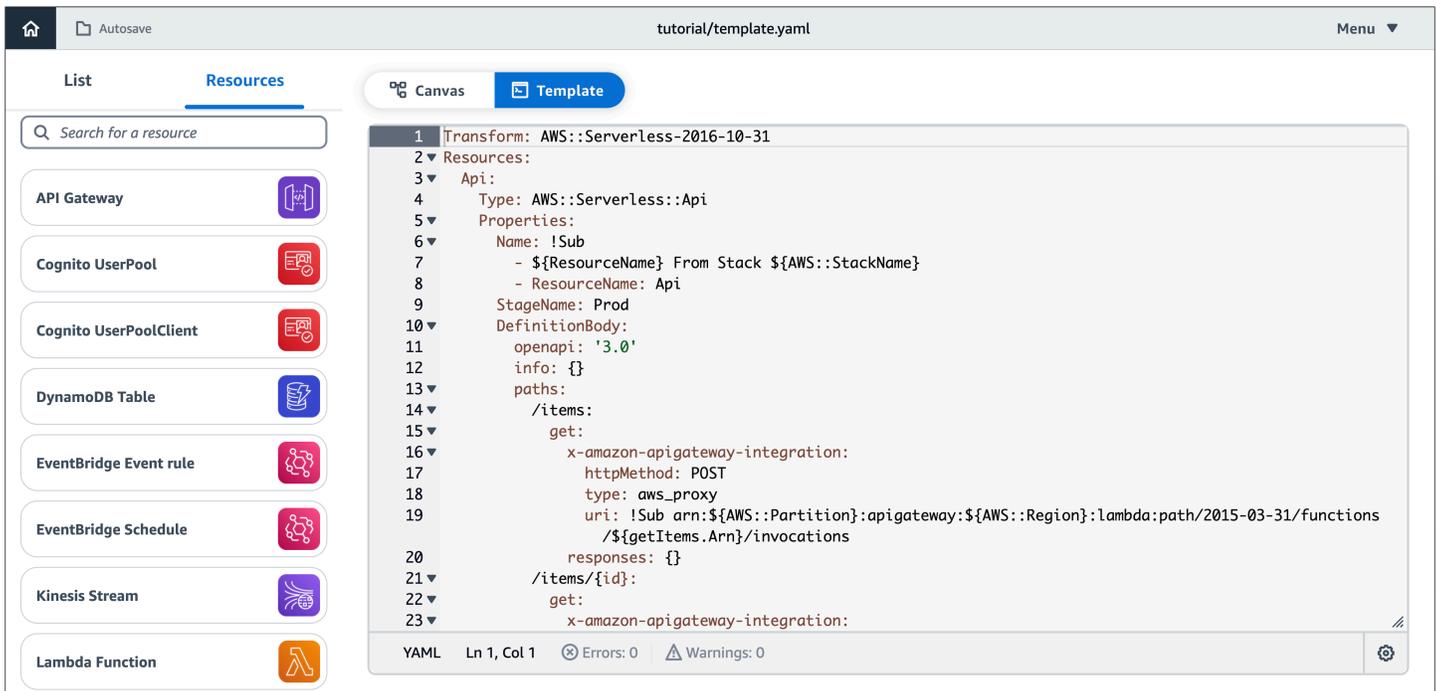


## Schritt 8: Überprüfen Sie Ihre Vorlage AWS CloudFormation

Herzlichen Glückwunsch! Sie haben erfolgreich eine serverlose Anwendung entworfen, die bereit für die Bereitstellung ist. Wählen Sie abschließend Vorlage aus, um die AWS CloudFormation Vorlage zu überprüfen, die Infrastructure Composer automatisch für Sie generiert hat.

In der Vorlage hat Infrastructure Composer Folgendes definiert:

- Die Transform Deklaration, die die Vorlage als Vorlage AWS Serverless Application Model (AWS SAM) spezifiziert. Weitere Informationen finden Sie unter Aufbau der [AWS SAM Vorlage](#) im AWS Serverless Application Model Entwicklerhandbuch.
- Eine `AWS::Serverless::Api` Ressource, die Ihre API-Gateway-REST-API mit ihren fünf Routen spezifiziert.
- Fünf `AWS::Serverless::Function` Ressourcen, die die Konfigurationen Ihrer Lambda-Funktionen spezifizieren, einschließlich ihrer Umgebungsvariablen und Berechtigungsrichtlinien.
- Eine `AWS::DynamoDB::Table` Ressource, die Ihre DynamoDB-Tabelle und ihre Eigenschaften spezifiziert.
- Der Metadata Abschnitt, der Informationen über Ihre Ressourcengruppe (API) enthält. Weitere Informationen zu diesem Abschnitt finden Sie unter [Metadaten](#) im AWS CloudFormation Benutzerhandbuch.



## Schritt 9: Integrieren Sie es in Ihre Entwicklungsworkflows

Verwenden Sie die Vorlagendatei und die Projektverzeichnisse, die Infrastructure Composer für weitere Tests und Bereitstellungen erstellt hat.

- Mit der lokalen Synchronisierung können Sie Infrastructure Composer mit der IDE auf Ihrem lokalen Computer verbinden, um die Entwicklung zu beschleunigen. Weitere Informationen hierzu finden Sie unter [Connect die Infrastructure Composer-Konsole mit Ihrer lokalen IDE](#).
- Mit der lokalen Synchronisierung können Sie die AWS Serverless Application Model Befehlszeilenschnittstelle (AWS SAM CLI) auf Ihrem lokalen Computer verwenden, um Ihre Anwendung zu testen und bereitzustellen. Weitere Informationen hierzu finden Sie unter [Stellen Sie Ihre serverlose Infrastructure Composer-Anwendung in der AWS Cloud bereit](#).

## Nächste Schritte

Sie sind jetzt bereit, Ihre eigenen Anwendungen mit Infrastructure Composer zu erstellen. Ausführliche Informationen zur Verwendung von Infrastructure Composer finden Sie unter [Wie komponiert man in AWS-Infrastruktur-Composer](#). Wenn Sie bereit sind, Ihre Anwendung bereitzustellen, finden Sie weitere Informationen unter [Stellen Sie Ihre serverlose Infrastructure Composer-Anwendung in der AWS Cloud bereit](#).

# Wo Sie Infrastructure Composer verwenden können

Sie können Infrastructure Composer von seiner Konsole aus AWS Toolkit for Visual Studio Code, von und in Infrastructure Composer im CloudFormation Konsolenmodus verwenden. Zwar unterscheidet sich jeder für leicht unterschiedliche Anwendungsfälle, insgesamt handelt es sich jedoch um ähnliche Erfahrungen. Dieser Abschnitt enthält Einzelheiten zu den einzelnen Erlebnissen.

Das Thema [Die AWS-Infrastruktur-Composer Konsole verwenden](#) bietet einen umfassenden Überblick über das Standard-Konsolenerlebnis. Das Thema [CloudFormation Konsolenmodus](#) enthält Einzelheiten zu einer Version von Infrastructure Composer, die in den AWS CloudFormation Stack-Workflow integriert ist. [AWS Toolkit for Visual Studio Code](#) enthält Informationen zum Zugriff auf und zur Verwendung von Infrastructure Composer in VS Code.

## Themen

- [Die AWS-Infrastruktur-Composer Konsole verwenden](#)
- [Verwenden von Infrastructure Composer im CloudFormation Konsolenmodus](#)
- [Verwenden von Infrastructure Composer aus dem AWS Toolkit for Visual Studio Code](#)

# Die AWS-Infrastruktur-Composer Konsole verwenden

Dieser Abschnitt enthält Einzelheiten zum Zugriff auf und zur Verwendung AWS-Infrastruktur-Composer von der Infrastructure Composer-Konsole aus. Dies ist die Standarderfahrung für Infrastructure Composer und eine gute Möglichkeit, sich mit Infrastructure Composer vertraut zu machen. Sie können die Infrastructure Composer-Konsole auch in Ihre lokale IDE integrieren. Details hierzu finden Sie unter [Connect die Infrastructure Composer-Konsole mit Ihrer lokalen IDE](#).

Sie können auch [über das AWS Toolkit in VS Code auf Infrastructure Composer zugreifen](#) und einen [Modus von Infrastructure Composer verwenden, der speziell für die Verwendung in AWS CloudFormation](#) entwickelt wurde.

Allgemeine Dokumentation zur Verwendung von Infrastructure Composer finden Sie unter [Wie komponiert man](#).

## Themen

- [AWS-Infrastruktur-Composer Visuelle Übersicht über die Konsole](#)
- [Verwalten Sie Ihr Projekt von der Infrastructure Composer-Konsole aus](#)

- [Connect die Infrastructure Composer-Konsole mit Ihrer lokalen IDE](#)
- [Webseitenzugriff auf lokale Dateien in Infrastructure Composer zulassen](#)
- [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#)
- [Funktionen von der Lambda-Konsole in Infrastructure Composer importieren](#)
- [Exportieren Sie ein Bild der visuellen Leinwand von Infrastructure Composer](#)

## AWS-Infrastruktur-Composer Visuelle Übersicht über die Konsole

Dieser Abschnitt bietet einen visuellen Überblick über die AWS-Infrastruktur-Composer Konsole.

Themen

- [Homepage](#)
- [Visueller Designer und visuelle Leinwand](#)

## Homepage

Das folgende Bild zeigt die Startseite in der Infrastructure Composer-Konsole:

The screenshot shows the AWS Infrastructure Composer console homepage. On the left is a navigation sidebar with the following items: 'Infrastructure Composer', 'Canvas', 'Documentation', 'AWS Toolkit for VS Code', and 'Import a Lambda Function'. The main content area has a breadcrumb 'Infrastructure Composer > Home' and a 'New!' announcement: 'Infrastructure Composer is now available as part of the AWS Toolkit for Visual Studio Code. [Learn more](#)'. Below the announcement are three buttons: 'Download VS Code Extension', 'Open demo', and 'Create project'. The 'About' section describes the tool's purpose and includes a 'How it works' list:

- 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

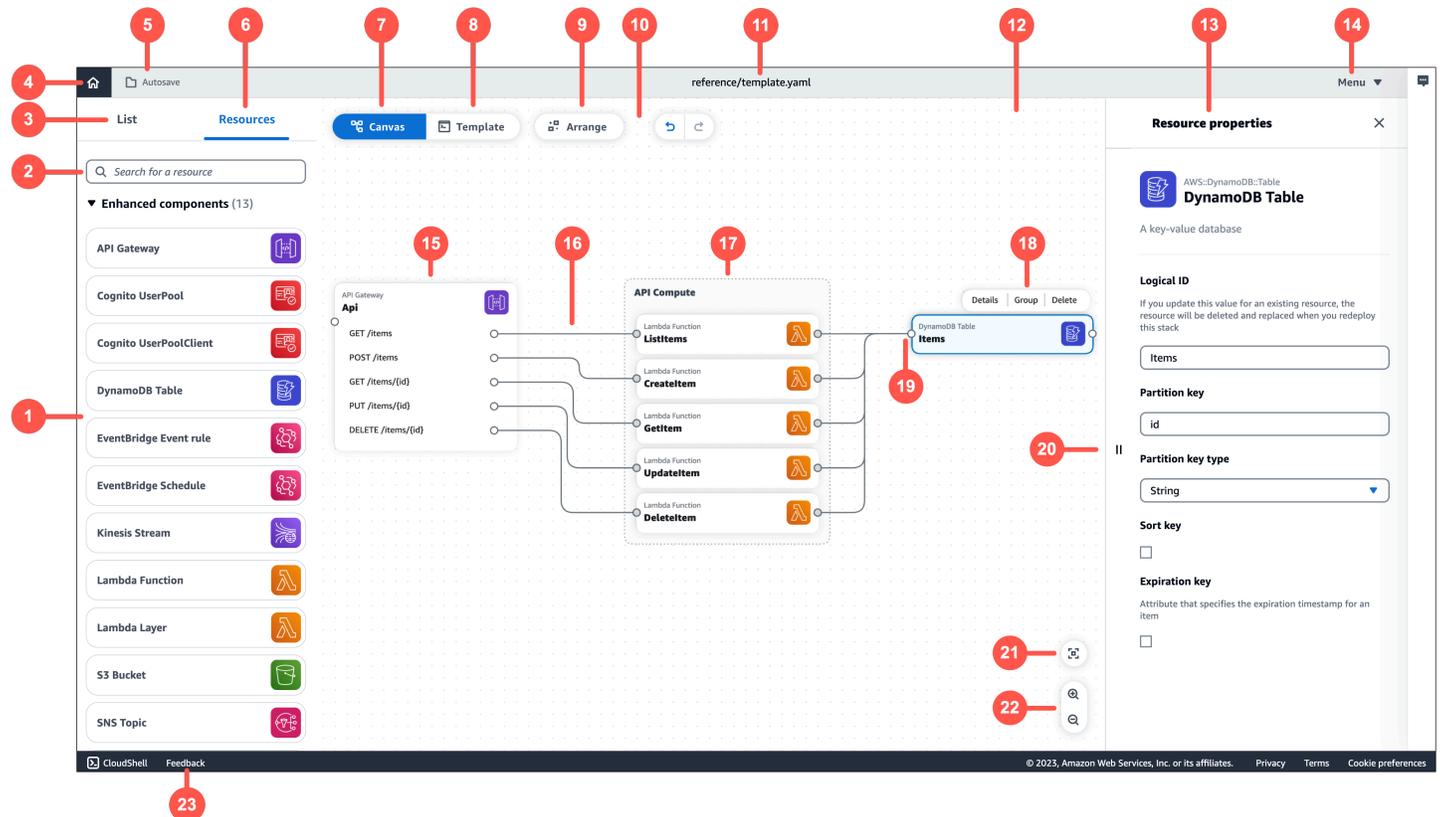
At the bottom, there is a 'Start building' section with a visual canvas showing a project diagram with various AWS resources connected.

1. Dokumentation — Gehen Sie zur Infrastructure Composer-Dokumentation.
2. Canvas — Gehen Sie zum Canvas und erstellen oder laden Sie ein Projekt.
3. Demo — Öffnen Sie die Infrastructure Composer-Demoanwendung.

- Projekt erstellen — Erstellen oder laden Sie ein Projekt.
- Mit der Erstellung beginnen — Schnelllinks, um mit der Erstellung einer Anwendung zu beginnen.
- Feedback — Klicken Sie hier, um Feedback einzureichen.

## Visueller Designer und visuelle Leinwand

Das folgende Bild zeigt den visuellen Designer und die visuelle Leinwand von Infrastructure Composer:



- Ressourcenpalette — Zeigt Karten an, mit denen Sie entwerfen können.
- Suchleiste für Ressourcen — Suchen Sie nach Karten, die Sie der Arbeitsfläche hinzufügen können.
- Liste — Zeigt eine Strukturansicht Ihrer Anwendungsressourcen an.
- Home — Wählen Sie hier, um zur Infrastructure Composer-Startseite zu gelangen.
- Speicherstatus — Gibt an, ob Infrastructure Composer-Änderungen auf Ihrem lokalen Computer gespeichert wurden. Zu den Staaten gehören:
  - Autosave — Die lokale Synchronisierung ist aktiviert und Ihr Projekt wird automatisch synchronisiert und gespeichert.

- Änderungen gespeichert — Ihre Anwendungsvorlage wird auf Ihrem lokalen Computer gespeichert.
  - Nicht gespeicherte Änderungen — Ihre Anwendungsvorlage enthält Änderungen, die nicht auf Ihrem lokalen Computer gespeichert wurden.
6. Ressourcen — Zeigt die Ressourcenpalette an.
7. Leinwand — Zeigt die Leinwandansicht Ihrer Anwendung im Hauptansichtsbereich an.
8. Vorlage — Zeigt die Vorlagenansicht Ihrer Anwendung im Hauptansichtsbereich an.
9. Anordnen — Ordnet Ihre Anwendungsarchitektur auf der Arbeitsfläche an.
10. Rückgängig machen und wiederherstellen — Führen Sie Aktionen zum Rückgängigmachen und Wiederherstellen durch, sofern dies unterstützt wird.
11. Vorlagename — Gibt den Namen der Vorlage an, die Sie entwerfen.
12. Hauptansichtsbereich — Zeigt je nach Ihrer Auswahl entweder die Leinwand oder die Vorlage an.
13. Bedienfeld „Ressourceneigenschaften“ — Zeigt die relevanten Eigenschaften für die Karte an, die auf der Arbeitsfläche ausgewählt wurde. Dieses Fenster ist dynamisch. Die angezeigten Eigenschaften ändern sich, wenn Sie Ihre Karte konfigurieren.
14. Menü — Bietet allgemeine Optionen wie die folgenden:
- Erstellen eines Projekts
  - Öffnen Sie eine Vorlagendatei oder ein Projekt
  - Speichern Sie eine Vorlagendatei
  - [Aktivieren Sie die lokale Synchronisierung](#)
  - [Leinwand exportieren](#)
  - Holen Sie sich Unterstützung
  - Tastenkombination
15. Karte — Zeigt eine Ansicht Ihrer Karte auf der Leinwand an.
16. Linie — Stellt eine Verbindung zwischen Karten dar.
17. Gruppe — Gruppiert ausgewählte Karten zur visuellen Organisation zusammen.
18. Kartenaktionen — Bietet Aktionen, die Sie mit Ihrer Karte ausführen können.
- a. Details — Öffnet den Eigenschaftenbereich der Ressource.
  - b. Gruppe — Gruppiert die ausgewählten Karten zusammen.
  - c. Löschen — Löscht die Karte von Ihrer Leinwand.
19. Port — Verbindungspunkte zu anderen Karten.

20Eigenschaftsfelder für Ressourcen — Ein kuratierter Satz von Eigenschaftsfeldern, die Sie für Ihre Karten konfigurieren können.

21Neu zentrieren — Zentrieren Sie Ihr Anwendungsdiagramm auf der visuellen Leinwand neu.

22Zoom — Zoomen Sie auf Ihrer Leinwand hinein und heraus.

23Feedback — Klicken Sie hier, um Feedback einzureichen.

## Verwalten Sie Ihr Projekt von der Infrastructure Composer-Konsole aus

Dieses Thema enthält Anleitungen zu den grundlegenden Aufgaben, die Sie ausführen, um Ihr Projekt von der Infrastructure Composer-Konsole aus zu verwalten. Dazu gehören allgemeine Aufgaben wie das Erstellen eines neuen Projekts, das Speichern eines Projekts und das Importieren eines Projekts oder einer Vorlage. Sie können auch ein vorhandenes Projekt laden, wenn Sie den [lokalen Synchronisierungsmodus](#) aktivieren. Nachdem Sie den lokalen Synchronisierungsmodus aktiviert haben, können Sie Folgendes tun:

- Erstellen Sie ein neues Projekt, das aus einer Startvorlage und einer Ordnerstruktur besteht.
- Laden Sie ein vorhandenes Projekt, indem Sie einen übergeordneten Ordner auswählen, der Ihre Projektvorlage und Ihre Dateien enthält.
- Verwenden Sie Infrastructure Composer, um Ihre Vorlagen und Ordner zu verwalten

Im lokalen Synchronisierungsmodus speichert Infrastructure Composer die Vorlagen- und Ordneränderungen Ihres Projekts automatisch auf Ihrem lokalen Computer. Wenn Ihr Browser den lokalen Synchronisierungsmodus nicht unterstützt oder wenn Sie Infrastructure Composer lieber ohne aktivierten lokalen Synchronisierungsmodus verwenden möchten, können Sie eine neue Vorlage erstellen oder eine vorhandene Vorlage laden. Um Änderungen zu speichern, müssen Sie die Vorlage auf Ihren lokalen Computer exportieren.

### Note

Infrastructure Composer unterstützt Anwendungen, die aus den folgenden Komponenten bestehen:

- Eine AWS CloudFormation AWS Serverless Application Model Oder-Vorlage, die Ihren Infrastrukturcode definiert.

- Eine Ordnerstruktur, die Ihre Projektdateien wie Lambda-Funktionscode, Konfigurationsdateien und Build-Ordner organisiert.

## Themen

- [Erstellen Sie ein neues Projekt in der Infrastructure Composer-Konsole](#)
- [Importieren Sie einen vorhandenen Projektordner in die Infrastructure Composer-Konsole](#)
- [Importieren Sie eine vorhandene Projektvorlage in die Infrastructure Composer-Konsole](#)
- [Speichern Sie eine vorhandene Projektvorlage in der Infrastructure Composer-Konsole](#)

## Erstellen Sie ein neues Projekt in der Infrastructure Composer-Konsole

Wenn Sie ein neues Projekt erstellen, generiert Infrastructure Composer eine Startvorlage. Während Sie Ihre Anwendung auf der Arbeitsfläche entwerfen, wird Ihre Vorlage geändert. Um Ihre Arbeit zu speichern, müssen Sie Ihre Vorlage exportieren oder den lokalen Synchronisierungsmodus aktivieren.

So erstellen Sie ein neues Projekt

1. Melden Sie sich bei der [Infrastructure Composer-Konsole](#) an.
2. Wählen Sie auf der Startseite die Option Projekt erstellen aus.

### Note

Sie können auch ein vorhandenes in Infrastructure Composer laden, müssen aber zuerst [den lokalen Synchronisierungsmodus aktivieren](#). Nach der Aktivierung finden Sie Informationen [Lädt ein vorhandenes Infrastructure Composer-Projekt mit aktivierter lokaler Synchronisierung](#) zum Laden eines vorhandenen Projekts.

## Importieren Sie einen vorhandenen Projektordner in die Infrastructure Composer-Konsole

Im lokalen Synchronisierungsmodus können Sie den übergeordneten Ordner eines vorhandenen Projekts importieren. Wenn Ihr Projekt mehrere Vorlagen enthält, können Sie die Vorlage auswählen, die geladen werden soll.

## Um ein vorhandenes Projekt von der Startseite zu importieren

1. Melden Sie sich bei der [Infrastructure Composer-Konsole](#) an.
2. Wählen Sie auf der Startseite die Option CloudFormation Vorlage laden aus.
3. Wählen Sie als Projektspeicherort die Option Ordner auswählen aus. Wählen Sie den übergeordneten Ordner Ihres Projekts aus und klicken Sie auf Auswählen.

### Note

Wenn Sie diese Aufforderung nicht erhalten, unterstützt Ihr Browser möglicherweise nicht die Dateisystem-Zugriffs-API, die für den lokalen Synchronisierungsmodus erforderlich ist. Weitere Informationen finden Sie unter [Webseitenzugriff auf lokale Dateien in Infrastructure Composer zulassen](#).

4. Wenn Sie von Ihrem Browser dazu aufgefordert werden, wählen Sie Dateien anzeigen aus.
5. Wählen Sie für Vorlagendatei Ihre Vorlage aus der Drop-down-Liste aus. Wenn Ihr Projekt eine einzige Vorlage enthält, wählt Infrastructure Composer diese automatisch für Sie aus.
6. Wählen Sie Create (Erstellen) aus.

## Um ein vorhandenes Projekt aus der Arbeitsfläche zu importieren

1. Wählen Sie auf der Leinwand „Menü“, um das Menü zu öffnen.
2. Wählen Sie im Bereich Öffnen die Option Projektordner aus.

### Note

Wenn die Option Projektordner nicht verfügbar ist, unterstützt Ihr Browser möglicherweise nicht die Dateisystem-Zugriffs-API, die für den lokalen Synchronisierungsmodus erforderlich ist. Weitere Informationen finden Sie unter [Webseitenzugriff auf lokale Dateien in Infrastructure Composer zulassen](#).

3. Wählen Sie als Speicherort für das Projekt die Option Ordner auswählen aus. Wählen Sie den übergeordneten Ordner Ihres Projekts aus und klicken Sie auf Auswählen.
4. Wenn Sie von Ihrem Browser dazu aufgefordert werden, wählen Sie Dateien anzeigen.
5. Wählen Sie für Vorlagendatei Ihre Vorlage aus der Drop-down-Liste aus. Wenn Ihr Projekt eine einzige Vorlage enthält, wählt Infrastructure Composer diese automatisch für Sie aus.

## 6. Wählen Sie Create (Erstellen) aus.

Wenn Sie einen vorhandenen Projektordner importieren, aktiviert Infrastructure Composer den lokalen Synchronisierungsmodus. Änderungen an der Vorlage oder den Dateien Ihres Projekts werden automatisch auf Ihrem lokalen Computer gespeichert.

## Importieren Sie eine vorhandene Projektvorlage in die Infrastructure Composer-Konsole

Wenn Sie ein vorhandenes Objekt AWS CloudFormation oder eine AWS SAM Vorlage importieren, generiert Infrastructure Composer automatisch eine Visualisierung Ihrer Anwendungsarchitektur auf der Arbeitsfläche.

Sie können eine Projektvorlage von Ihrem lokalen Computer importieren.

Um eine bestehende Projektvorlage zu importieren

1. Melden Sie sich bei der [Infrastructure Composer-Konsole](#) an.
2. Wählen Sie Projekt erstellen, um eine leere Leinwand zu öffnen.
3. Wählen Sie „Menü“, um das Menü zu öffnen.
4. Wählen Sie im Bereich Öffnen die Option Vorlagendatei aus.
5. Wählen Sie Ihre Vorlage aus und wählen Sie Öffnen.

Um Änderungen an Ihrer Vorlage zu speichern, müssen Sie Ihre Vorlage exportieren oder den lokalen Synchronisierungsmodus aktivieren.

## Speichern Sie eine vorhandene Projektvorlage in der Infrastructure Composer-Konsole

Wenn Sie den lokalen Synchronisierungsmodus nicht verwenden, müssen Sie Ihre Vorlage exportieren, um Ihre Änderungen zu speichern. Wenn Sie den lokalen Synchronisierungsmodus aktiviert haben, ist das manuelle Speichern Ihrer Vorlage nicht erforderlich. Änderungen werden automatisch auf Ihrem lokalen Computer gespeichert.

Um eine bestehende Projektvorlage zu speichern

1. Wählen Sie auf der Infrastructure Composer-Arbeitsfläche „Menü“, um das Menü zu öffnen.
2. Wählen Sie im Bereich Speichern die Option Vorlagendatei speichern aus.

3. Geben Sie einen Namen für Ihre Vorlage ein.
4. Wählen Sie einen Speicherort für Ihre Vorlage aus.
5. Wählen Sie Save (Speichern) aus.

## Connect die Infrastructure Composer-Konsole mit Ihrer lokalen IDE

Verwenden Sie den lokalen Synchronisierungsmodus, um die Infrastructure Composer-Konsole mit Ihrer lokalen integrierten Entwicklungsumgebung (IDE) zu verbinden. In diesem Modus werden Daten automatisch auf Ihrem lokalen Computer synchronisiert und gespeichert. Weitere Informationen zum lokalen Synchronisierungsmodus finden Sie unter [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#). Anweisungen zur Verwendung des lokalen Synchronisierungsmodus finden Sie unter [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#).

### Note

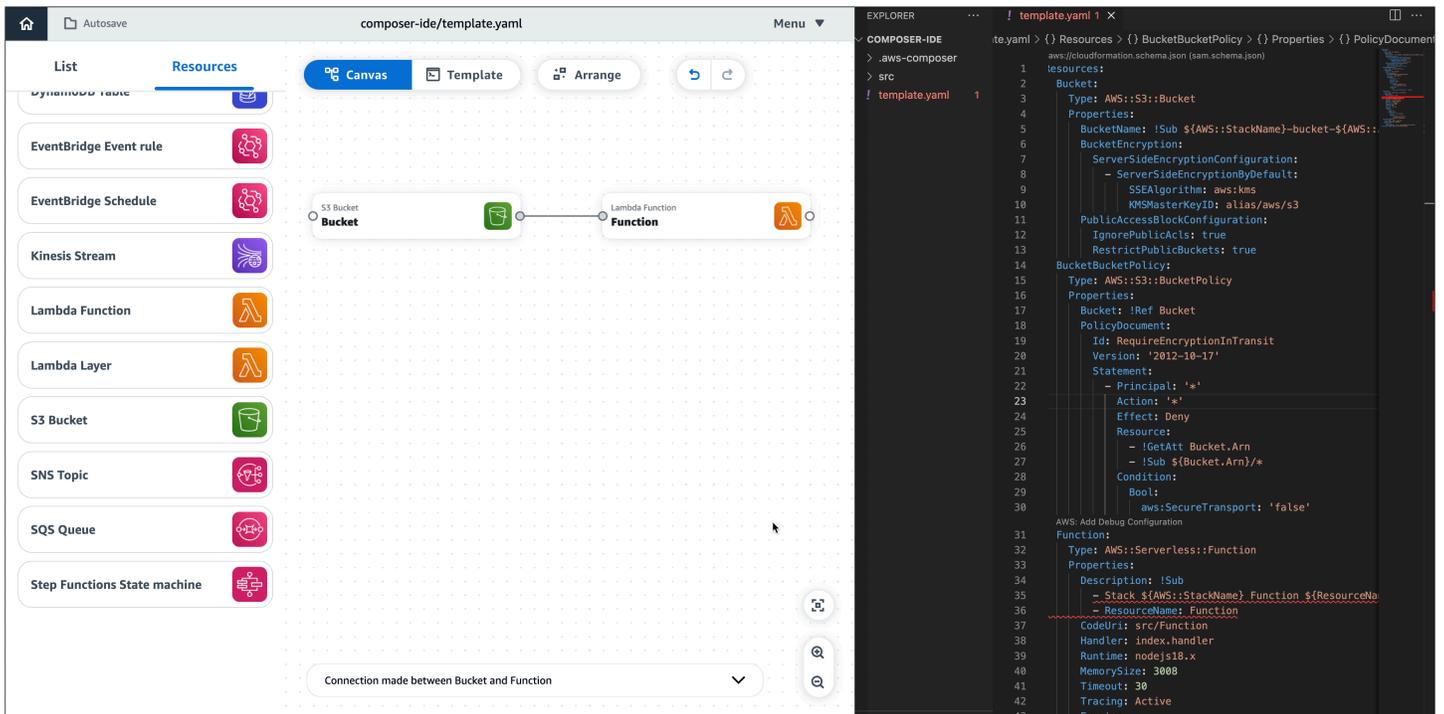
Die Option Lokale Synchronisierung aktivieren ist nicht in jedem Browser verfügbar. Es ist in Google Chrome und Microsoft Edge verfügbar.

## Vorteile der Verwendung von Infrastructure Composer mit Ihrer lokalen IDE

Während Sie in Infrastructure Composer entwerfen, werden Ihre lokale Vorlage und Ihr Projektverzeichnis automatisch synchronisiert und gespeichert.

Sie können Ihre lokale IDE verwenden, um Änderungen anzuzeigen und Ihre Vorlagen zu ändern. Änderungen, die Sie lokal vornehmen, werden automatisch mit Infrastructure Composer synchronisiert.

Sie können lokale Tools wie die AWS Serverless Application Model Befehlszeilenschnittstelle (AWS SAM CLI) verwenden, um Ihre Anwendung zu erstellen, zu testen, bereitzustellen und vieles mehr. Das folgende Beispiel zeigt, wie Sie Ressourcen per Drag-and-Drop auf die visuelle Arbeitsfläche von Infrastructure Composer ziehen können, wodurch wiederum Markup in Ihrer AWS SAM Vorlage in Ihrer lokalen IDE erstellt wird.



## Integrieren Sie Infrastructure Composer in Ihre lokale IDE

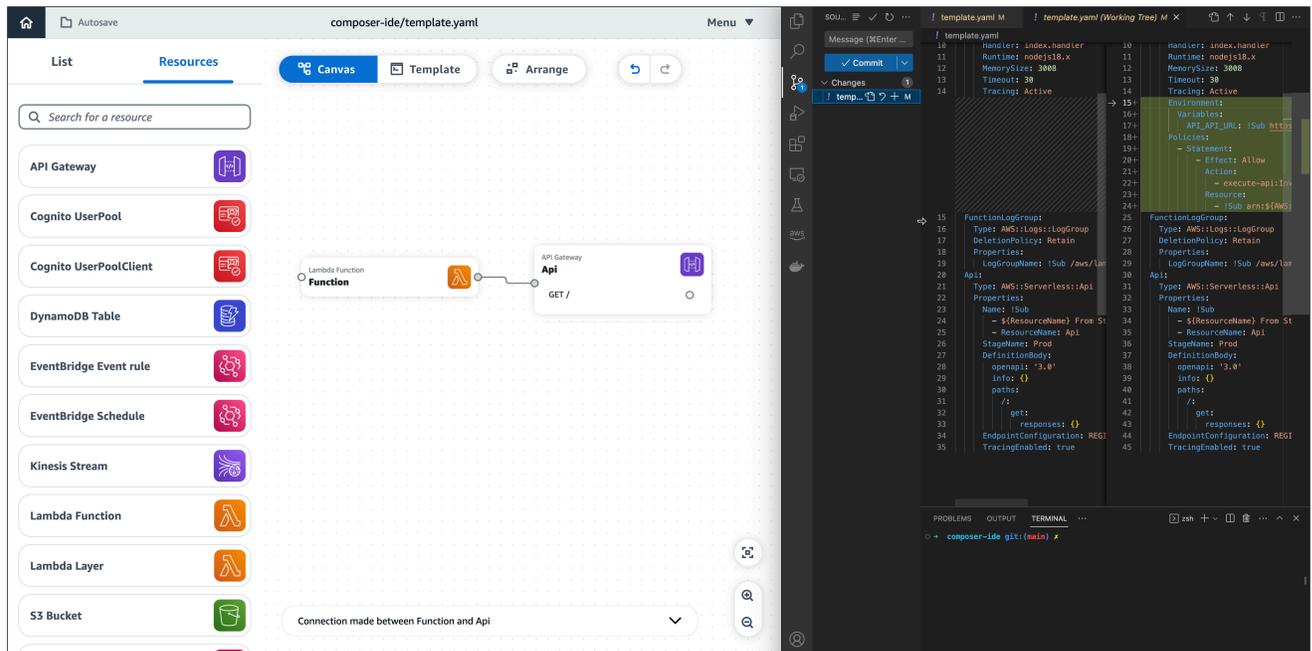
Um Infrastructure Composer in Ihre lokale IDE zu integrieren

1. Erstellen oder laden Sie in Infrastructure Composer ein Projekt und aktivieren Sie die lokale Synchronisierung, indem Sie oben rechts auf dem Bildschirm auf die Menüschaltfläche klicken und Lokale Synchronisierung aktivieren wählen.

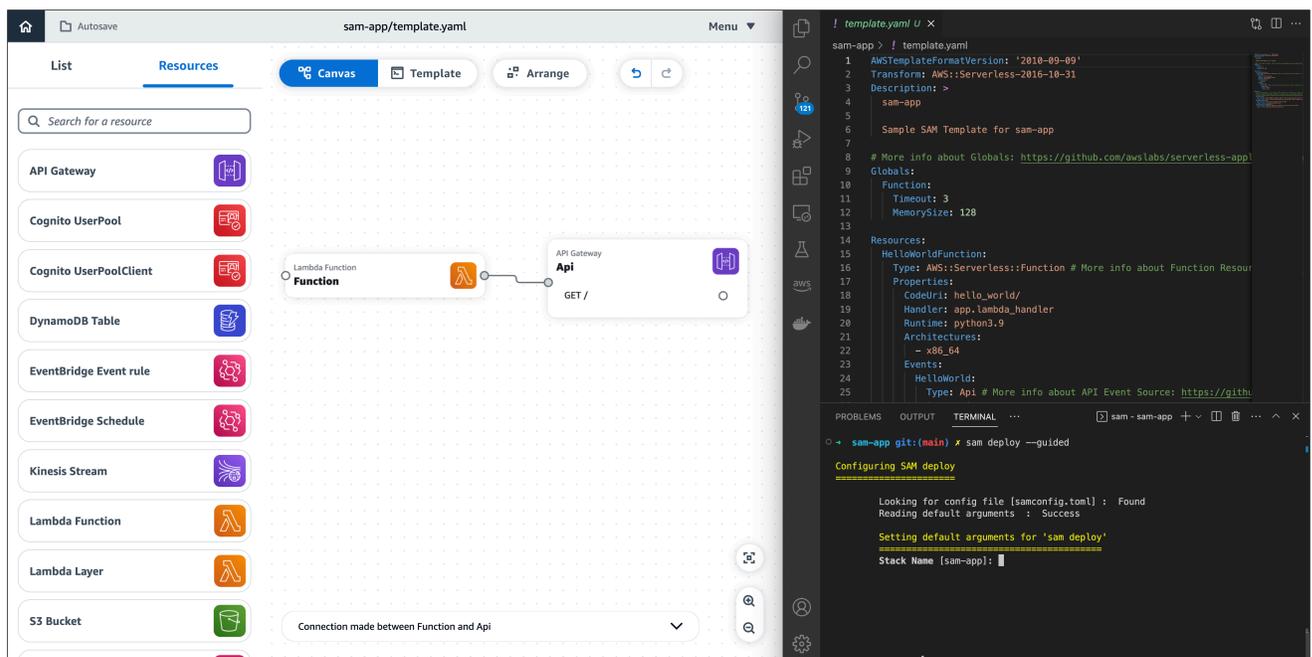
### Note

Die Option Lokale Synchronisierung aktivieren ist nicht in jedem Browser verfügbar. Es ist in Google Chrome und Microsoft Edge verfügbar.

2. Öffnen Sie in Ihrer lokalen IDE denselben Projektordner wie Infrastructure Composer.
3. Verwenden Sie Infrastructure Composer mit Ihrer lokalen IDE. In Infrastructure Composer vorgenommene Updates werden automatisch mit Ihrem lokalen Computer synchronisiert. Hier sind einige Beispiele dafür, was Sie tun können:
  - a. Verwenden Sie das Versionskontrollsystem Ihrer Wahl, um Updates zu verfolgen, die von Infrastructure Composer durchgeführt werden.



- b. Verwenden Sie die AWS SAM CLI lokal, um Ihre Anwendung zu erstellen, zu testen, bereitzustellen und vieles mehr. Weitere Informationen hierzu finden Sie unter [Stellen Sie Ihre serverlose Infrastructure Composer-Anwendung in der AWS Cloud bereit](#).



## Webseitenzugriff auf lokale Dateien in Infrastructure Composer zulassen

Die Infrastructure Composer-Konsole unterstützt den [lokalen Synchronisierungsmodus](#) und [Importfunktionen aus der Lambda-Konsole](#). Um diese Funktionen nutzen zu können, ist ein

Webbrowser erforderlich, der die File System Access API unterstützt. Jede aktuelle Version von Google Chrome und Microsoft Edge unterstützt alle Funktionen der File System Access API und kann im lokalen Synchronisierungsmodus in Infrastructure Composer verwendet werden.

Mit der File System Access API können Webseiten auf Ihr lokales Dateisystem zugreifen, um Dateien zu lesen, zu schreiben oder zu speichern. Diese Funktion ist standardmäßig deaktiviert und erfordert Ihre Zustimmung durch eine visuelle Aufforderung, um sie zuzulassen. Einmal gewährt, bleibt dieser Zugriff für die Dauer der Browsersitzung Ihrer Webseite bestehen.

Weitere Informationen zur File System Access API finden Sie unter:

- Die [Dateisystem-Zugriffs-API](#) in den MDN-Webdokumenten.
- [Die Dateisystem-Zugriffs-API: Vereinfachung des Zugriffs auf lokale Dateien auf](#) der Web.dev-Website.

## lokaler Synchronisierungsmodus

Im lokalen Synchronisierungsmodus können Sie Ihre Vorlagendateien und Projektordner automatisch synchronisieren und lokal speichern, während Sie in Infrastructure Composer entwerfen. Um diese Funktion nutzen zu können, ist ein Webbrowser erforderlich, der die File System Access API unterstützt.

## Data Infrastructure Composer erhält Zugriff auf

Infrastructure Composer erhält Lese- und Schreibzugriff auf den Projektordner, den Sie zulassen, zusammen mit allen untergeordneten Ordnern dieses Projektordners. Dieser Zugriff wird verwendet, um alle Vorlagendateien, Projektordner und Sicherungsverzeichnisse zu erstellen, zu aktualisieren und zu speichern, die während des Entwurfs generiert werden. Daten, auf die Infrastructure Composer zugreift, werden nicht für andere Zwecke verwendet und nicht außerhalb Ihres lokalen Dateisystems gespeichert.

### Zugriff auf sensible Daten

Die Dateisystem-Zugriffs-API schließt den Zugriff auf bestimmte Verzeichnisse, die vertrauliche Daten enthalten können, aus oder beschränkt ihn. Ein Fehler tritt auf, wenn Sie eines dieser Verzeichnisse für die Verwendung im lokalen Synchronisierungsmodus von Infrastructure Composer auswählen. Sie können ein anderes lokales Verzeichnis auswählen, mit dem Sie eine Verbindung herstellen möchten, oder Infrastructure Composer im Standardmodus verwenden, wobei die lokale Synchronisierung deaktiviert ist.

Weitere Informationen, einschließlich Beispielen für vertrauliche Verzeichnisse, finden Sie unter [Benutzer, die Zugriff auf mehr oder vertraulichere Dateien gewähren, als sie beabsichtigt hatten](#), im W3C Draft Community Group Report.

Wenn Sie verwenden Windows Subsystem for Linux (WSL), schließt die File System Access API den Zugriff auf das gesamte Linux Verzeichnis aufgrund seiner Position in Ihrem Windows System. Sie können Infrastructure Composer mit deaktivierter lokaler Synchronisierung verwenden oder eine Lösung für die Synchronisierung von Projektdateien aus Ihrem WSL Verzeichnis in ein Arbeitsverzeichnis in Windows. Verwenden Sie dann den lokalen Synchronisierungsmodus von Infrastructure Composer mit Ihrem Windows Verzeichnis.

## Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole

Dieser Abschnitt enthält Informationen zur Verwendung des lokalen Synchronisierungsmodus von Infrastructure Composer, um Ihr Projekt automatisch zu synchronisieren und auf Ihrem lokalen Computer zu speichern.

Wir empfehlen Ihnen, die lokale Synchronisierung aus den folgenden Gründen zu verwenden:

Sie können die lokale Synchronisierung für ein neues Projekt aktivieren oder ein vorhandenes Projekt laden, bei dem die lokale Synchronisierung aktiviert ist.

- Standardmäßig müssen Sie Ihre Anwendungsvorlage während des Entwurfs manuell speichern. Verwenden Sie die lokale Synchronisierung, um Ihre Anwendungsvorlage automatisch auf Ihrem lokalen Computer zu speichern, wenn Sie Änderungen vornehmen.
- Local Sync verwaltet Ihre Projektordner, Backup-Ordner und [unterstützte externe Dateien](#) und synchronisiert sie automatisch mit Ihrem lokalen Computer.
- Wenn Sie die lokale Synchronisierung verwenden, können Sie Infrastructure Composer mit Ihrer lokalen IDE verbinden, um die Entwicklung zu beschleunigen. Weitere Informationen hierzu finden Sie unter [Connect die Infrastructure Composer-Konsole mit Ihrer lokalen IDE](#).

### Welcher lokale Synchronisierungsmodus speichert

Der lokale Synchronisierungsmodus synchronisiert automatisch Folgendes und speichert es auf Ihrem lokalen Computer:

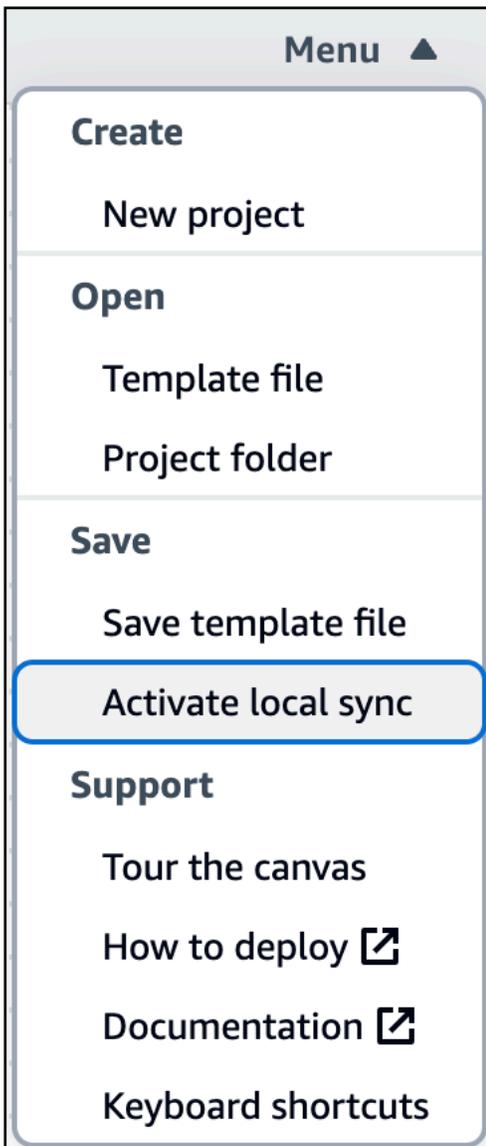
- Anwendungsvorlagendatei — Die Vorlage AWS CloudFormation oder AWS Serverless Application Model (AWS SAM), die Ihre Infrastruktur als Code (IaC) enthält.
- Projektordner — Eine allgemeine Verzeichnisstruktur, die Ihre AWS Lambda Funktionen organisiert.
- Backup-Verzeichnis — Ein Backup-Verzeichnis mit dem Namen `.aws-composer`, das im Stammverzeichnis Ihres Projektverzeichnisses erstellt wurde. Dieses Verzeichnis enthält eine Sicherungskopie Ihrer Anwendungsvorlagendatei und Ihrer Projektordner.
- Externe Dateien — Unterstützte externe Dateien, die Sie in Infrastructure Composer verwenden können. Weitere Informationen hierzu finden Sie unter [Verweisen auf externe Dateien in Infrastructure Composer](#).

## Browser-Anforderungen

Für den lokalen Synchronisierungsmodus ist ein Browser erforderlich, der die File System Access API unterstützt. Weitere Informationen finden Sie unter [Webseitenzugriff auf lokale Dateien in Infrastructure Composer zulassen](#).

## Lokaler Synchronisierungsmodus wird aktiviert

Der lokale Synchronisierungsmodus ist standardmäßig deaktiviert. Sie können den lokalen Synchronisierungsmodus über das Infrastructure Composer-Menü aktivieren.



Anweisungen zur Aktivierung der lokalen Synchronisierung und vorhandener Ladeprojekte finden Sie in den folgenden Themen:.

- [Aktivieren Sie die lokale Synchronisierung in Infrastructure Composer](#)
- [Lädt ein vorhandenes Infrastructure Composer-Projekt mit aktivierter lokaler Synchronisierung](#)

## Aktivieren Sie die lokale Synchronisierung in Infrastructure Composer

Gehen Sie wie folgt vor, um die lokale Synchronisierung zu aktivieren:

1. Wählen Sie auf der Infrastructure [Composer-Startseite](#) die Option Projekt erstellen aus.

2. Wählen Sie im Infrastructure Composer-Menü die Option Lokale Synchronisierung aktivieren aus.
3. Drücken Sie für den Speicherort des Projekts auf Ordner auswählen und wählen Sie ein Verzeichnis aus. Hier speichert und synchronisiert Infrastructure Composer Ihre Vorlagendateien und Ordner, während Sie entwerfen.

 Note

Der Projektstandort darf keine vorhandene Anwendungsvorlage enthalten.

4. Wenn Sie aufgefordert werden, den Zugriff zu gewähren, wählen Sie Dateien anzeigen aus.
5. Drücken Sie auf Aktivieren. Wenn Sie aufgefordert werden, die Änderungen zu speichern, wählen Sie Änderungen speichern.

Wenn diese Option aktiviert ist, wird die Anzeige für automatisches Speichern im oberen linken Bereich Ihrer Leinwand angezeigt.

## Lädt ein vorhandenes Infrastructure Composer-Projekt mit aktivierter lokaler Synchronisierung

Gehen Sie wie folgt vor, um ein vorhandenes Projekt mit aktivierter lokaler Synchronisierung zu laden:

1. Wählen Sie auf der Infrastructure [Composer-Startseite](#) die Option AWS CloudFormation Vorlage laden aus.
2. Wählen Sie im Infrastructure Composer-Menü Öffnen > Projektordner aus.
3. Drücken Sie unter Projektspeicherort auf Ordner auswählen und wählen Sie den Stammordner Ihres Projekts aus.
4. Wenn Sie aufgefordert werden, den Zugriff zuzulassen, wählen Sie Dateien anzeigen aus.
5. Wählen Sie unter Vorlagendatei Ihre Anwendungsvorlage aus und klicken Sie auf Erstellen.
6. Wenn Sie aufgefordert werden, die Änderungen zu speichern, wählen Sie Änderungen speichern aus.

Wenn diese Option aktiviert ist, wird die Anzeige für automatisches Speichern im oberen linken Bereich Ihrer Leinwand angezeigt.

## Funktionen von der Lambda-Konsole in Infrastructure Composer importieren

Infrastructure Composer bietet eine Integration mit der AWS Lambda Konsole. Sie können eine Lambda-Funktion von der Lambda-Konsole in die Infrastructure Composer-Konsole importieren. Verwenden Sie dann die Infrastructure Composer-Canvas, um Ihre Anwendungsarchitektur weiter zu entwerfen.

- Für diese Integration ist ein Browser erforderlich, der die File System Access API unterstützt. Weitere Informationen finden Sie unter [Webseitenzugriff auf lokale Dateien in Infrastructure Composer zulassen](#).
- Wenn Sie Ihre Lambda-Funktion in Infrastructure Composer importieren, müssen Sie den lokalen Synchronisierungsmodus aktivieren, um alle Änderungen zu speichern. Weitere Informationen finden Sie unter [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#).

Informationen zu den ersten Schritten mit dieser Integration finden Sie unter [Using AWS Lambda with AWS-Infrastruktur-Composer](#) im AWS Lambda Developer Guide.

## Exportieren Sie ein Bild der visuellen Leinwand von Infrastructure Composer

In diesem Thema wird die Canvas-Funktion für den AWS-Infrastruktur-Composer Konsolen-Export beschrieben.

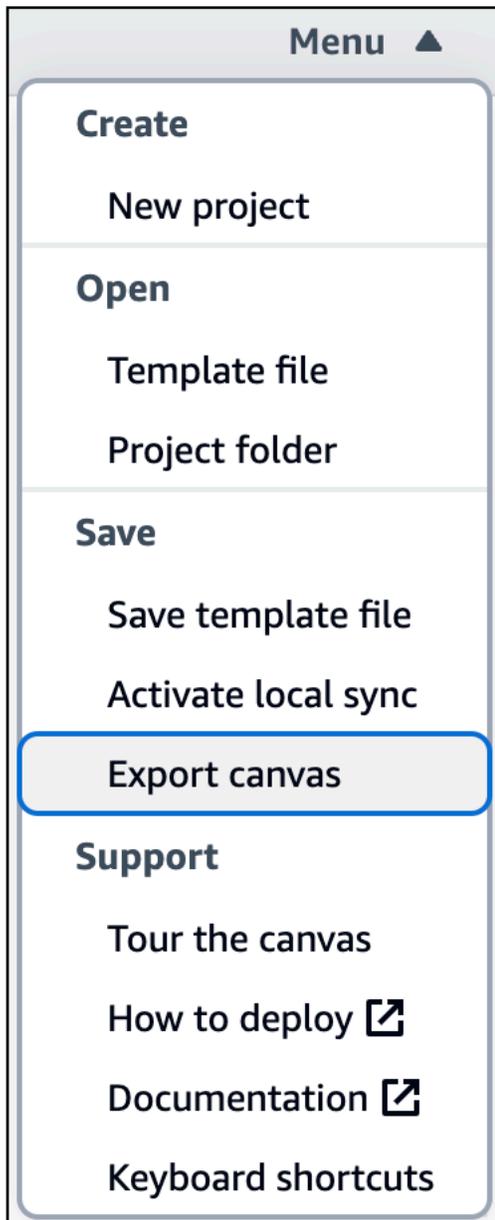
Einen visuellen Überblick über alle Funktionen von Infrastructure Composer finden Sie unter [AWS-Infrastruktur-Composer Visuelle Übersicht über die Konsole](#).

### Über Export Canvas

Die Export-Canvas-Funktion exportiert die Leinwand Ihrer Anwendung als Bild auf Ihren lokalen Computer.

- Infrastructure Composer entfernt die Benutzeroberflächenelemente des visuellen Designers und exportiert nur das Diagramm Ihrer Anwendung.
- Das Standardformat für Bilddateien ist png.
- Die Datei wird an den Standard-Download-Speicherort Ihres lokalen Computers exportiert.

Sie können über das Menü auf die Export-Canvas-Funktion zugreifen.



Leinwand wird exportiert

Wenn Sie Ihren Canvas exportieren, zeigt Infrastructure Composer eine Statusmeldung an.

Wenn der Export erfolgreich ist, wird die folgende Meldung angezeigt:



**Canvas export successful**  
Check your downloads folder.

Wenn der Export nicht erfolgreich war, wird eine Fehlermeldung angezeigt. Wenn Sie eine Fehlermeldung erhalten, versuchen Sie erneut zu exportieren.



**Canvas export error**  
Unexpected error occurred

## Verwenden von Infrastructure Composer im CloudFormation Konsolenmodus

Infrastructure Composer im CloudFormation Konsolenmodus ist das empfohlene Tool zur Visualisierung Ihrer AWS CloudFormation Vorlagen. Sie können dieses Tool auch verwenden, um AWS CloudFormation Vorlagen zu erstellen und zu bearbeiten.

### Wie unterscheidet sich dieser Modus von der Infrastructure Composer-Konsole?

Infrastructure Composer im CloudFormation Konsolenmodus bietet im Allgemeinen dieselbe Funktionalität wie die [Infrastructure Composer-Standardkonsole](#), es gibt jedoch einige Unterschiede, die es zu beachten gilt.

- Dieser Modus ist in den Stack-Workflow in der AWS CloudFormation Konsole integriert. Auf diese Weise können Sie Infrastructure Composer direkt in verwenden AWS CloudFormation.
- [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#), eine Funktion, die Daten automatisch synchronisiert und auf Ihrem lokalen Computer speichert, wird nicht unterstützt.

- Lambda-bezogene Karten (Lambda Function und Lambda Layer) erfordern Code-Builds und Paketierungslösungen, die in diesem Modus nicht verfügbar sind.

#### Note

Diese Karten und die lokale Synchronisierung können in der [Infrastructure](#) Composer-Konsole oder im verwendet werden. AWS Toolkit for Visual Studio Code

Wenn Sie Infrastructure Composer von der AWS CloudFormation Konsole aus öffnen, wird Infrastructure Composer im CloudFormation Konsolenmodus geöffnet. In diesem Modus können Sie Infrastructure Composer verwenden, um Ihre Vorlagen zu visualisieren, zu erstellen und zu aktualisieren.

## Wie greife ich im CloudFormation Konsolenmodus auf Infrastructure Composer zu

Infrastructure Composer im CloudFormation Konsolenmodus ist ein Upgrade von AWS CloudFormation Designer. Wir empfehlen die Verwendung von Infrastructure Composer zur Visualisierung Ihrer AWS CloudFormation Vorlagen. Sie können dieses Tool auch verwenden, um AWS CloudFormation Vorlagen zu erstellen und zu bearbeiten.

1. Gehen Sie zur [Cloudformation-Konsole](#) und melden Sie sich an.
2. Wählen Sie im linken Navigationsmenü Infrastructure Composer aus. Dadurch gelangen Sie im CloudFormation Konsolenmodus zu Infrastructure Composer.

#### Note

Informationen zur Verwendung von Infrastructure Composer im CloudFormation Konsolenmodus finden Sie unter [Verwenden von Infrastructure Composer im CloudFormation Konsolenmodus](#).

## Visualisieren Sie eine Bereitstellung in Infrastructure Composer im CloudFormation Konsolenmodus

Folgen Sie den Anweisungen in diesem Thema, um eine bereitgestellte AWS CloudFormation Stack-/Infrastructure Composer-Vorlage zu visualisieren.

1. Gehen Sie zur [AWS CloudFormation Konsole](#) und melden Sie sich an.
2. Wählen Sie den Stack aus, den Sie bearbeiten möchten.
3. Wählen Sie den Tab Vorlage aus.
4. Wählen Sie Infrastructure Composer aus.

Infrastructure Composer visualisiert Ihren Stack/Ihre Vorlage. Änderungen können auch hier vorgenommen werden.

## Erstellen Sie eine neue Vorlage in Infrastructure Composer im CloudFormation Konsolenmodus

Folgen Sie den Anweisungen in diesem Thema, um eine neue Vorlage zu erstellen.

1. Gehen Sie zur [AWS CloudFormation Konsole](#) und melden Sie sich an.
2. Wählen Sie im linken Navigationsmenü Infrastructure Composer aus. Dadurch wird Infrastructure Composer im CloudFormation Konsolenmodus geöffnet.
3. Ziehen Sie die benötigten Ressourcen ([Karten](#)) aus der Palette Ressourcen per Drag-and-Drop, konfigurieren und verbinden Sie sie.

### Note

Einzelheiten [Wie komponiert man](#) zur Verwendung von Infrastructure Composer finden Sie unter. Beachten Sie, dass Lambda-bezogene Karten (Lambda Function und Lambda Layer) Code-Builds und Paketierungslösungen erfordern, die in Infrastructure Composer im Konsolenmodus nicht verfügbar sind. CloudFormation Diese Karten können in der [Infrastructure](#) Composer-Konsole oder im verwendet werden. AWS Toolkit for Visual Studio Code Informationen zur Verwendung dieser Tools finden Sie unter [Wo Sie Infrastructure Composer verwenden können](#).

4. Doppelklicken Sie auf Karten, um im Bereich mit den Ressourceneigenschaften festzulegen, wie Karten konfiguriert werden.
5. [Connect Ihre Karten](#), um den ereignisgesteuerten Workflow Ihrer Anwendung zu spezifizieren.
6. Wählen Sie Vorlage aus, um Ihren Infrastrukturcode anzuzeigen und zu bearbeiten. Änderungen werden automatisch mit Ihrer Canvas-Ansicht synchronisiert.
7. Sobald Ihre Vorlage bereit ist, in einen Stapel exportiert zu werden, wählen Sie Vorlage erstellen aus.
8. Wählen Sie die CloudFormation Schaltfläche Bestätigen und exportieren nach. Dadurch gelangen Sie zurück zum Workflow „Stack erstellen“ mit einer Meldung, die bestätigt, dass Ihre Vorlage erfolgreich importiert wurde.

 Note

Nur Vorlagen, die Ressourcen enthalten, können exportiert werden.

9. Wählen Sie im Workflow „Stack erstellen“ die Option Weiter aus.
10. Geben Sie einen Stack-Namen ein, überprüfen Sie alle aufgelisteten Parameter und wählen Sie Weiter aus.

 Note

Der Stackname muss mit einem Buchstaben beginnen und darf nur Buchstaben, Zahlen und Bindestriche enthalten.

11. Wählen Sie Weiter aus, nachdem Sie die folgenden Informationen eingegeben haben:
  - Dem Stapel zugeordnete Tags
  - Berechtigungen stapeln
  - Die Fehleroptionen des Stacks

 Note

Anleitungen zur Verwaltung von Stacks finden Sie in den [AWS CloudFormation Best Practices](#) im AWS CloudFormation Benutzerhandbuch.

12. Vergewissern Sie sich, dass Ihre Stack-Details korrekt sind, überprüfen Sie die Bestätigungen unten auf der Seite und klicken Sie auf die Schaltfläche Senden.

AWS CloudFormation beginnt mit der Erstellung des Stacks auf der Grundlage der Daten in Ihrer Vorlage.

## Aktualisieren Sie einen vorhandenen Stack in Infrastructure Composer im CloudFormation Konsolenmodus

Folgen Sie den Anweisungen in diesem Thema, um einen vorhandenen AWS CloudFormation Stack zu aktualisieren.

### Note

Wenn Ihre Datei lokal gespeichert ist, empfehlen wir die Verwendung von [AWS Toolkit for Visual Studio Code](#).

1. Gehen Sie zur [AWS CloudFormation Konsole](#) und melden Sie sich an.
2. Wählen Sie den Stack aus, den Sie bearbeiten möchten.
3. Wählen Sie die Schaltfläche „Aktualisieren“. Dadurch gelangen Sie zum Update-Stack-Assistenten.
4. Wählen Sie auf der rechten Seite In Infrastructure Composer bearbeiten aus.
5. Wählen Sie unten die Schaltfläche mit der Bezeichnung In Infrastructure Composer bearbeiten aus. Dadurch gelangen Sie im CloudFormation Konsolenmodus zu Infrastructure Composer.
6. Hier können Sie Ressourcen ([Karten](#)) aus der Ressourcenpalette ziehen, ablegen, konfigurieren und verbinden.

### Note

Einzelheiten [Wie komponiert man](#) zur Verwendung von Infrastructure Composer finden Sie unter. Beachten Sie, dass Lambda-bezogene Karten (Lambda Function und Lambda Layer) Code-Builds und Paketierungslösungen erfordern, die in Infrastructure Composer im Konsolenmodus nicht verfügbar sind. CloudFormation Diese Karten können in der [Infrastructure](#) Composer-Konsole oder im verwendet werden. AWS Toolkit for Visual

Studio Code Informationen zur Verwendung dieser Tools finden Sie unter [Wo Sie Infrastructure Composer verwenden können](#).

7. Wenn Sie bereit sind, Änderungen zu exportieren AWS CloudFormation, wählen Sie Vorlage aktualisieren aus.
8. Wählen Sie Bestätigen und fahren Sie fort mit CloudFormation. Dadurch gelangen Sie zurück zum Workflow „Stack aktualisieren“ mit einer Meldung, die bestätigt, dass Ihre Vorlage erfolgreich importiert wurde.

 Note

Nur Vorlagen, die Ressourcen enthalten, können exportiert werden.

9. Wählen Sie im Workflow Stack aktualisieren die Option Weiter aus.
10. Überprüfen Sie alle aufgelisteten Parameter und wählen Sie Weiter aus.
11. Wählen Sie Weiter aus, nachdem Sie die folgenden Informationen eingegeben haben:
  - Dem Stapel zugeordnete Tags
  - Berechtigungen stapeln
  - Die Fehleroptionen des Stacks

 Note

Anleitungen zur Verwaltung von Stacks finden Sie in den [AWS CloudFormation Best Practices](#) im AWS CloudFormation Benutzerhandbuch.

12. Vergewissern Sie sich, dass Ihre Stack-Details korrekt sind, überprüfen Sie die Bestätigungen unten auf der Seite und klicken Sie auf die Schaltfläche Senden.

AWS CloudFormation beginnt mit der Aktualisierung des Stacks auf der Grundlage der Aktualisierungen, die Sie in Ihrer Vorlage vorgenommen haben.

# Verwenden von Infrastructure Composer aus dem AWS Toolkit for Visual Studio Code

In diesem Abschnitt wird beschrieben, wie Sie AWS-Infrastruktur-Composer von der aus verwenden können [AWS Toolkit for Visual Studio Code](#). Dies beinhaltet einen visuellen Überblick über Infrastructure Composer aus dem AWS Toolkit for Visual Studio Code. Es enthält auch Anweisungen, die zeigen, wie Sie auf diese Erfahrung zugreifen und Ihr Projekt von VS Code mit der AWS Cloud synchronisieren können. Zum Synchronisieren verwenden Sie den `sam sync` Befehl von AWS SAM CLI. Dieser Abschnitt enthält auch Anleitungen zur Verwendung Amazon Q während Sie sich in Infrastructure Composer befinden, von der AWS Toolkit for Visual Studio Code.

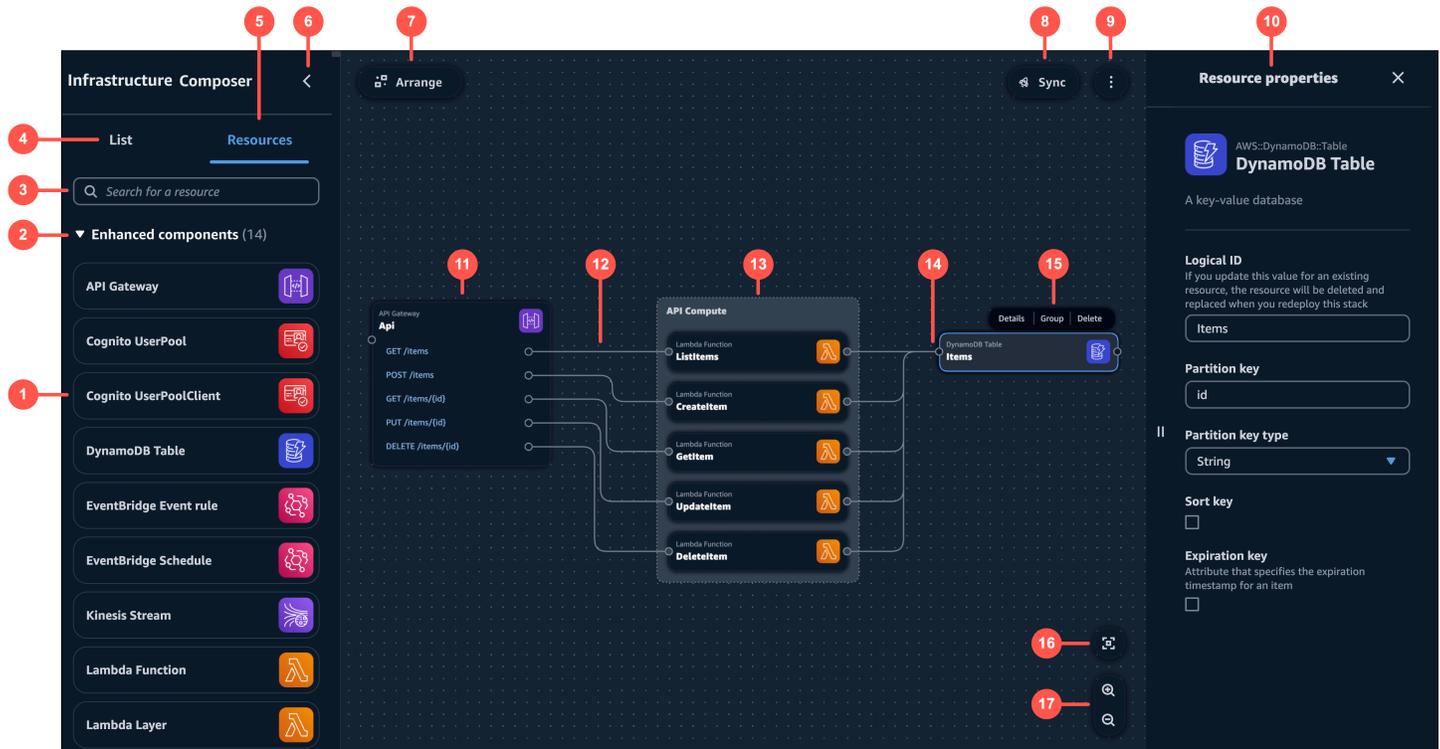
Weitere Anleitungen zur Verwendung von Infrastructure Composer finden Sie unter [Wie komponiert man](#). AWS Toolkit for Visual Studio Code Der Inhalt dieses Abschnitts bezieht sich sowohl auf dieses Erlebnis als auch auf das Erlebnis der Infrastructure Composer-Konsole.

## Themen

- [Visueller Überblick über Infrastructure Composer aus dem AWS Toolkit for Visual Studio Code](#)
- [Greifen Sie auf Infrastructure Composer zu über AWS Toolkit for Visual Studio Code](#)
- [Synchronisieren Sie Infrastructure Composer zur Bereitstellung auf dem AWS Cloud](#)
- [Verwenden AWS-Infrastruktur-Composer mit Amazon Q Developer](#)

## Visueller Überblick über Infrastructure Composer aus dem AWS Toolkit for Visual Studio Code

Der visuelle Designer von Infrastructure Composer AWS Toolkit for Visual Studio Code umfasst eine visuelle Zeichenfläche, die Komponenten enthält, die in der folgenden Abbildung nummeriert und unten aufgeführt sind.



1. Ressourcenpalette — Zeigt Karten an, mit denen Sie entwerfen können.
2. Kartenkategorien — Karten sind nach Kategorien geordnet, die es nur in Infrastructure Composer gibt.
3. Suchleiste für Ressourcen — Suchen Sie nach Karten, die Sie der Arbeitsfläche hinzufügen können.
4. Liste — Zeigt eine Strukturansicht Ihrer Anwendungsressourcen an.
5. Ressourcen — Zeigt die Ressourcenpalette an.
6. Umschalten im linken Bereich — Den linken Bereich ein- oder ausblenden.
7. Anordnen — Ordnet Ihre Anwendungsarchitektur auf der Arbeitsfläche an.
8. Synchronisieren — Initiiert das AWS Serverless Application Model (AWS SAM) CLI `aws sam sync` Befehl zur Bereitstellung Ihrer Anwendung.
9. Menü — Bietet allgemeine Optionen wie die folgenden:
  - Leinwand exportieren
  - Besichtigen Sie die Leinwand
  - Links zur Dokumentation
  - Tastenkombinationen

- 10 Bedienfeld „Ressourceneigenschaften“ — Zeigt die relevanten Eigenschaften der Karte an, die auf der Arbeitsfläche ausgewählt wurde. Dieses Fenster ist dynamisch. Die angezeigten Eigenschaften ändern sich, wenn Sie Ihre Karte konfigurieren.
- 11 Karte — Zeigt eine Ansicht Ihrer Karte auf der Leinwand an.
- 12 Linie — Stellt eine Verbindung zwischen Karten dar.
- 13 Gruppe — Eine Gruppe von Karten. Sie können Karten zur visuellen Organisation gruppieren.
- 14 Port — Verbindungspunkte zu anderen Karten.
- 15 Kartenaktionen — Bietet Aktionen, die Sie auf Ihrer Karte ausführen können.
- Details — Öffnet den Bereich mit den Ressourceneigenschaften.
  - Gruppe — Gruppiert die ausgewählten Karten zusammen.
  - Löschen — Löscht die Karte aus Ihrer Leinwand und Vorlage.
- 16 Neu zentrieren — Zentrieren Sie Ihr Anwendungsdiagramm auf der visuellen Leinwand neu.
- 17 Zoom — Zoomen Sie auf Ihrer Leinwand hinein und heraus.

## Greifen Sie auf Infrastructure Composer zu über AWS Toolkit for Visual Studio Code

Folgen Sie den Anweisungen in diesem Thema, um über den auf Infrastructure Composer zuzugreifen AWS Toolkit for Visual Studio Code.

### Note

Bevor Sie über den auf Infrastructure Composer zugreifen können AWS Toolkit for Visual Studio Code, müssen Sie zunächst das Toolkit for VS Code herunterladen und installieren. Anweisungen finden Sie unter [Herunterladen des Toolkit for VS Code](#).

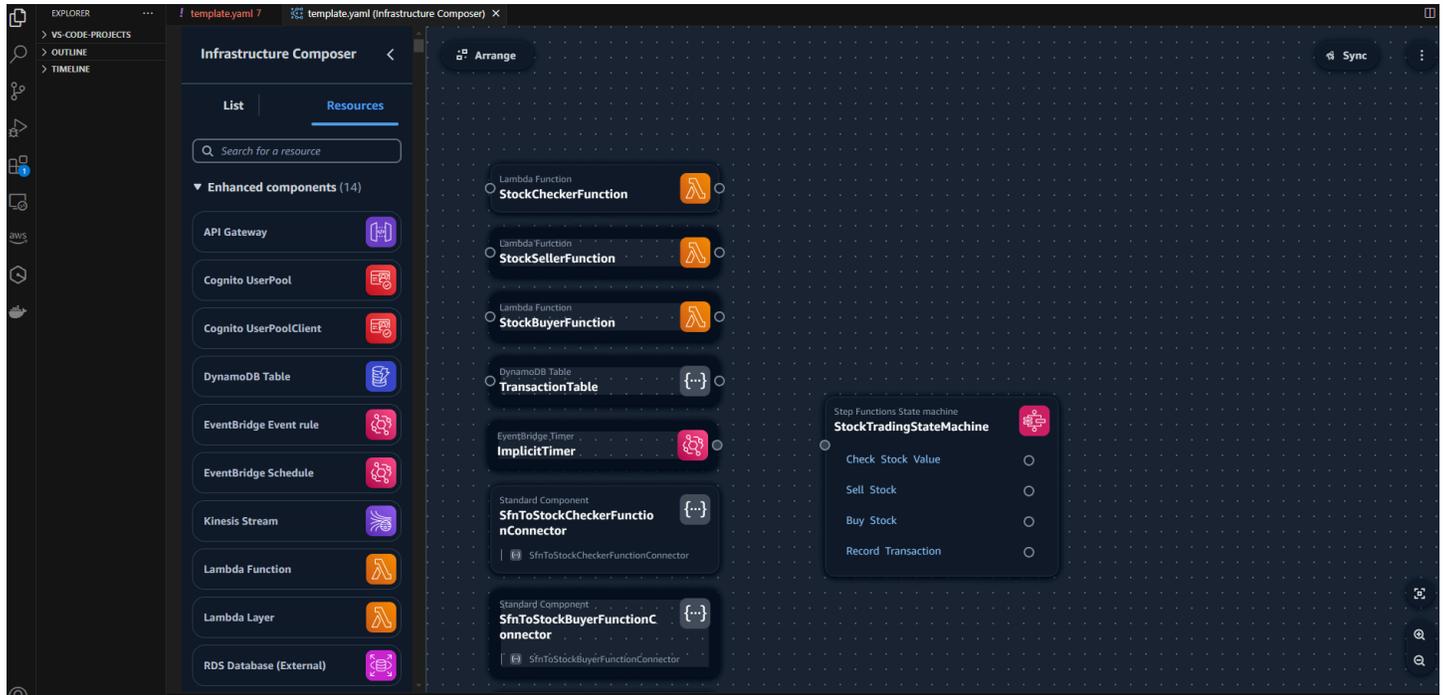
So greifen Sie über das Toolkit for VS Code auf Infrastructure Composer zu

Sie können auf eine der folgenden Arten auf Infrastructure Composer zugreifen:

1. Indem Sie die Infrastructure Composer-Schaltfläche aus einer beliebigen AWS CloudFormation AWS SAM Vorlage auswählen.
2. Über das Kontextmenü, indem Sie mit der rechten Maustaste auf Ihre AWS SAM Vorlage AWS CloudFormation oder auf Ihre Vorlage klicken.

### 3. Aus der VS Code-Befehlspalette.

Im Folgenden finden Sie ein Beispiel für den Zugriff auf Infrastructure Composer über die Infrastructure Composer-Schaltfläche:



Weitere Informationen zum Zugriff auf Infrastructure Composer finden Sie unter [Zugreifen AWS-Infrastruktur-Composer über das Toolkit](#).

## Synchronisieren Sie Infrastructure Composer zur Bereitstellung auf dem AWS Cloud

Verwenden Sie die Synchronisierungsschaltfläche in AWS-Infrastruktur-Composer von AWS Toolkit for Visual Studio Code, um Ihre Anwendung auf dem bereitzustellen AWS Cloud.

Die Synchronisierungsschaltfläche initiiert den `sam sync` Befehl über die AWS SAM Befehlszeilenschnittstelle (CLI).

Mit dem Befehl können Sie neue Anwendungen bereitstellen oder Änderungen, die Sie lokal vornehmen, schnell mit dem AWS Cloud synchronisieren. Das Ausführen `sam sync` kann Folgendes beinhalten:

- Erstellen Sie Ihre Anwendung mitsam `build`, um Ihre lokalen Anwendungsdateien für die Bereitstellung vorzubereiten, indem Sie ein lokales `.aws-sam` Verzeichnis erstellen oder aktualisieren.
- Ressourcen, die den AWS Service unterstützen APIs, finden Sie unter AWS SAM CLI verwendet die APIs , um Ihre Änderungen bereitzustellen. Das AWS SAM CLI tut dies, um Ihre Ressourcen in der Cloud schnell zu aktualisieren.
- Falls erforderlich, die AWS SAM CLI führt eine AWS CloudFormation Bereitstellung durch, um Ihren gesamten Stack mithilfe eines Änderungssatzes zu aktualisieren.

Der `sam sync` Befehl eignet sich am besten für schnelle Entwicklungsumgebungen, in denen eine schnelle Aktualisierung Ihrer Cloud-Ressourcen Ihren Entwicklungs- und Testworkflows zugute kommen kann.

Weitere Informationen dazu finden Sie `sam sync` unter [Using sam Sync](#) im AWS Serverless Application Model Entwicklerhandbuch.

## Einrichten

Um die Synchronisierungsfunktion in Infrastructure Composer verwenden zu können, benötigen Sie AWS SAM CLI auf Ihrem lokalen Computer installiert. Anweisungen finden Sie unter [Installation des AWS SAM CLI](#) im Entwicklerhandbuch für AWS Serverless Application Model .

Wenn Sie die Synchronisierungsfunktion in Infrastructure Composer verwenden, AWS SAM CLI verweist auf Ihre Konfigurationsdatei für die Informationen, die sie zum Synchronisieren Ihrer Anwendung mit der benötigt AWS Cloud. Anweisungen zum Erstellen, Ändern und Verwenden von Konfigurationsdateien finden [Sie im AWS Serverless Application Model Entwicklerhandbuch unter Projekteinstellungen konfigurieren](#).

## Synchronisieren und implementieren Sie Ihre Anwendung

Um Ihre Anwendung mit dem zu synchronisieren AWS Cloud

1. Wählen Sie die Synchronisierungsschaltfläche auf der Infrastructure Composer-Arbeitsfläche aus.
2. Möglicherweise werden Sie aufgefordert, zu bestätigen, dass Sie mit einem Entwicklungs-Stack arbeiten. Wählen Sie OK, um fortzufahren.
3. Infrastructure Composer fordert Sie möglicherweise auf, die folgenden Optionen zu konfigurieren:

- **AWS-Region**— Die Region, mit der Ihre Anwendung synchronisiert werden soll.
- **AWS CloudFormation Stack-Name** — Der Name Ihres AWS CloudFormation Stacks. Sie können einen vorhandenen Stacknamen auswählen oder einen neuen erstellen.
- **Amazon Simple Storage Service (Amazon S3) -Bucket** — Der Name Ihres Amazon S3 S3-Buckets. Der AWS SAM CLI wird Ihre Anwendungsdateien und Ihren Funktionscode hier verpacken und speichern. Sie können einen vorhandenen Bucket auswählen oder einen neuen erstellen.

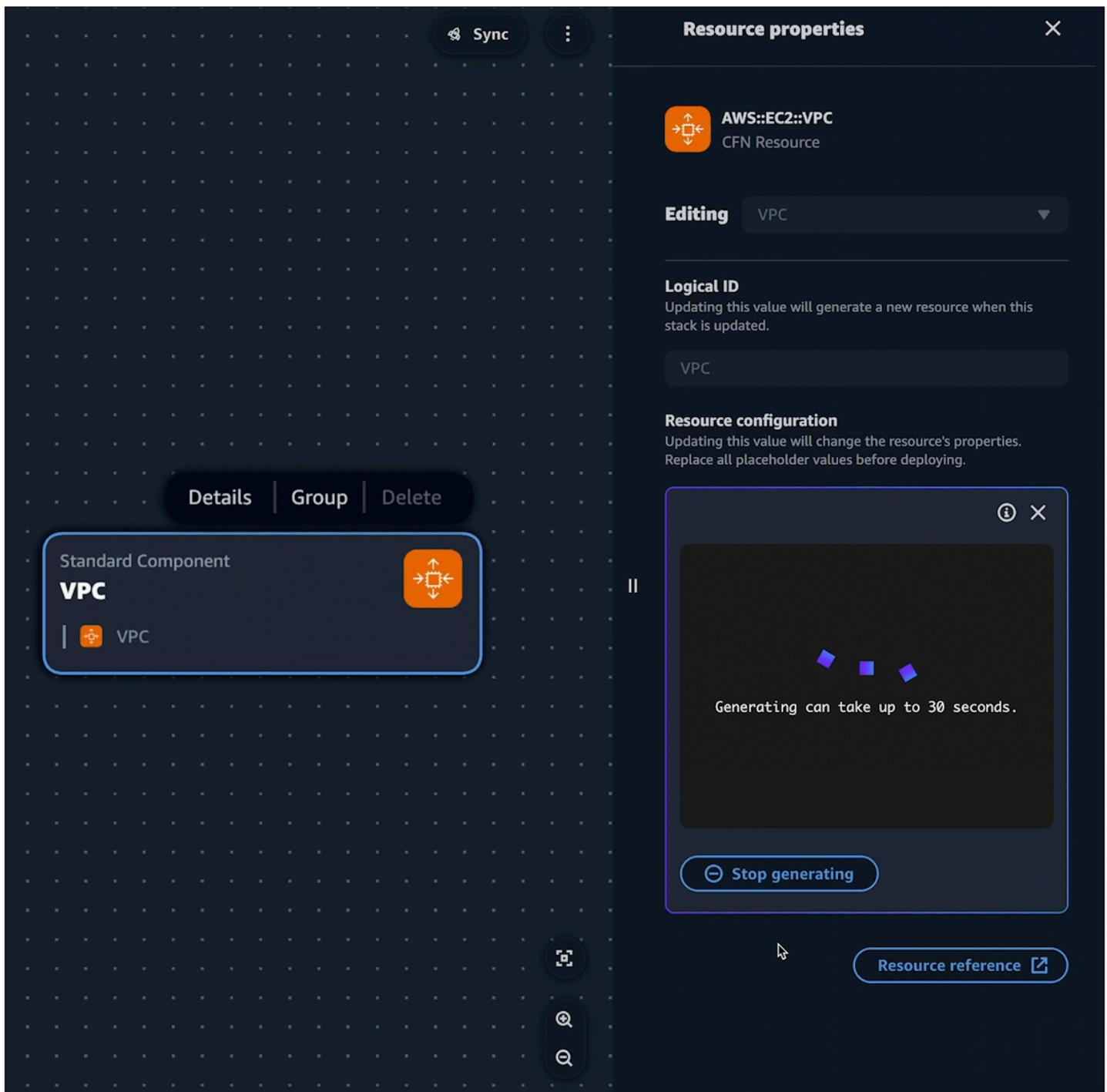
Infrastructure Composer initiiert das AWS SAM CLI `sam sync` Befehl und öffnet ein Terminalfenster in Ihrer IDE, um den Fortschritt anzuzeigen.

## Verwenden AWS-Infrastruktur-Composer mit Amazon Q Developer

AWS-Infrastruktur-Composer von der AWS Toolkit for Visual Studio Code bietet eine Integration mit Amazon Q. Du kannst benutzen Amazon Q innerhalb von Infrastructure Composer, um den Infrastrukturcode für Ihre AWS Ressourcen zu generieren, während Sie Ihre Anwendung entwerfen.

Amazon Q ist ein auf maschinelles Lernen basierender Allzweck-Codegenerator. Weitere Informationen finden Sie unter Was ist [Amazon Q?](#) in der Amazon Q Developer Benutzerleitfaden.

Für Standardressourcen - und Standardkomponentenkarten können Sie Folgendes verwenden Amazon Q um Vorschläge für den Infrastrukturcode für Ihre Ressourcen zu generieren.



Standardressourcen - und Standardkomponentenkarten können eine AWS CloudFormation Ressource oder eine Sammlung von AWS CloudFormation Ressourcen darstellen. Weitere Informationen hierzu finden Sie unter [Karten in Infrastructure Composer konfigurieren und ändern](#).

## Einrichtung

Zur Verwendung Amazon Q In Infrastructure Composer müssen Sie sich authentifizieren mit Amazon Q im Toolkit. Anweisungen finden Sie unter [Erste Schritte mit Amazon Q in VS Code und JetBrains](#) in Amazon Q Developer Benutzerleitfaden.

### Die Verwendung von Amazon Q Developer im Infrastructure Composer

Sie können Folgendes verwenden ... Amazon Q Developer aus dem Bereich mit den Ressourceneigenschaften einer beliebigen Standardressource oder Standardkomponentenkarte.

Zur Verwendung Amazon Q im Infrastructure Composer

1. Öffnen Sie auf einer Standardressource - oder Standardkomponentenkarte den Bereich mit den Ressourceneigenschaften.
2. Suchen Sie das Feld Ressourcenkonfiguration. Dieses Feld enthält den Infrastrukturcode für die Karte.
3. Wählen Sie die Schaltfläche Vorschläge generieren. Amazon Q generiert einen Vorschlag.

#### Note

Der in dieser Phase generierte Code überschreibt den vorhandenen Infrastrukturcode aus Ihrer Vorlage nicht.

4. Um weitere Vorschläge zu generieren, wählen Sie Regenerieren aus. Sie können zwischen den Stichproben wechseln, um die Ergebnisse zu vergleichen.
5. Um eine Option auszuwählen, wählen Sie „Auswählen“. Sie können den Code hier ändern, bevor Sie ihn in Ihrer Anwendung speichern. Um den Vorgang ohne Speichern zu beenden, wählen Sie das Exit-Symbol (X).
6. Um den Code in Ihrer Anwendungsvorlage zu speichern, wählen Sie im Bereich mit den Ressourceneigenschaften die Option Speichern aus.

## Weitere Informationen

Für weitere Informationen über Amazon Q, siehe [Was ist Amazon Q?](#) in der Amazon Q Developer Benutzerleitfaden.

# Wie komponiert man in AWS-Infrastruktur-Composer

In diesem Abschnitt werden die Grundlagen der Verwendung von Infrastructure Composer aus den [CloudFormation Konsolenmodus](#) Bereichen [Infrastructure Composer-Konsole](#), und beschrieben [AWS Toolkit for Visual Studio Code](#). Insbesondere enthalten die Themen in diesem Abschnitt wichtige Informationen zur Erstellung einer Anwendung mit Infrastructure Composer sowie Einzelheiten zu zusätzlichen Funktionen und Tastenkombinationen. Es gibt einige Unterschiede in der Funktionalität zwischen Konsole und VS Code. In den Themen in diesem Abschnitt werden diese Variationen, wo sie vorkommen, identifiziert und beschrieben.

Nachdem Sie Ihre Anwendung erstellt haben, können Sie nach Informationen [Stellen Sie Ihre serverlose Infrastructure Composer-Anwendung in der AWS Cloud bereit](#) zur Bereitstellung Ihrer Anwendung suchen.

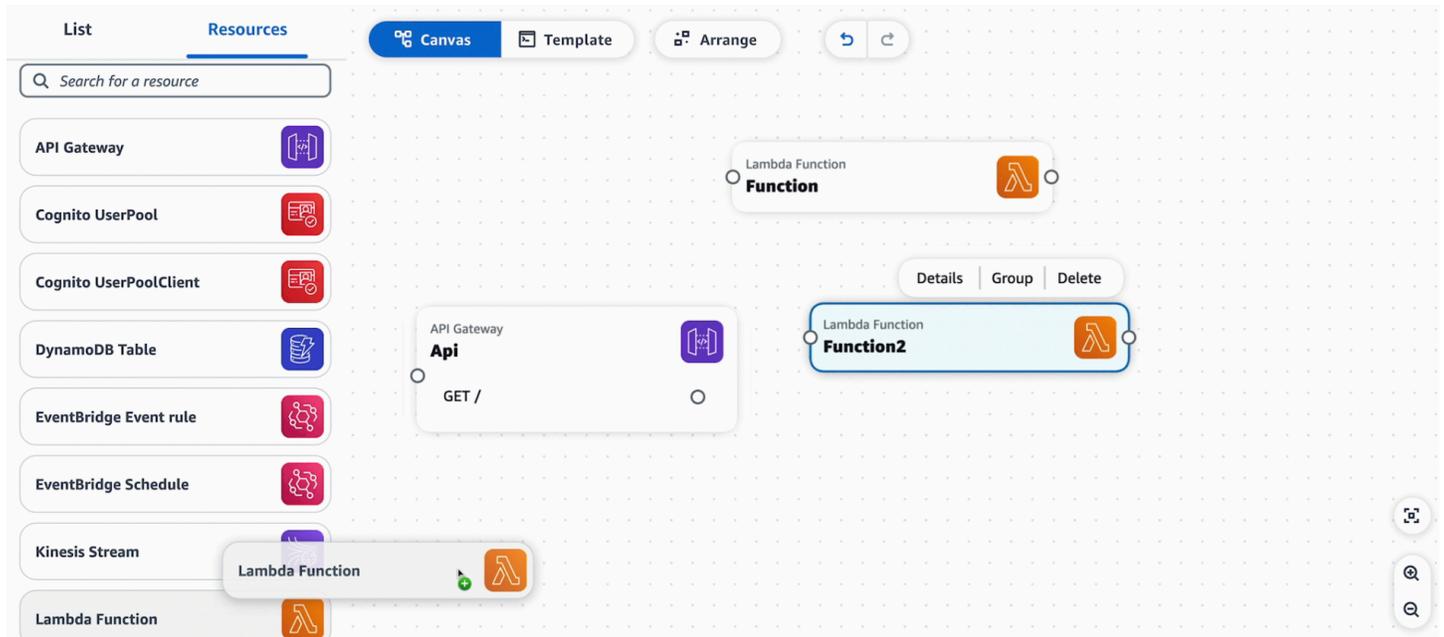
## Themen

- [Platzieren Sie Karten auf der visuellen Leinwand von Infrastructure Composer](#)
- [Gruppieren Sie Karten auf der visuellen Leinwand von Infrastructure Composer](#)
- [Karten auf der visuellen Leinwand von Infrastructure Composer Connect](#)
- [Trennen Sie Karten in Infrastructure Composer](#)
- [Ordnen Sie Karten auf der visuellen Leinwand von Infrastructure Composer an](#)
- [Karten in Infrastructure Composer konfigurieren und ändern](#)
- [Karten in Infrastructure Composer löschen](#)
- [Codeaktualisierungen mit dem Change Inspector in Infrastructure Composer anzeigen](#)
- [Verweisen auf externe Dateien in Infrastructure Composer](#)
- [Integrieren Sie Infrastructure Composer in Amazon Virtual Private Cloud \(Amazon VPC\)](#)

## Platzieren Sie Karten auf der visuellen Leinwand von Infrastructure Composer

In diesem Abschnitt wird beschrieben, wie Sie Infrastructure [Composer-Karten](#) auf der visuellen Leinwand auswählen und ziehen. Bevor Sie beginnen, sollten Sie herausfinden, welche Ressourcen Ihre Anwendung benötigt und wie sie miteinander interagieren müssen. Tipps dazu finden Sie unter [Erstellen Sie Ihre erste Anwendung mit Infrastructure Composer](#).

Um Ihrer Anwendung eine Karte hinzuzufügen, ziehen Sie sie aus der Ressourcenpalette und legen Sie sie auf die visuelle Leinwand ab.



Sie können zwischen zwei Kartentypen wählen: [Erweiterte Komponentenkarten](#) und [Standard-IaC-Ressourcenkarten](#).

Nachdem Sie Ihre Karten auf der visuellen Leinwand platziert haben, können Sie Ihre Karten gruppieren, verbinden, anordnen und konfigurieren. Informationen dazu finden Sie in den folgenden Themen:

- [Gruppieren Sie Karten auf der visuellen Leinwand von Infrastructure Composer](#)
- [Karten auf der visuellen Leinwand von Infrastructure Composer Connect](#)
- [Ordnen Sie Karten auf der visuellen Leinwand von Infrastructure Composer an](#)
- [Karten in Infrastructure Composer konfigurieren und ändern](#)

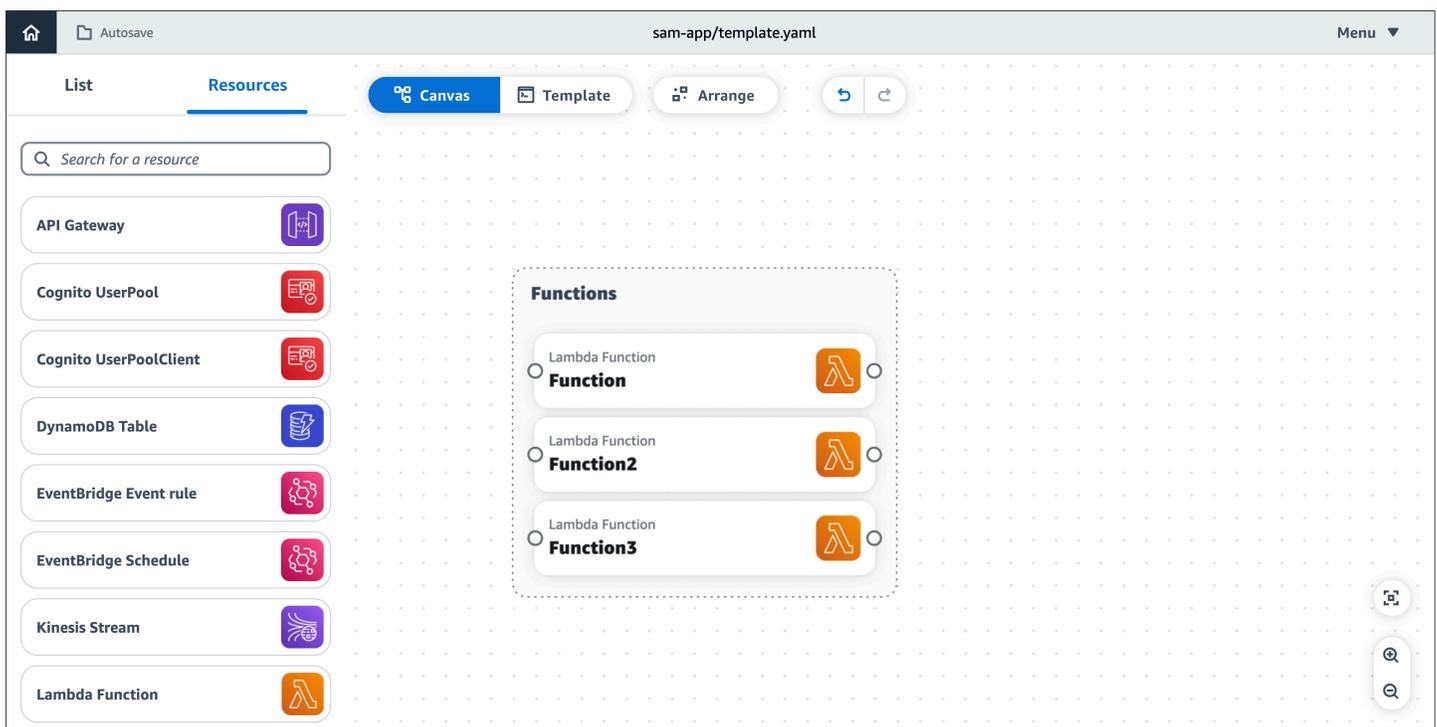
## Gruppieren Sie Karten auf der visuellen Leinwand von Infrastructure Composer

Dieses Thema enthält Einzelheiten zur Gruppierung erweiterter Komponentenkarten und Standardkomponentenkarten. Mithilfe von Gruppierungskarten können Sie Ihre Ressourcen kategorisieren und organisieren, ohne sich Gedanken darüber machen zu müssen, welchen Code oder welches Markup Sie schreiben müssen.

## Gruppieren erweiterter Komponentenkarten

Es gibt zwei Möglichkeiten, Karten mit erweiterten Komponenten zu gruppieren:

- Halten Sie die Umschalttaste gedrückt und wählen Sie die zu gruppierenden Karten aus. Wählen Sie dann im Menü mit den Ressourcenaktionen die Option Gruppieren aus.
- Wählen Sie eine Karte aus, die Sie in einer Gruppe haben möchten. Wählen Sie im angezeigten Menü die Option Gruppe aus. Dadurch wird eine Gruppe erstellt, in die Sie andere Karten ziehen und dort ablegen können.



## Gruppieren einer Standardkomponentenkarte in eine andere

Das folgende Beispiel zeigt, wie eine Standardkomponentenkarte über das Bedienfeld „Ressourceneigenschaften“ in eine andere Karte gruppiert werden kann:

The screenshot displays the AWS Infrastructure Composer interface. On the left, a component card for 'Function' is shown, containing a 'Role' component. The card has buttons for 'Details', 'Group', and 'Delete'. On the right, the 'Resource properties' panel is open for an 'AWS::Lambda::Function' resource. The 'Editing' section shows a dropdown menu with 'Function' selected. The 'Logical ID' section shows 'Function' with a checkmark and a note: 'Updating this value will generate a new resource when this stack is updated.' Below this, a text input field contains 'Function'. The 'Resource configuration' section has a note: 'Updating this value will change the resource's properties. Replace all placeholder values before deploying.' A code editor shows the following configuration:

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

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

Im Feld Ressourcenkonfiguration im Bereich Ressourceneigenschaften `Role` wurde in der Lambda-Funktion auf die verwiesen. Dies führt dazu, dass die Karte „Rolle“ in der Karte „Funktion“ auf der Arbeitsfläche gruppiert wird.

# Karten auf der visuellen Leinwand von Infrastructure Composer Connect

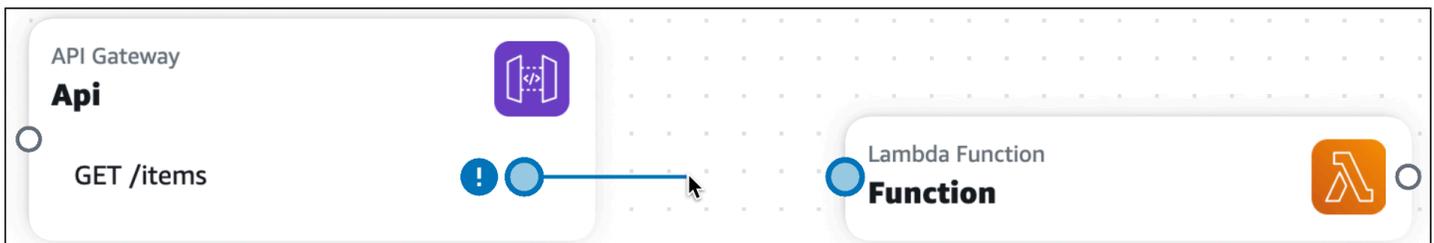
In diesem Thema erfahren Sie, wie Sie Karten in Infrastructure Composer verbinden. Dieser Abschnitt enthält Einzelheiten zum Anschließen erweiterter Komponentenkarten und Standardkomponentenkarten. Außerdem finden Sie einige Beispiele, die die verschiedenen Verbindungsmöglichkeiten von Karten veranschaulichen.

## Anschließen erweiterter Komponentenkarten

Auf Karten mit erweiterten Komponenten wird anhand der Anschlüsse visuell angezeigt, wo Verbindungen hergestellt werden können.

- Ein Anschluss auf der rechten Seite einer Karte weist darauf hin, dass die Karte die Möglichkeit hat, eine andere Karte aufzurufen.
- Ein Anschluss auf der linken Seite einer Karte weist darauf hin, dass die Karte von einer anderen Karte aufgerufen werden kann.

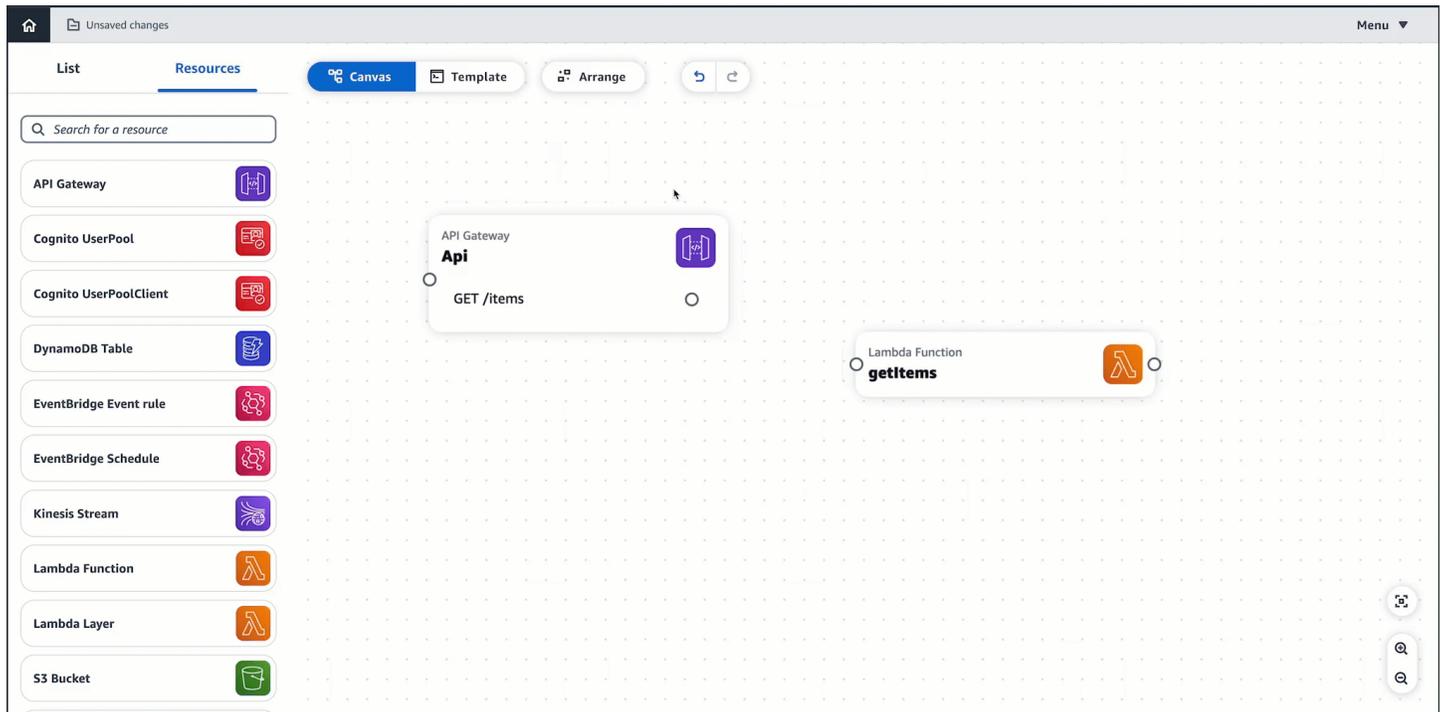
Connect Karten miteinander, indem du auf den rechten Anschluss einer Karte klickst und ihn auf einen linken Anschluss einer anderen Karte ziehst.



Wenn Sie eine Verbindung herstellen, wird eine Meldung angezeigt, die Sie darüber informiert, ob die Verbindung erfolgreich hergestellt wurde. Wählen Sie die Meldung aus, um zu sehen, was Infrastructure Composer geändert hat, um eine Verbindung bereitzustellen. Wenn die Verbindung nicht erfolgreich war, können Sie die Vorlagenansicht auswählen, um Ihren Infrastrukturcode manuell zu aktualisieren, um die Verbindung bereitzustellen.

- Wenn der Vorgang erfolgreich war, klicken Sie auf die Nachricht, um den Change Inspector aufzurufen. Hier können Sie sehen, welche Änderungen Infrastructure Composer zur Bereitstellung Ihrer Verbindung vorgenommen hat.

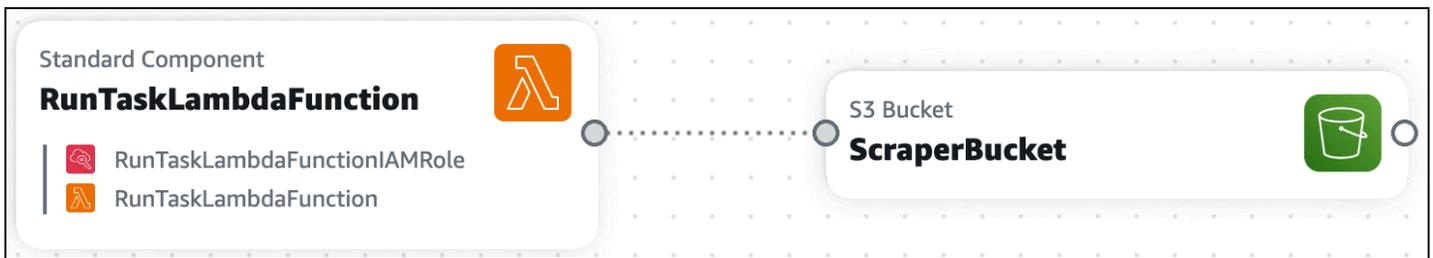
- Wenn dies nicht erfolgreich ist, wird eine Meldung angezeigt. Sie können die Vorlagenansicht auswählen und Ihren Infrastrukturcode manuell aktualisieren, um die Verbindung bereitzustellen.



Wenn Sie erweiterte Komponentenkarten miteinander verbinden, erstellt Infrastructure Composer automatisch den Infrastrukturcode in Ihrer Vorlage, um die ereignisgesteuerte Beziehung zwischen Ihren Ressourcen bereitzustellen.

## Verbinden von Standard-Komponentenkarten (Standard-IaC-Ressourcenkarten)

Standard-IaC-Ressourcenkarten enthalten keine Anschlüsse zum Herstellen von Verbindungen mit anderen Ressourcen. Während der [Kartenkonfiguration](#) geben Sie ereignisgesteuerte Beziehungen in der Vorlage Ihrer Anwendung an. Infrastructure Composer erkennt diese Verbindungen automatisch und visualisiert sie mit einer gepunkteten Linie zwischen Ihren Karten. Im Folgenden finden Sie ein Beispiel für eine Verbindung zwischen einer Standard-Komponentenkarte und einer erweiterten Komponentenkarte:



Das folgende Beispiel zeigt, wie eine Lambda-Funktion mit einer Amazon API Gateway-Rest-API verbunden werden kann:

```

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
```

Im obigen Beispiel `Integration`: spezifiziert der `ApiGatewayMethod`: unter aufgeführte Codeausschnitt die ereignisgesteuerte Beziehung, die die beiden Karten verbindet.

## Beispiele für das Verbinden von Karten in Infrastructure Composer

Verwenden Sie die Beispiele in diesem Abschnitt, um zu verstehen, wie Karten in Infrastructure Composer verbunden werden können.

Rufen Sie eine AWS Lambda Funktion auf, wenn ein Artikel in einen Amazon Simple Storage Service (Amazon S3) -Bucket gelegt wird

In diesem Beispiel ist eine Amazon S3 S3-Bucket-Karte mit einer Lambda-Funktionskarte verbunden. Wenn ein Artikel in den Amazon S3 S3-Bucket gestellt wird, wird die Funktion aufgerufen. Die

Funktion kann dann verwendet werden, um den Artikel zu verarbeiten oder andere Ereignisse in Ihrer Anwendung auszulösen.



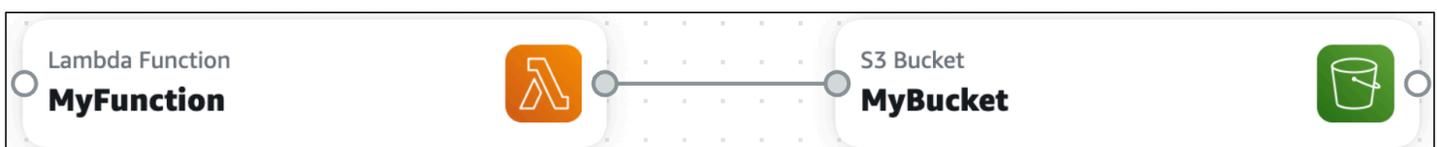
Diese Interaktion erfordert, dass ein Ereignis für die Funktion definiert wird. Infrastructure Composer bietet Folgendes:

```

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
  
```

Rufen Sie einen Amazon S3 S3-Bucket von einer Lambda-Funktion aus auf

In diesem Beispiel ruft eine Lambda-Funktionskarte eine Amazon S3 S3-Bucket-Karte auf. Die Lambda-Funktion kann verwendet werden, um CRUD-Operationen für Artikel im Amazon S3 S3-Bucket durchzuführen.



Für diese Interaktion ist Folgendes erforderlich, das von Infrastructure Composer bereitgestellt wird:

- IAM-Richtlinien, die es der Lambda-Funktion ermöglichen, mit dem Amazon S3 S3-Bucket zu interagieren.
- Umgebungsvariablen, die das Verhalten der Lambda-Funktion beeinflussen.

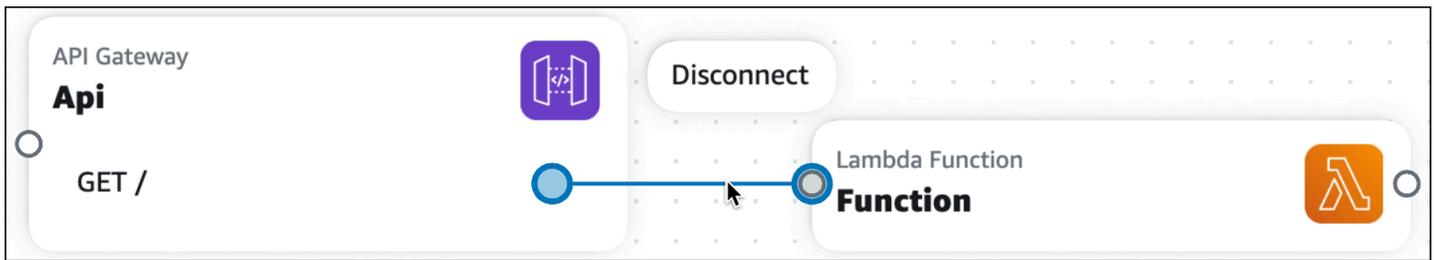
```
Transform: AWS::Serverless-2016-10-31
...
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    ...
  MyBucketBucketPolicy:
    Type: AWS::S3::BucketPolicy
    ...
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      Environment:
        Variables:
          BUCKET_NAME: !Ref MyBucket
          BUCKET_ARN: !GetAtt MyBucket.Arn
      Policies:
        - S3CrudPolicy:
            BucketName: !Ref MyBucket
```

## Trennen Sie Karten in Infrastructure Composer

In Infrastructure Composer verbinden und trennen Sie AWS Ressourcen mithilfe erweiterter Komponentenkarten und Standardkomponentenkarten. In diesem Abschnitt wird beschrieben, wie Sie beide Kartentypen trennen.

### Verbesserte Komponentenkarten

Um die Verbindung zwischen Karten mit erweiterten Komponenten zu trennen, wählen Sie die Leitung aus und klicken Sie auf Trennen.



Infrastructure Composer ändert Ihre Vorlage automatisch, um die ereignisgesteuerte Beziehung aus Ihrer Anwendung zu entfernen.

## Standardkomponentenkarten

Standardkomponentenkarten enthalten keine Anschlüsse zum Herstellen von Verbindungen mit anderen Ressourcen. Während der [Kartenkonfiguration](#) geben Sie ereignisgesteuerte Beziehungen in der Vorlage Ihrer Anwendung an. Infrastructure Composer erkennt diese Verbindungen automatisch und visualisiert sie mit einer gepunkteten Linie zwischen Ihren Karten. Um eine Standardkomponentenkarte zu trennen, entfernen Sie die ereignisgesteuerte Beziehung in der Vorlage Ihrer Anwendung.

Das folgende Beispiel zeigt eine Lambda-Funktion, die mit einer Amazon API Gateway Gateway-REST-API verbunden ist:

```
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

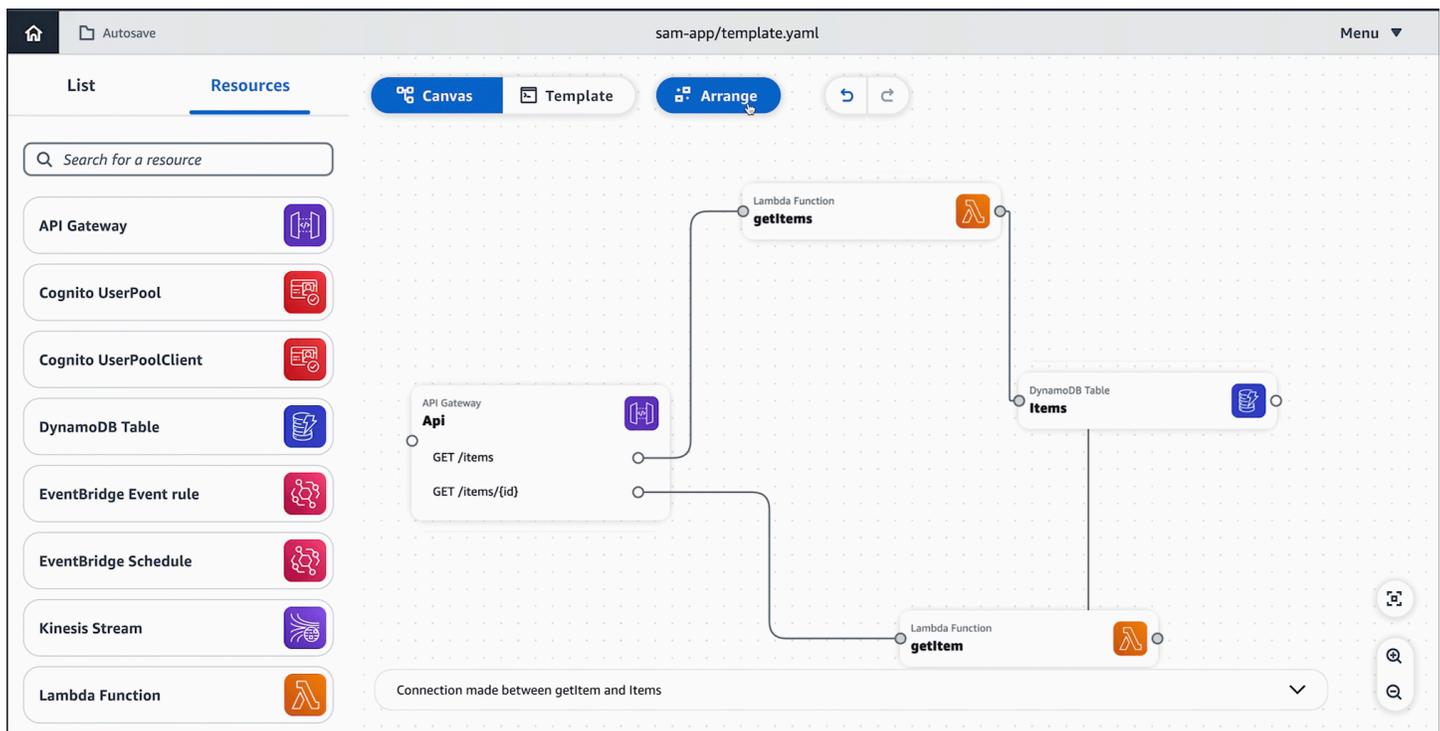
LambdaExecutionRole:
Type: 'AWS::IAM::Role'
Properties:
  AssumeRolePolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal:
          Service: lambda.amazonaws.com
        Action: 'sts:AssumeRole'
  Policies:
    - PolicyName: LambdaExecutionPolicy
      PolicyDocument:
        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
```

Um die Verbindung zwischen den beiden Karten zu entfernen, entfernen Sie die unter MyLambdaFunction aufgeführten Verweise auf. `ApiGatewayMethod: Integration`

## Ordnen Sie Karten auf der visuellen Leinwand von Infrastructure Composer an

Wählen Sie Anordnen, um Karten auf der Leinwand visuell anzuordnen und zu organisieren.

Die Verwendung der Schaltfläche „Anordnen“ ist besonders nützlich, wenn sich viele Karten und Verbindungen auf der Leinwand befinden.



## Karten in Infrastructure Composer konfigurieren und ändern

In Infrastructure Composer stellen Karten Ressourcen dar, die Sie zum Entwerfen Ihrer Anwendungsarchitektur verwenden. Wenn Sie eine Karte in Infrastructure Composer konfigurieren, definieren Sie die Details der Ressourcen in Ihrer Anwendung. Dazu gehören Details wie die logische ID und der Partitionsschlüssel einer Karte. Die Art und Weise, wie diese Informationen definiert werden, variiert zwischen Karten mit erweiterter Komponente und Standardkarten.

Eine erweiterte Komponentenkarte ist eine Sammlung von AWS CloudFormation Ressourcen, die zu einer einzigen kuratierten Karte zusammengefasst wurden, die Benutzerfreundlichkeit und

Funktionalität verbessert und für eine Vielzahl von Anwendungsfällen konzipiert ist. Eine Standard-laC-Ressourcenkarte steht für eine einzelne AWS CloudFormation Ressource. Jede Standard-laC-Ressourcenkarte trägt, sobald sie auf die Arbeitsfläche gezogen wurde, die Bezeichnung Standardkomponente.

Dieses Thema enthält Einzelheiten zur Konfiguration von erweiterten Komponentenkarten und Standard-Komponentenkarten.

#### Note

Dieses Thema bezieht sich auf die Verwendung von Karten aus der Infrastructure Composer-Konsole, der AWS Toolkit for Visual Studio Code Erweiterung und in Infrastructure Composer im CloudFormation Konsolenmodus. Lambda-bezogene Karten (Lambda Function und Lambda Layer) erfordern Code-Builds und Paketierungslösungen, die in Infrastructure Composer im Konsolenmodus nicht verfügbar sind. CloudFormation Weitere Informationen finden Sie unter [Verwenden von Infrastructure Composer im CloudFormation Konsolenmodus](#).

## Themen

- [Verbesserte Komponentenkarten in Infrastructure Composer](#)
- [Standardkarten in Infrastructure Composer](#)

## Verbesserte Komponentenkarten in Infrastructure Composer

Um erweiterte Komponentenkarten zu konfigurieren, stellt Infrastructure Composer im Bereich Ressourceneigenschaften ein Formular zur Verfügung. Dieses Formular wurde speziell zusammengestellt, um Sie durch die Konfiguration der einzelnen erweiterten Komponentenkarten zu führen. Beim Ausfüllen des Formulars ändert Infrastructure Composer Ihren Infrastrukturcode.

Einige erweiterte Komponentenkarten verfügen über zusätzliche Funktionen. In diesem Abschnitt werden die Grundlagen der Verwendung erweiterter Komponentenkarten beschrieben und es werden Einzelheiten zu Karten mit zusätzlichen Funktionen beschrieben.

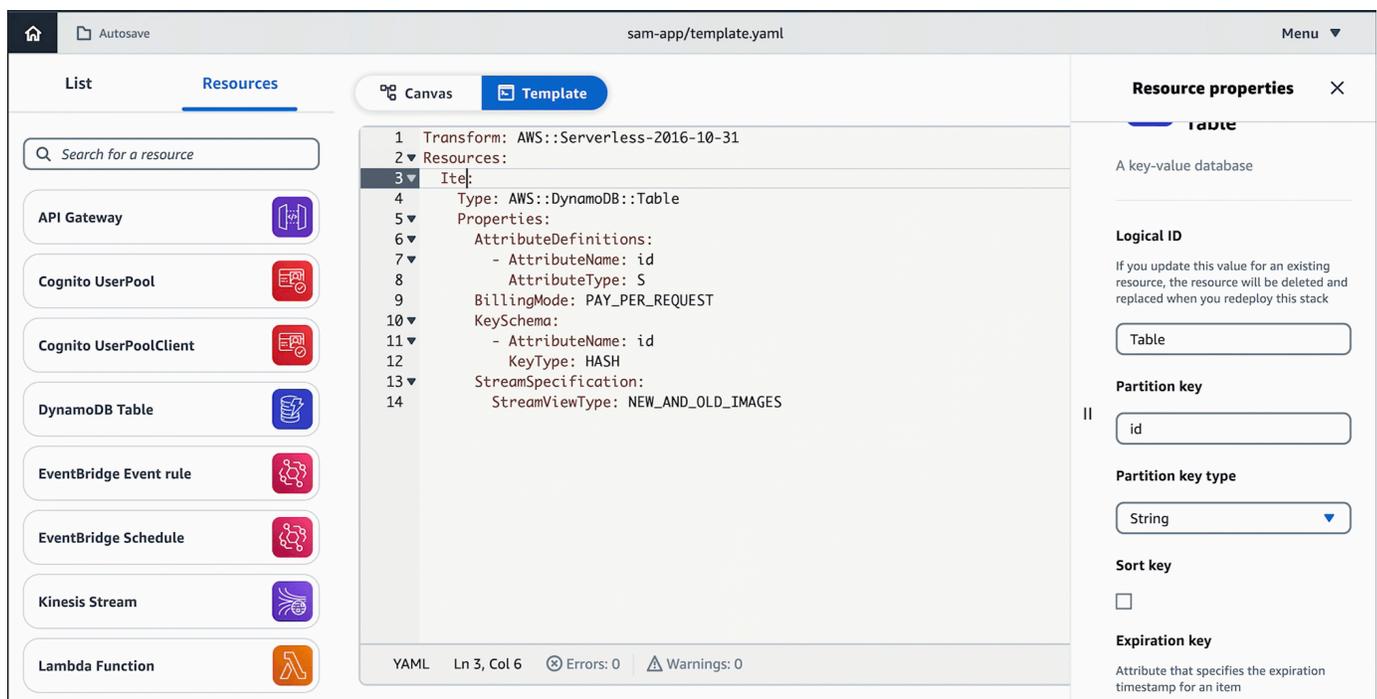
Weitere Informationen zu erweiterten Komponentenkarten finden Sie unter [Verbesserte Komponentenkarten in Infrastructure Composer](#) und [Verbesserte Komponentenkarten in Infrastructure Composer](#)

## Verfahren

Der Bereich mit den Ressourceneigenschaften vereinfacht die Konfiguration und fügt Leitfäden hinzu, die die Kartenkonfiguration vereinfachen. Gehen Sie wie folgt vor, um dieses Fenster zu verwenden:

1. Doppelklicken Sie auf eine Karte, um das Fenster mit den Ressourceneigenschaften aufzurufen.
2. Klicken Sie auf eine Karte und wählen Sie Details aus, um den Bereich mit den Ressourceneigenschaften aufzurufen.
3. Wählen Sie für Infrastructure Composer aus der die Option Vorlage aus AWS Management Console, um Ihren Anwendungscode anzuzeigen. Konfigurieren Sie direkt von hier aus.

Das folgende Bild zeigt, wie das gemacht werden kann:



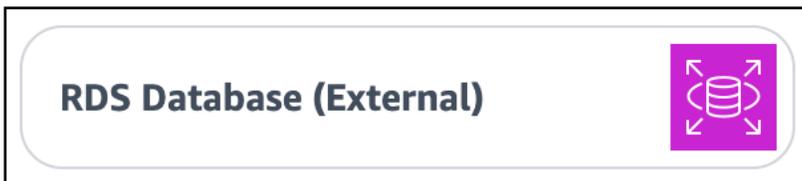
## Verwenden von Infrastructure Composer mit Amazon Relational Database Service (Amazon RDS)

AWS-Infrastruktur-Composer bietet eine Integration mit Amazon Relational Database Service (Amazon RDS). Mithilfe der erweiterten Komponentenkarte RDS-Datenbank (extern) in Infrastructure Composer können Sie Ihre Anwendung mit Amazon RDS verbinden DB Cluster, Instances und Proxys, die auf einer anderen AWS CloudFormation oder AWS Serverless Application Model (AWS SAM) -Vorlage definiert sind.

Die erweiterte Komponentenkarte RDS-Datenbank (extern) stellt Amazon RDS-Ressourcen dar, die in einer anderen Vorlage definiert sind. Dies umfasst:

- Amazon RDS DB Cluster oder Instance, die in einer anderen Vorlage definiert ist
- Amazon RDS DB Proxy

Die erweiterte Komponentenkarte RDS-Datenbank (extern) ist in der Ressourcenpalette verfügbar.



Um diese Karte zu verwenden, ziehen Sie sie auf die Infrastructure Composer-Arbeitsfläche, konfigurieren Sie sie und verbinden Sie sie mit anderen Ressourcen.

Sie können Ihre Anwendung mit dem externen Amazon RDS verbinden DB Cluster oder Instanz über eine Lambda-Funktion.

### Voraussetzungen

Um diese Funktion nutzen zu können, müssen Sie die folgenden Anforderungen erfüllen:

1. Ihr externer Amazon RDS DB Cluster, Instance oder Proxy müssen verwendet werden, AWS Secrets Manager um das Benutzerkennwort zu verwalten. Weitere Informationen finden Sie unter [Passwortverwaltung mit Amazon RDS und AWS Secrets Manager](#) im Amazon RDS-Benutzerhandbuch.
2. Ihre Anwendung in Infrastructure Composer muss ein neues Projekt sein oder ursprünglich in Infrastructure Composer erstellt worden sein.

### Verfahren

#### Schritt 1: Konfigurieren Sie die externe RDS-Datenbankkarte

Ziehen Sie aus der Ressourcenpalette eine erweiterte RDS-Datenbank-Komponentenkarte (extern) auf die Arbeitsfläche.

Wählen Sie die Karte und dann Details aus oder doppelklicken Sie auf die Karte, um den Bereich mit den Ressourceneigenschaften aufzurufen. Das Fenster mit den Ressourceneigenschaften der Karte wird angezeigt:

Details | Group | Delete

VPC

RDS Database (External)

ExternalRDS

## RDS Database (External)

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

---

**Logical ID**

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

ExternalRDS

**Database Secret**

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

**Database Hostname**

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

**Database Port**

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

Sie können hier Folgendes konfigurieren:

- **Logische ID** — Ein eindeutiger Name für Ihren externen Amazon RDS DB Cluster, Instance oder Proxy. Diese ID muss nicht mit dem logischen ID-Wert Ihres externen Amazon RDS übereinstimmen DB Ressource.
- **Datenbankgeheimnis** — Eine Kennung für das AWS Secrets Manager Geheimnis, das mit Ihrem Amazon RDS verknüpft ist DB Cluster, Instance oder Proxy. Dieses Feld akzeptiert die folgenden Werte:
  - **Statischer Wert** — Eine eindeutige Kennung des Datenbankgeheimnisses, z. B. der geheime ARN. Im Folgenden wird ein Beispiel gezeigt: `arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-name-1a2b3c`. Weitere

Informationen finden Sie unter [AWS Secrets Manager -Konzepte](#) im AWS Secrets Manager - Benutzerhandbuch.

- **Ausgabewert** — Wenn ein Secrets Manager Manager-Geheimnis bereitgestellt wird AWS CloudFormation, wird ein Ausgabewert erstellt. Sie können den Ausgabewert hier mithilfe der [Fn::ImportValue](#) systeminternen Funktion angeben. Beispiel, `!ImportValue MySecret`.
- **Wert aus dem SSM-Parameterspeicher** — Sie können Ihr Geheimnis im SSM-Parameterspeicher speichern und seinen Wert mithilfe einer dynamischen Referenz angeben. Beispiel, `{{resolve:ssm:MySecret}}`. Weitere Informationen finden Sie unter [SSM-Parameter](#) im AWS CloudFormation Benutzerhandbuch.
- **Datenbank-Hostname** — Der Hostname, der für die Verbindung zu Ihrem Amazon RDS verwendet werden kann DB Cluster, Instance oder Proxy. Dieser Wert ist in der externen Vorlage angegeben, die Ihre Amazon RDS-Ressource definiert. Die folgenden Werte werden akzeptiert:
  - **Statischer Wert** — Eine eindeutige Kennung des Datenbank-Hostnamens, z. B. die Endpunktadresse. Im Folgenden wird ein Beispiel gezeigt: `mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-2.rds.amazonaws.com`.
  - **Ausgabewert** — Der Ausgabewert eines bereitgestellten Amazon RDS DB Cluster, Instance oder Proxy. Sie können den Ausgabewert mithilfe der [Fn::ImportValue](#) systeminternen Funktion angeben. Beispiel, `!ImportValue myStack-myDatabase-abcd1234`.
  - **Wert aus dem SSM-Parameterspeicher** — Sie können den Datenbank-Hostnamen im SSM-Parameterspeicher speichern und seinen Wert mithilfe einer dynamischen Referenz angeben. Beispiel, `{{resolve:ssm:MyDatabase}}`.
- **Datenbank-Port** — Die Portnummer, die für die Verbindung zu Ihrem Amazon RDS verwendet werden kann DB Cluster, Instance oder Proxy. Dieser Wert ist in der externen Vorlage angegeben, die Ihre Amazon RDS-Ressource definiert. Die folgenden Werte werden akzeptiert:
  - **Statischer Wert** — Der Datenbankport. Beispiel, `3306`.
  - **Ausgabewert** — Der Ausgabewert eines bereitgestellten Amazon RDS DB Cluster, Instance oder Proxy. Beispiel, `!ImportValue myStack-MyRDSInstancePort`.
  - **Wert aus dem SSM-Parameterspeicher** — Sie können den Datenbank-Hostnamen im SSM-Parameterspeicher speichern und seinen Wert mithilfe einer dynamischen Referenz angeben. Beispiel, `{{resolve:ssm:MyRDSInstancePort}}`.

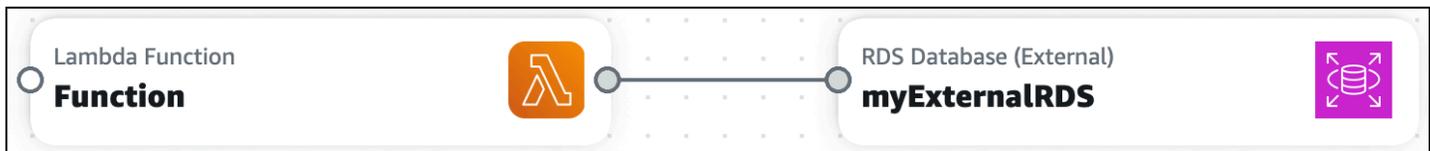
**Note**

Nur der logische ID-Wert muss hier konfiguriert werden. Sie können die anderen Eigenschaften bei der Bereitstellung konfigurieren, wenn Sie dies bevorzugen.

Schritt 2: Connect eine Lambda-Funktionskarte an

Ziehen Sie aus der Ressourcenpalette eine mit Lambda-Funktionen erweiterte Komponentenkarte auf die Arbeitsfläche.

Connect den linken Anschluss der Lambda-Funktionskarte mit dem rechten Anschluss der RDS-Datenbankkarte (extern).



Infrastructure Composer stellt Ihre Vorlage bereit, um diese Verbindung zu erleichtern.

Was unternimmt Infrastructure Composer, um Ihre Verbindung herzustellen

Wenn Sie das oben aufgeführte Verfahren abgeschlossen haben, führt Infrastructure Composer bestimmte Aktionen aus, um Ihre Lambda-Funktion mit Ihrer Datenbank zu verbinden.

Bei der Angabe des externen Amazon RDS DB Cluster, Instance oder Proxy

Wenn Sie eine (externe) RDS-Datenbankkarte auf die Arbeitsfläche ziehen, aktualisiert Infrastructure Composer die Parameters Abschnitte Metadata und die Abschnitte Ihrer Vorlage nach Bedarf. Im Folgenden wird ein Beispiel gezeigt:

```
Metadata:
  AWS::Composer::ExternalResources:
    ExternalRDS:
      Type: externalRDS
      Settings:
        Port: !Ref ExternalRDSPort
        Hostname: !Ref ExternalRDSHostname
        SecretArn: !Ref ExternalRDSSecretArn
Parameters:
```

```

ExternalRDSPort:
  Type: Number
ExternalRDSHostname:
  Type: String
ExternalRDSecretArn:
  Type: String

```

[Metadaten](#) sind ein AWS CloudFormation Vorlagenabschnitt, in dem Details zu Ihrer Vorlage gespeichert werden. Metadaten, die für Infrastructure Composer spezifisch sind, werden unter dem `AWS::Composer::ExternalResources` Metadatenschlüssel gespeichert. Hier speichert Infrastructure Composer die Werte, die Sie für Ihren Amazon RDS angeben DB Cluster, Instance oder Proxy.

Der Abschnitt „[Parameter](#)“ einer AWS CloudFormation Vorlage wird verwendet, um benutzerdefinierte Werte zu speichern, die bei der Bereitstellung in Ihre gesamte Vorlage eingefügt werden können. Abhängig von der Art der Werte, die Sie angeben, kann Infrastructure Composer hier Werte für Ihren Amazon RDS speichern. DB Cluster, Instance oder Proxy und spezifizieren Sie sie in Ihrer gesamten Vorlage.

Zeichenkettenwerte im `Parameters` Abschnitt `Metadata` und verwenden den logischen ID-Wert, den Sie auf Ihrer RDS-Datenbankkarte (`extern`) angeben. Wenn Sie die logische ID aktualisieren, ändern sich die Zeichenkettenwerte.

Wenn Sie die Lambda-Funktion mit Ihrer Datenbank verbinden

Wenn Sie eine Lambda-Funktionskarte mit der (externen) RDS-Datenbankkarte verbinden, stellt Infrastructure Composer Umgebungsvariablen und AWS Identity and Access Management (IAM-) Richtlinien bereit. Im Folgenden wird ein Beispiel gezeigt:

```

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
```

[Umgebungsvariablen](#) sind Variablen, die von Ihrer Funktion zur Laufzeit verwendet werden können. Weitere Informationen finden Sie unter [Verwenden von Lambda-Umgebungsvariablen](#) im AWS Lambda Entwicklerhandbuch.

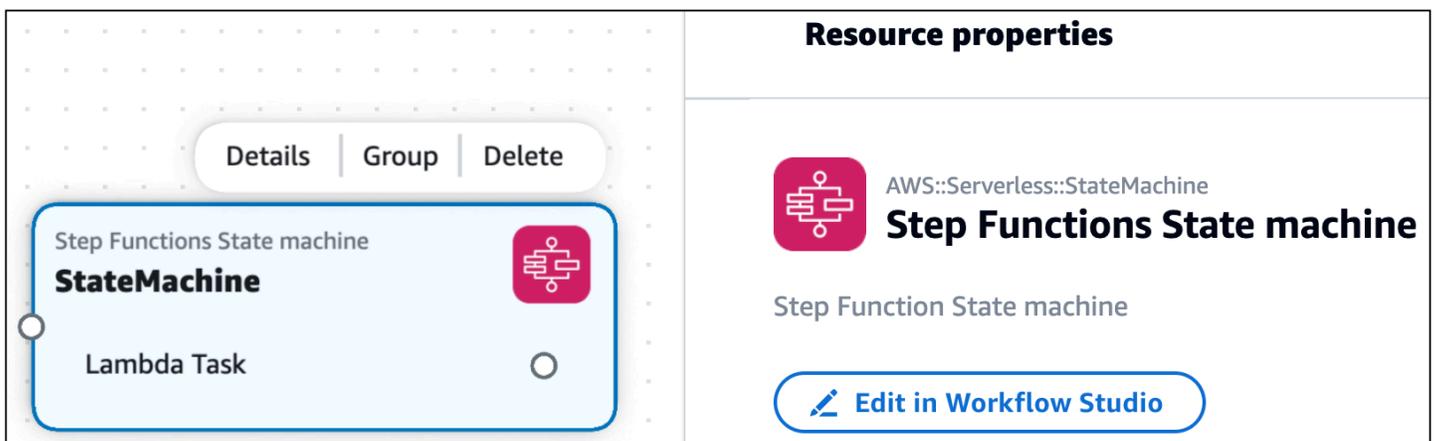
[Richtlinien](#) gewähren Berechtigungen für Ihre Funktion. Hier erstellt Infrastructure Composer eine Richtlinie, die den Lesezugriff von Ihrer Funktion auf Secrets Manager ermöglicht, um Ihr Passwort für den Zugriff auf den Amazon RDS zu erhalten DB Cluster, Instance oder Proxy.

## Verwenden AWS-Infrastruktur-Composer mit AWS Step Functions

AWS-Infrastruktur-Composer bietet eine Integration mit [AWS Step Functions Workflow Studio](#). Verwenden Sie Infrastructure Composer, um Folgendes zu tun:

- Step Functions starten Workflow Studio direkt in Infrastructure Composer.
- Erstellen und verwalten Sie neue Workflows oder importieren Sie bestehende Workflows in Infrastructure Composer.
- Integrieren Sie Ihre Workflows mithilfe der Infrastructure Composer-Arbeitsfläche in andere AWS Ressourcen.

Das folgende Bild zeigt eine Step Functions State Machine-Karte



Mit Step Functions Workflow Studio In Infrastructure Composer können Sie die Vorteile von zwei leistungsstarken visuellen Designern an einem einzigen Ort nutzen. Während Sie Ihren Workflow und Ihre Anwendung entwerfen, erstellt Infrastructure Composer Ihre Infrastruktur als Code (IaC), um Sie bei der Implementierung zu unterstützen.

## Themen

- [IAM-Richtlinien](#)
- [Erste Schritte mit Step Functions Workflow Studio im Infrastructure Composer](#)
- [Step Functions verwenden Workflow Studio im Infrastructure Composer](#)
- [Weitere Informationen](#)

## IAM-Richtlinien

Wenn Sie Aufgaben aus Ihrem Workflow mit Ressourcen verbinden, erstellt Infrastructure Composer automatisch die AWS Identity and Access Management (IAM-) Richtlinien, die für die Autorisierung der Interaktion zwischen Ihren Ressourcen erforderlich sind. Im Folgenden wird ein Beispiel gezeigt:

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

Bei Bedarf können Sie Ihrer Vorlage weitere IAM-Richtlinien hinzufügen.

## Erste Schritte mit Step Functions Workflow Studio im Infrastructure Composer

Zu Beginn können Sie neue Workflows erstellen oder bestehende Workflows importieren.

Um einen neuen Workflow zu erstellen

1. Ziehen Sie aus der Ressourcenpalette eine für Step Functions State Machine optimierte Komponentenkarte auf die Arbeitsfläche.



Wenn Sie eine Step Functions State-Maschinenkarte auf die Arbeitsfläche ziehen, erstellt Infrastructure Composer Folgendes:

- Eine [AWS::Serverless::StateMachine](#) Ressource, die Ihre Zustandsmaschine definiert. Standardmäßig erstellt Infrastructure Composer einen Standard-Workflow. Um einen Express-Workflow zu erstellen, ändern Sie den Type Wert in Ihrer Vorlage von STANDARD auf EXPRESS.
  - Eine [AWS::Logs::LogGroup](#) Ressource, die eine CloudWatch Amazon-Protokollgruppe für Ihren Zustandsmaschine definiert.
2. Öffnen Sie den Bereich mit den Ressourceneigenschaften der Karte und wählen Sie „In Workflow Studio bearbeiten“, um ihn zu öffnen Workflow Studio innerhalb von Infrastructure Composer.

Step Functions Workflow Studio wird im Entwurfsmodus geöffnet. Weitere Informationen finden Sie im AWS Step Functions Entwicklerhandbuch unter [Designmodus](#).

 Note

Sie können Infrastructure Composer so ändern, dass Ihre Zustandsmaschinen-Definition in einer externen Datei gespeichert wird. Weitere Informationen hierzu finden Sie unter [Arbeiten mit externen Dateien](#).

3. Erstellen Sie Ihren Workflow und wählen Sie Speichern. Um den Vorgang zu beenden Workflow Studio, wählen Sie Zurück zu Infrastructure Composer.

Infrastructure Composer definiert Ihren Workflow anhand der Definition Eigenschaft der `AWS::Serverless::StateMachine` Ressource.

4. Sie können Ihren Workflow ändern, indem Sie einen der folgenden Schritte ausführen:
- Öffnen Workflow Studio wiederholen Sie und ändern Sie Ihren Workflow.
  - Für Infrastructure Composer können Sie von der Konsole aus die Vorlagenansicht Ihrer Anwendung öffnen und Ihre Vorlage ändern. Wenn Sie die lokale Synchronisierung verwenden, können Sie Ihren Workflow in Ihrer lokalen IDE ändern. Infrastructure Composer erkennt Ihre Änderungen und aktualisiert Ihren Workflow in Infrastructure Composer.
  - Für Infrastructure Composer aus dem Toolkit for VS Code können Sie Ihre Vorlage direkt ändern. Infrastructure Composer erkennt Ihre Änderungen und aktualisiert Ihren Workflow in Infrastructure Composer.

## Um bestehende Workflows zu importieren

Sie können Workflows aus Anwendungen importieren, die mithilfe von AWS Serverless Application Model (AWS SAM) -Vorlagen definiert wurden. Verwenden Sie eine beliebige Zustandsmaschine, die mit dem `AWS::Serverless::StateMachine` Ressourcentyp definiert ist, und sie wird als Step Functions State Machine-erweiterte Komponentenkarte visualisiert, die Sie zum Starten verwenden können Workflow Studio.

Die `AWS::Serverless::StateMachine` Ressource kann Workflows mithilfe einer der folgenden Eigenschaften definieren:

- [Definition](#)— Der Workflow ist in der AWS SAM Vorlage als Objekt definiert.
- [DefinitionUri](#)— Der Workflow wird in einer externen Datei in der [Sprache Amazon States](#) definiert. Der lokale Pfad der Datei wird dann mit dieser Eigenschaft angegeben.

### Definitionseigenschaft

#### Infrastructure Composer von der Konsole aus

Für Workflows, die mithilfe der `Definition` Eigenschaft definiert wurden, können Sie eine einzelne Vorlage oder das gesamte Projekt importieren.

- Vorlage — Anweisungen zum Importieren einer Vorlage finden Sie unter [Importieren Sie eine vorhandene Projektvorlage in die Infrastructure Composer-Konsole](#). Um Änderungen zu speichern, die Sie in Infrastructure Composer vornehmen, müssen Sie Ihre Vorlage exportieren.
- Projekt — Wenn Sie ein Projekt importieren, müssen Sie die lokale Synchronisierung aktivieren. Von Ihnen vorgenommene Änderungen werden automatisch auf Ihrem lokalen Computer gespeichert. Anweisungen zum Importieren eines Projekts finden Sie unter [Importieren Sie einen vorhandenen Projektordner in die Infrastructure Composer-Konsole](#).

#### Infrastructure Composer aus dem Toolkit for VS Code

Für Workflows, die mithilfe der `Definition` Eigenschaft definiert wurden, können Sie Infrastructure Composer von Ihrer Vorlage aus öffnen. Detaillierte Anweisungen finden Sie unter [Greifen Sie auf Infrastructure Composer zu über AWS Toolkit for Visual Studio Code](#).

## DefinitionUri Eigenschaft

### Infrastructure Composer von der Konsole aus

Für Workflows, die mithilfe der `DefinitionUri` Eigenschaft definiert wurden, müssen Sie das Projekt importieren und die lokale Synchronisierung aktivieren. Anweisungen zum Importieren eines Projekts finden Sie unter [Importieren Sie einen vorhandenen Projektordner in die Infrastructure Composer-Konsole](#).

### Infrastructure Composer aus dem Toolkit for VS Code

Für Workflows, die mithilfe der `DefinitionUri` Eigenschaft definiert wurden, können Sie Infrastructure Composer von Ihrer Vorlage aus öffnen. Detaillierte Anweisungen finden Sie unter [Greifen Sie auf Infrastructure Composer zu über AWS Toolkit for Visual Studio Code](#).

## Step Functions verwenden Workflow Studio im Infrastructure Composer

### Workflows erstellen

Infrastructure Composer verwendet Definitionersetzungen, um Workflow-Aufgaben Ressourcen in Ihrer Anwendung zuzuordnen. Weitere Informationen zu Definitionersetzungen finden Sie [DefinitionSubstitutions](#) im AWS Serverless Application Model Entwicklerhandbuch.

Wenn Sie Aufgaben erstellen in Workflow Studio, geben Sie für jede Aufgabe eine Definitionersetzung an. Anschließend können Sie Aufgaben mit Ressourcen auf der Infrastructure Composer-Arbeitsfläche verbinden.

### Um eine Definitionersetzung anzugeben in Workflow Studio

1. Öffnen Sie die Registerkarte Konfiguration der Aufgabe und suchen Sie das Feld API-Parameter.

The screenshot displays a state machine definition on the left and its configuration panel on the right.

**State Machine Definition:**

```

graph TD
    Start((Start)) --> Lambda1[Lambda: Invoke  
Check Stock Value]
    Lambda1 --> Choice[Choice state  
Choice]
    Choice -- "$stock_price <= 50" --> Lambda2[Lambda: Invoke  
Buy Stock]
    Choice -- Default --> Lambda3[Lambda: Invoke  
Sell Stock]
    Lambda2 --> DynamoDB[DynamoDB: PutItem  
Record Transaction]
    Lambda3 --> DynamoDB
    DynamoDB --> End((End))
  
```

**Configuration Panel (Check Stock Value):**

- State name:** Check Stock Value
- API:** Lambda: Invoke
- Integration type:** Optimized
- API Parameters:** Edit as JSON (checked)
- Function name:** Enter a CloudFormation substitution (Substitutions can be used to parameterize your workflow definition which will be...)
- Substitution:** `#{LambdaFunction1}`

Substitutions must be specified in `#{dollar_sign_brace}` notation. They will be mapped via the `DefinitionSubstitution` property inside your `StateMachine` resource in the Application Composer Canvas.

2. Wenn das Feld API-Parameter über eine Dropdownoption verfügt, wählen Sie **AWS CloudFormation Ersetzung** eingeben aus. Geben Sie dann einen eindeutigen Namen ein.

Geben Sie für Aufgaben, die eine Verbindung zu derselben Ressource herstellen, für jede Aufgabe dieselbe Definitionersersetzung an. Um eine bestehende Definitionersersetzung zu verwenden, wählen Sie „**AWS CloudFormation Substitution auswählen**“ und wählen Sie die zu verwendende Substitution aus.

3. Wenn das Feld API-Parameter ein JSON-Objekt enthält, ändern Sie den Eintrag, der den Ressourcennamen angibt, so, dass eine Definitionersersetzung verwendet wird. Im folgenden Beispiel ändern wir `"MyDynamoDBTable"` zu `"#{RecordTransaction}"`.

The screenshot shows a workflow diagram on the left and the configuration for the 'Record Transaction' state on the right.

**Workflow Diagram:**

```

    graph TD
      Start((Start)) --> Check[Lambda: Invoke Check Stock Value]
      Check --> Choice[Choice state Choice]
      Choice -- "$stock_price <= 50" --> Buy[Lambda: Invoke Buy Stock]
      Choice -- Default --> Sell[Lambda: Invoke Sell Stock]
      Buy --> Record[DynamoDB: PutItem Record Transaction]
      Sell --> Record
      Record --> End((End))
    
```

**Record Transaction Configuration:**

- State name:** Record Transaction
- API:** DynamoDB: PutItem
- Integration type:** Optimized
- API Parameters:**

```

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

```

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

4. Wählen Sie Speichern und zurück zu Infrastructure Composer.

Die Aufgaben aus Ihrem Workflow werden auf der Step Functions State-Maschinenkarte visualisiert.

The screenshot shows the 'Step Functions State machine' console. The state machine is named 'StateMachine' and contains four tasks:

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

Each task has a radio button next to it, indicating that all tasks are currently selected.

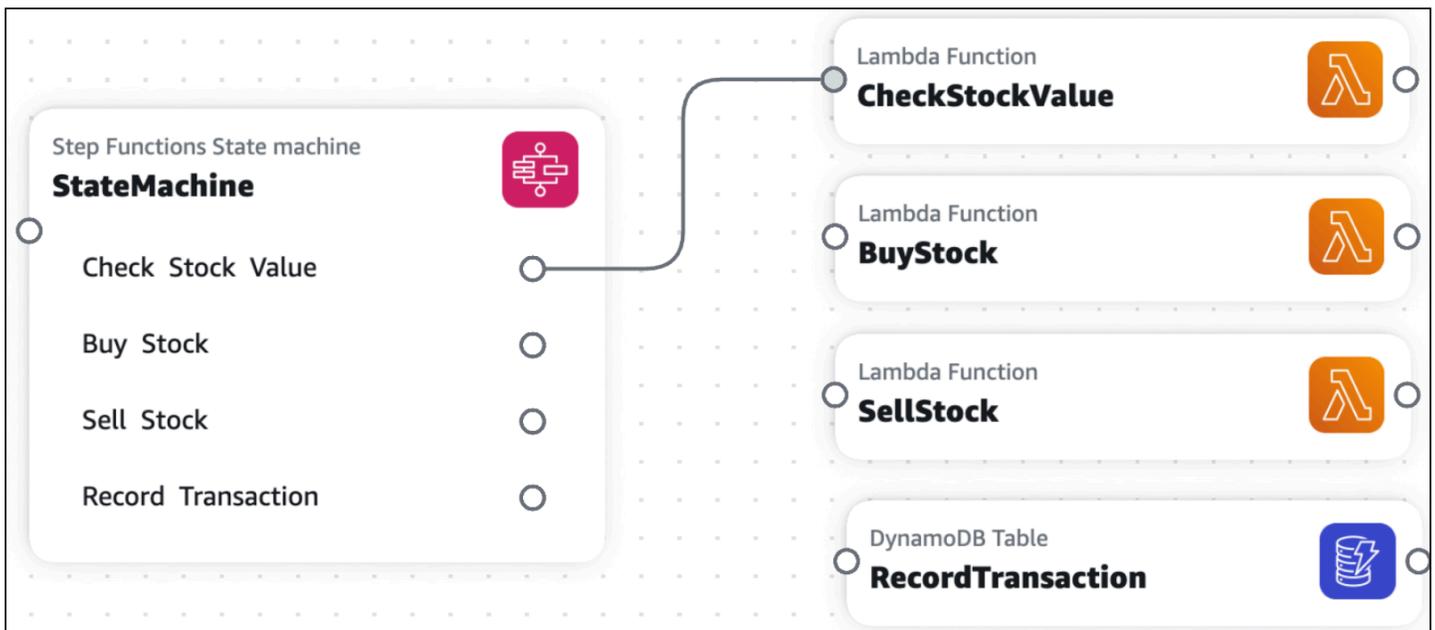
## Ressourcen mit Workflow-Aufgaben verbinden

Sie können in Infrastructure Composer Verbindungen zwischen unterstützten Workflow-Aufgaben und unterstützten Infrastructure Composer-Karten herstellen.

- Unterstützte Workflow-Aufgaben — Aufgaben AWS-Services dafür sind für Step Functions optimiert. Weitere Informationen finden Sie unter [Optimierte Integrationen für Step Functions](#) im AWS Step Functions Entwicklerhandbuch.
- Unterstützte Infrastructure Composer-Karten — Erweiterte Komponentenkarten werden unterstützt. Weitere Informationen zu Karten in Infrastructure Composer finden Sie unter [Karten in Infrastructure Composer konfigurieren und ändern](#).

Beim Herstellen einer Verbindung müssen Aufgabe und Karte übereinstimmen. AWS-Service Sie können beispielsweise eine Workflow-Aufgabe, die eine Lambda-Funktion aufruft, mit einer erweiterten Lambda-Funktions-Komponentenkarte verbinden.

Um eine Verbindung herzustellen, klicken Sie auf den Port einer Aufgabe und ziehen Sie ihn auf den linken Anschluss einer erweiterten Komponentenkarte.



Infrastructure Composer aktualisiert Ihren DefinitionSubstitution Wert automatisch, um Ihre Verbindung zu definieren. Im Folgenden wird ein Beispiel gezeigt:

```
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:
    ...
```

## Arbeiten mit externen Dateien

Wenn Sie einen Workflow über die Step Functions State Machine-Karte erstellen, speichert Infrastructure Composer Ihre Zustandsmaschinen-Definition mithilfe der `Definition` Eigenschaft in Ihrer Vorlage. Sie können Infrastructure Composer so konfigurieren, dass Ihre Zustandsmaschinen-Definition in einer externen Datei gespeichert wird.

### Note

Um diese Funktion mit Infrastructure Composer von aus verwenden zu können AWS Management Console, müssen Sie die lokale Synchronisierung aktiviert haben. Weitere Informationen finden Sie unter [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#).

Um Ihre State Machine-Definition in einer externen Datei zu speichern

1. Öffnen Sie den Bereich mit den Ressourceneigenschaften Ihrer Step Functions State-Maschinenkarte.

2. Wählen Sie die Option Externe Datei für die Zustandsmaschinen-Definition verwenden.
3. Geben Sie einen relativen Pfad und Namen für Ihre Zustandsmaschinen-Definitionsdatei an.
4. Wählen Sie Save (Speichern) aus.

Infrastructure Composer führt Folgendes aus:

1. Verschieben Sie Ihre State-Machine-Definition aus dem Definition Feld in Ihre externe Datei.
2. Speichern Sie Ihre State Machine-Definition in einer externen Datei in der Amazon States-Sprache.
3. Ändern Sie Ihre Vorlage so, dass sie mithilfe des DefinitionUri Felds auf die externe Datei verweist.

Weitere Informationen

Weitere Informationen zu Step Functions in Infrastructure Composer finden Sie im Folgenden:

- [Verwenden Workflow Studio in Infrastructure Composer](#) im AWS Step Functions Entwicklerhandbuch.
- [DefinitionSubstitutions in AWS SAM Vorlagen](#) im AWS Step Functions Entwicklerhandbuch.

## Standardkarten in Infrastructure Composer

Alle AWS CloudFormation Ressourcen können in der Ressourcenpalette als Standard-IaC-Ressourcenkarten verwendet werden. Nachdem eine Standard-IaC-Ressourcenkarte auf die visuelle Leinwand gezogen wurde, wird sie zu einer Standard-Komponentenkarte. Das bedeutet einfach, dass es sich bei der Karte um eine oder mehrere Standard-IaC-Ressourcen handelt. Weitere Beispiele und Einzelheiten finden Sie in den Themen in diesem Abschnitt.

Sie können Ihren Infrastrukturcode in der Vorlagenansicht und im Fenster mit den Ressourceneigenschaften ändern. Im Folgenden finden Sie beispielsweise ein Beispiel für die Startvorlage einer `Alexa::ASK::Skill` Standard-IaC-Ressource:

```
Resources:
  Skill:
    Type: Alexa::ASK::Skill
  Properties:
```

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

Eine standardmäßige Startvorlage für eine IaC-Ressourcenkarte besteht aus folgenden Elementen:

- Der AWS CloudFormation Ressourcentyp.
- Erforderliche oder häufig verwendete Eigenschaften.
- Der erforderliche Typ des Werts, der für jede Eigenschaft bereitgestellt werden soll.

#### Note

Sie können Folgendes verwenden ... Amazon Q um Vorschläge für Infrastrukturcodes für Standardressourcenkarten zu generieren. Weitere Informationen hierzu finden Sie unter [Verwenden AWS-Infrastruktur-Composer mit Amazon Q Developer](#).

## Verfahren

Sie können den Infrastrukturcode für jede Ressource auf einer Standardkomponentenkarte im Bereich mit den Ressourceneigenschaften ändern.

Um eine Standardkomponentenkarte zu ändern

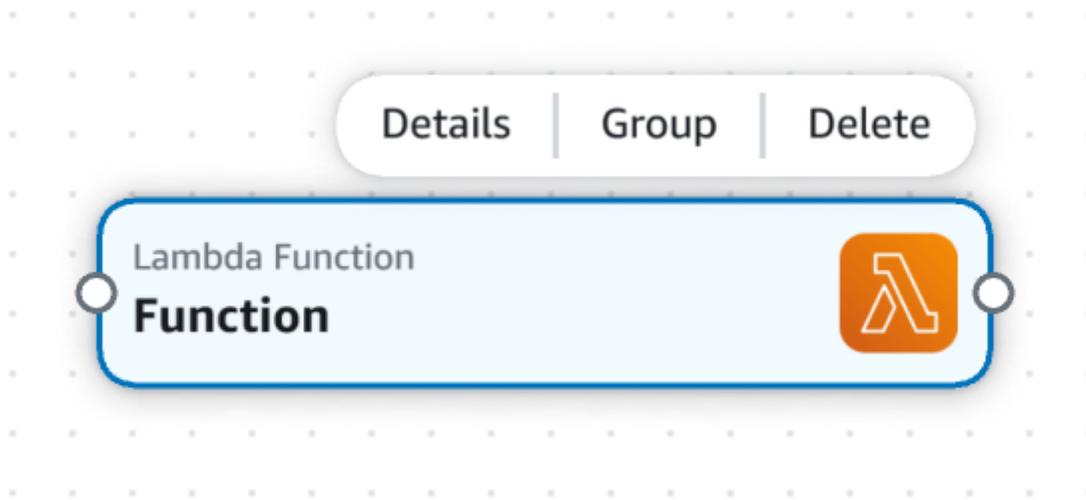
1. Öffnen Sie den Bereich mit den Ressourceneigenschaften der Standard-IaC-Komponentenkarte.
2. Wählen Sie im Feld Bearbeitung aus der Dropdownliste die zu bearbeitende Standard-IaC-Ressource aus.
3. Ändern Sie Ihren Infrastrukturcode und klicken Sie auf Speichern.

## Karten in Infrastructure Composer löschen

Dieser Abschnitt enthält Anweisungen zum Löschen von Karten in AWS-Infrastruktur-Composer.

## Verbesserte Komponentenkarten

Um eine erweiterte Komponentenkarte zu löschen, wählen Sie eine Karte aus, die Sie auf der visuellen Leinwand platziert haben. Wählen Sie im Menü Kartenaktionen die Option Löschen aus.



## Standardkomponentenkarten

Um Standardkomponentenkarten zu löschen, müssen Sie den Infrastrukturcode für jede AWS CloudFormation Ressource manuell aus Ihrer Vorlage entfernen. Dies lässt sich auf einfache Weise wie folgt bewerkstelligen:

1. Notieren Sie sich die logische ID der zu löschenden Ressource.
2. Suchen Sie in Ihrer Vorlage die Ressource anhand ihrer logischen ID aus dem Outputs Abschnitt `Resources` oder.
3. Löschen Sie die Ressource aus Ihrer Vorlage. Dazu gehören die logische ID der Ressource und ihre verschachtelten Werte, z. B. `Type` und `Properties`.
4. Überprüfen Sie in der Canvas-Ansicht, ob die Ressource aus Ihrem Canvas entfernt wurde.

## Codeaktualisierungen mit dem Change Inspector in Infrastructure Composer anzeigen

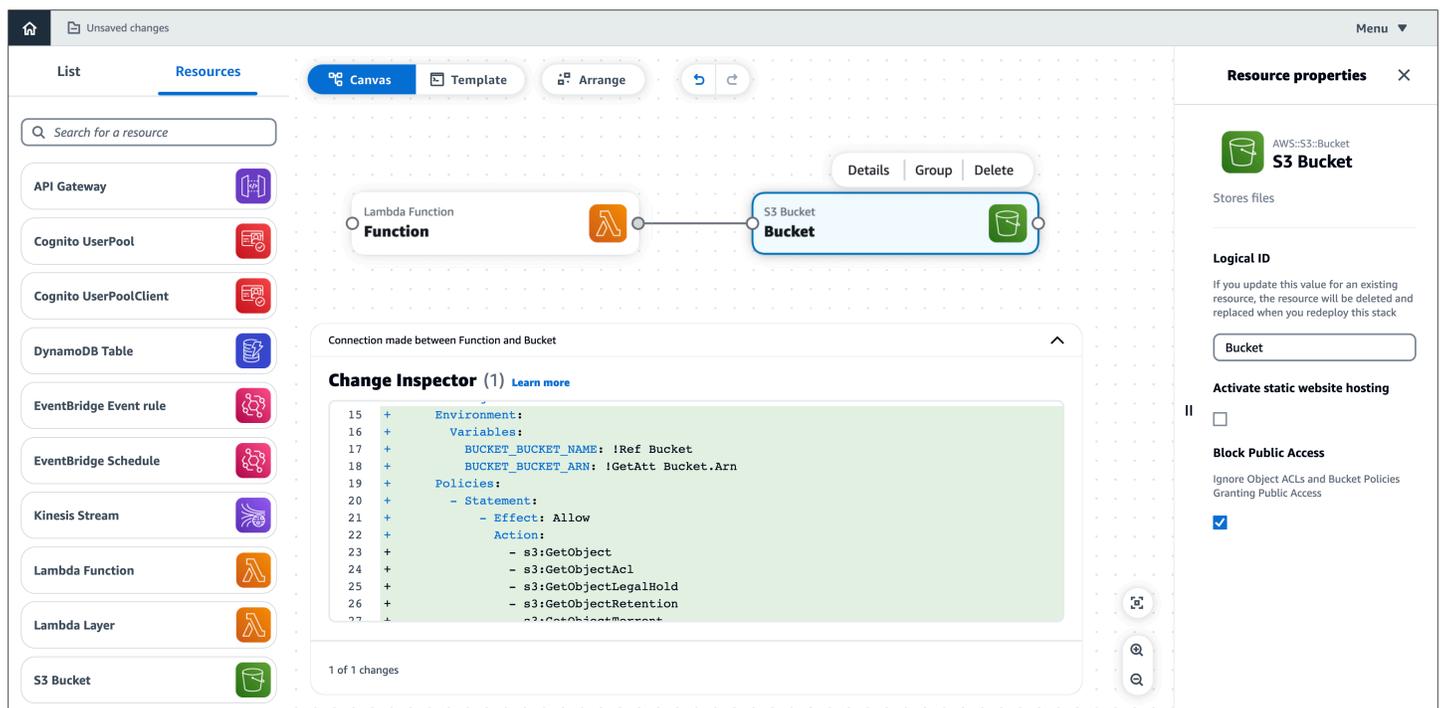
Während Sie in der Infrastructure Composer-Konsole entwerfen, wird Ihr Infrastrukturcode automatisch erstellt. Verwenden Sie den Change Inspector, um Ihre Vorlagencode-Aktualisierungen anzusehen und zu erfahren, was Infrastructure Composer für Sie erstellt.

Dieses Thema behandelt die Verwendung von Infrastructure Composer aus der AWS Management Console oder der AWS Toolkit for Visual Studio Code Erweiterung.

Der Change Inspector ist ein visuelles Tool in Infrastructure Composer, das Ihnen aktuelle Codeaktualisierungen anzeigt.

- Während Sie Ihre Anwendung entwerfen, werden Meldungen am unteren Rand der visuellen Leinwand angezeigt. Diese Meldungen enthalten Kommentare zu den von Ihnen ausgeführten Aktionen.
- Wenn dies unterstützt wird, können Sie eine Nachricht erweitern, um den Change Inspector anzuzeigen.
- Der Change Inspector zeigt Codeänderungen aus Ihrer letzten Interaktion an.

Das folgende Beispiel zeigt, wie Change Inspector funktioniert:



The screenshot shows the AWS Infrastructure Composer interface. On the left is a 'Resources' list with various AWS services. The main canvas shows a 'Lambda Function' resource connected to an 'S3 Bucket' resource. A 'Change Inspector' window is open, displaying a list of changes. The first change is for the 'S3 Bucket' resource, showing a code diff for its configuration. The code diff includes the following sections:

```

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

```

At the bottom of the Change Inspector window, it indicates '1 of 1 changes'.

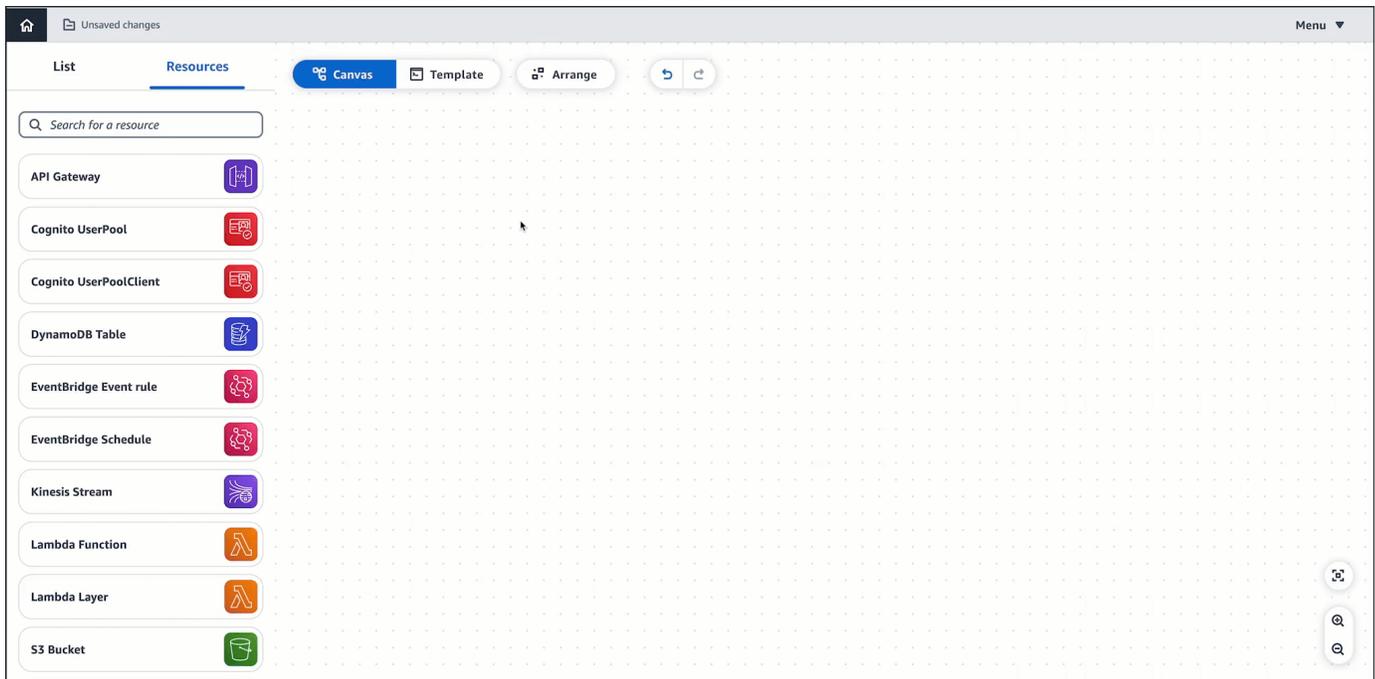
## Vorteile des Change Inspector

Der Change Inspector ist eine hervorragende Möglichkeit, den Vorlagencode anzuzeigen, den Infrastructure Composer für Sie erstellt. Es ist auch eine hervorragende Möglichkeit, um zu lernen, wie man Infrastrukturcode schreibt. Sehen Sie sich beim Entwerfen von Anwendungen in Infrastructure Composer die Codeaktualisierungen im Change Inspector an, um mehr über den Code zu erfahren, der für die Bereitstellung Ihres Designs erforderlich ist.

# Verfahren

So verwenden Sie den Change Inspector

1. Erweitern Sie eine Nachricht, um den Change Inspector aufzurufen.



2. Sehen Sie sich den Code an, der automatisch für Sie erstellt wurde.



- a. Grün hervorgehobener Code weist auf neu hinzugefügten Code hin.
  - b. Rot hervorgehobener Code weist auf neu entfernten Code hin.
  - c. Zeilennummern geben die Position innerhalb Ihrer Vorlage an.
3. Wenn mehrere Abschnitte Ihrer Vorlage aktualisiert wurden, organisiert der Change Inspector sie. Wählen Sie die Schaltflächen Zurück und Weiter, um alle Änderungen anzuzeigen.

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

Für Infrastructure Composer können Sie von der Konsole aus Codeänderungen im Kontext Ihrer gesamten Vorlage anzeigen, indem Sie die Vorlagenansicht verwenden. Sie können Infrastructure Composer auch mit einer lokalen IDE synchronisieren und Ihre gesamte Vorlage auf Ihrem lokalen Computer anzeigen. Weitere Informationen hierzu finden Sie unter [Connect die Infrastructure Composer-Konsole mit Ihrer lokalen IDE](#).

## Weitere Informationen

Weitere Informationen zu dem Code, den Infrastructure Composer erstellt, finden Sie im Folgenden:

- [Kartenverbindungen in Infrastructure Composer](#).

# Verweisen auf externe Dateien in Infrastructure Composer

Sie können externe Dateien mit Ihren AWS Serverless Application Model (AWS SAM) -Vorlagen verwenden, um wiederholten Code wiederzuverwenden und Ihre Projekte zu organisieren. Beispielsweise verfügen Sie möglicherweise über mehrere Amazon API Gateway Gateway-REST-API-Ressourcen, die durch einen OpenAPI Spezifikation. Anstatt das zu replizieren OpenAPI Mit dem Spezifikationscode in Ihrer Vorlage können Sie eine externe Datei erstellen und für jede Ihrer Ressourcen darauf verweisen.

AWS-Infrastruktur-Composer unterstützt die folgenden Anwendungsfälle für externe Dateien:

- API Gateway REST API Ressourcen, die von extern definiert sind OpenAPI Spezifikationsdateien.
- AWS Step Functions Zustandsmaschinenressourcen, die durch externe Zustandsmaschinen-Definitionsdateien definiert sind.

Weitere Informationen zur Konfiguration externer Dateien für unterstützte Ressourcen finden Sie im Folgenden:

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

## Note

Um mit Infrastructure Composer von der Infrastructure Composer-Konsole aus auf externe Dateien zu verweisen, müssen Sie Infrastructure Composer im lokalen Synchronisierungsmodus verwenden. Weitere Informationen finden Sie unter [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#).

## Themen

- [Bewährte Methoden für externe Referenzdateien von Infrastructure Composer](#)
- [Erstellen Sie eine externe Dateireferenz in Infrastructure Composer](#)
- [Laden Sie ein Projekt mit einer externen Dateireferenz in Infrastructure Composer](#)
- [Erstellen Sie eine Anwendung, die auf eine externe Datei in Infrastructure Composer verweist](#)
- [Verweisen Sie auf OpenAPI Spezifikation externe Datei mit Infrastructure Composer](#)

## Bewährte Methoden für externe Referenzdateien von Infrastructure Composer

### Verwenden Sie Infrastructure Composer mit einer lokalen IDE

Wenn Sie Infrastructure Composer mit einer lokalen IDE im lokalen Synchronisierungsmodus verwenden, können Sie Ihre lokale IDE verwenden, um externe Dateien anzuzeigen und zu ändern. Inhalte aus unterstützten externen Dateien, auf die in Ihrer Vorlage verwiesen wird, werden automatisch auf der Infrastructure Composer-Arbeitsfläche aktualisiert. Weitere Informationen hierzu finden Sie unter [Connect die Infrastructure Composer-Konsole mit Ihrer lokalen IDE](#).

### Bewahren Sie externe Dateien im übergeordneten Verzeichnis Ihres Projekts auf

Sie können Unterverzeichnisse im übergeordneten Verzeichnis Ihres Projekts erstellen, um Ihre externen Dateien zu organisieren. Infrastructure Composer kann nicht auf externe Dateien zugreifen, die in einem Verzeichnis außerhalb des übergeordneten Verzeichnisses Ihres Projekts gespeichert sind.

### Stellen Sie Ihre Anwendung mit dem bereit AWS SAM CLI

Wenn Sie Ihre Anwendung auf dem bereitstellen AWS Cloud, müssen lokale externe Dateien zunächst an einen zugänglichen Ort wie Amazon Simple Storage Service (Amazon S3) hochgeladen werden. Sie können die AWS SAM CLI verwenden, um diesen Vorgang automatisch zu vereinfachen. Weitere Informationen finden Sie unter [Hochladen lokaler Dateien bei der Bereitstellung](#) im AWS Serverless Application Model Entwicklerhandbuch.

## Erstellen Sie eine externe Dateireferenz in Infrastructure Composer

Sie können im Eigenschaftenbereich der unterstützten Ressourcen eine externe Dateireferenz erstellen.

Um eine externe Dateireferenz zu erstellen

1. Wählen Sie auf einer erweiterten API-Gateway-Komponentenkarte oder Step Functions die Option Details aus, um den Bereich mit den Ressourceneigenschaften aufzurufen.
2. Suchen Sie die Option Externe Datei verwenden und wählen Sie sie aus.
3. Geben Sie den relativen Pfad zur externen Datei an. Dies ist der Pfad von Ihrer `template.yaml` Datei zur externen Datei.

Um beispielsweise aus der Struktur des folgenden Projekts auf die `api-spec.yaml` externe Datei zu verweisen, geben Sie `./api-spec.yaml` als relativen Pfad an.

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

#### Note

Wenn die externe Datei und der angegebene Pfad nicht existieren, erstellt Infrastructure Composer sie.

4. Speichern Sie Ihre Änderungen.

## Laden Sie ein Projekt mit einer externen Dateireferenz in Infrastructure Composer

Folgen Sie den auf dieser Seite aufgeführten Schritten, um ein Infrastructure Composer-Projekt mit einer externen Dateireferenz zu laden.

Über die Infrastructure Composer-Konsole

1. Führen Sie die Schritte unter [Importieren Sie eine vorhandene Projektvorlage in die Infrastructure Composer-Konsole](#) aus.
2. Bestätigen Sie, dass Infrastructure Composer Sie auffordert, eine Verbindung zum Stammordner Ihres Projekts herzustellen

Wenn Ihr Browser die File System Access API unterstützt, werden Sie von Infrastructure Composer aufgefordert, eine Verbindung zum Stammordner Ihres Projekts herzustellen. Infrastructure Composer öffnet Ihr Projekt im lokalen Synchronisierungsmodus, um Ihre externe Datei zu unterstützen. Wenn die referenzierte externe Datei nicht unterstützt wird, erhalten Sie eine Fehlermeldung. Weitere Hinweise zu Fehlermeldungen finden Sie unter [Fehlerbehebung](#).

## Aus dem Toolkit for VS Code

1. Führen Sie die Schritte unter [Greifen Sie auf Infrastructure Composer zu über AWS Toolkit for Visual Studio Code](#) aus.
2. Öffnen Sie die Vorlage, die Sie anzeigen möchten, in Infrastructure Composer.

Wenn Sie über eine Vorlage auf Infrastructure Composer zugreifen, erkennt Infrastructure Composer Ihre externe Datei automatisch. Wenn die referenzierte externe Datei nicht unterstützt wird, erhalten Sie eine Fehlermeldung. Weitere Hinweise zu Fehlermeldungen finden Sie unter [Fehlerbehebung](#).

## Erstellen Sie eine Anwendung, die auf eine externe Datei in Infrastructure Composer verweist

In diesem Beispiel wird der AWS SAM CLI um eine Anwendung zu erstellen, die für ihre Zustandsmaschinen-Definition auf eine externe Datei verweist. Anschließend laden Sie Ihr Projekt in Infrastructure Composer, wobei Ihre externe Datei ordnungsgemäß referenziert wird.

### Beispiel

1. Verwenden Sie zunächst den AWS SAM `sam init` CLI-Befehl, um eine neue Anwendung mit dem Namen `demo` zu initialisieren. Wählen Sie während des interaktiven Ablaufs die Vorlage für den Schnellstart eines Workflows mit mehreren Schritten aus.

```
$ 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
```

```
...
```

Diese Anwendung verweist auf eine externe Datei für die State-Machine-Definition.

```
...
```

```
Resources:
```

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

Die externe Datei befindet sich im `statemachine` Unterverzeichnis unserer Anwendung.

```
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. Laden Sie als Nächstes Ihre Anwendung von der Konsole aus in Infrastructure Composer. Wählen Sie auf der Infrastructure Composer-Startseite die Option CloudFormation Vorlage laden aus.
3. Wählen Sie unseren demo Projektordner aus und lassen Sie die Eingabeaufforderung zum Anzeigen der Dateien erscheinen. Wählen Sie unsere `template.yaml` Datei aus und wählen Sie Erstellen. Wenn Sie dazu aufgefordert werden, wählen Sie Änderungen speichern.

## 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 erkennt automatisch die externe State-Machine-Definitionsdatei und lädt sie. Wählen Sie unsere `StockTradingStateMachine` Ressource und dann Details aus, um den Bereich mit den Ressourceneigenschaften anzuzeigen. Hier können Sie sehen, dass Infrastructure Composer automatisch eine Verbindung zu unserer externen State-Machine-Definitionsdatei hergestellt hat.

The screenshot shows the AWS Infrastructure Composer interface. The main canvas displays a resource graph with several components: `StockCheckerFunction`, `StockSellerFunction`, `StockBuyerFunction`, `TransactionTable`, and `ImplicitTimer`. The `StockTradingStateMachine` resource is highlighted, and its details are shown in a pop-up window. The details window shows the state machine definition, including the state `Check Stock Value` and its configuration. The right sidebar shows the `Resource properties` for the `Step Functions State machine` resource, including the logical ID `StockTradingStateMachine` and the state machine definition.

```

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

Alle an der State-Machine-Definitionsdatei vorgenommenen Änderungen werden automatisch in Infrastructure Composer übernommen.

## Verweisen Sie auf OpenAPI Spezifikation externe Datei mit Infrastructure Composer

In diesem Beispiel wird Infrastructure Composer von der Konsole aus verwendet, um auf ein externes Objekt zu verweisen OpenAPI Spezifikationsdatei, die ein API Gateway definiert REST API.

Erstellen Sie zunächst auf der Infrastructure Composer-Startseite ein neues Projekt.

Aktivieren Sie anschließend die lokale Synchronisierung, indem Sie im Menü die Option Lokale Synchronisierung aktivieren auswählen. Erstellen Sie einen neuen Ordner mit dem Namendemo, warten Sie auf die Aufforderung, Dateien anzuzeigen, und wählen Sie Aktivieren. Wenn Sie dazu aufgefordert werden, wählen Sie Änderungen speichern aus.

### Activate local sync ✕

Automatically sync your project to your local machine.

**Project location**  
Select the folder where you want your project files to be saved.

 **Select folder**

 demo

Cancel Activate

Ziehen Sie als Nächstes eine Amazon API Gateway Gateway-Karte auf die Leinwand. Wählen Sie Details aus, um den Bereich mit den Ressourceneigenschaften aufzurufen.

The screenshot shows the AWS CloudFormation console interface. On the left, there is a 'Resources' list with various AWS services. The main canvas displays a resource named 'Api' of type 'API Gateway' with a 'GET /' route. The right-hand panel shows the 'Resource properties' for the 'Api' resource, including fields for 'Logical ID' (set to 'Api'), 'Default authorizer' (set to 'None'), 'Method' (set to 'GET'), and 'Path'.

Konfigurieren Sie im Bereich mit den Ressourceneigenschaften Folgendes und speichern Sie es.

- Wählen Sie die Option Externe Datei für API-Definition verwenden.
- Eingabe `./api-spec.yaml` als relativer Pfad zur externen Datei

## Use external file for api definition



### Relative path to external file

`./api-spec.yaml`

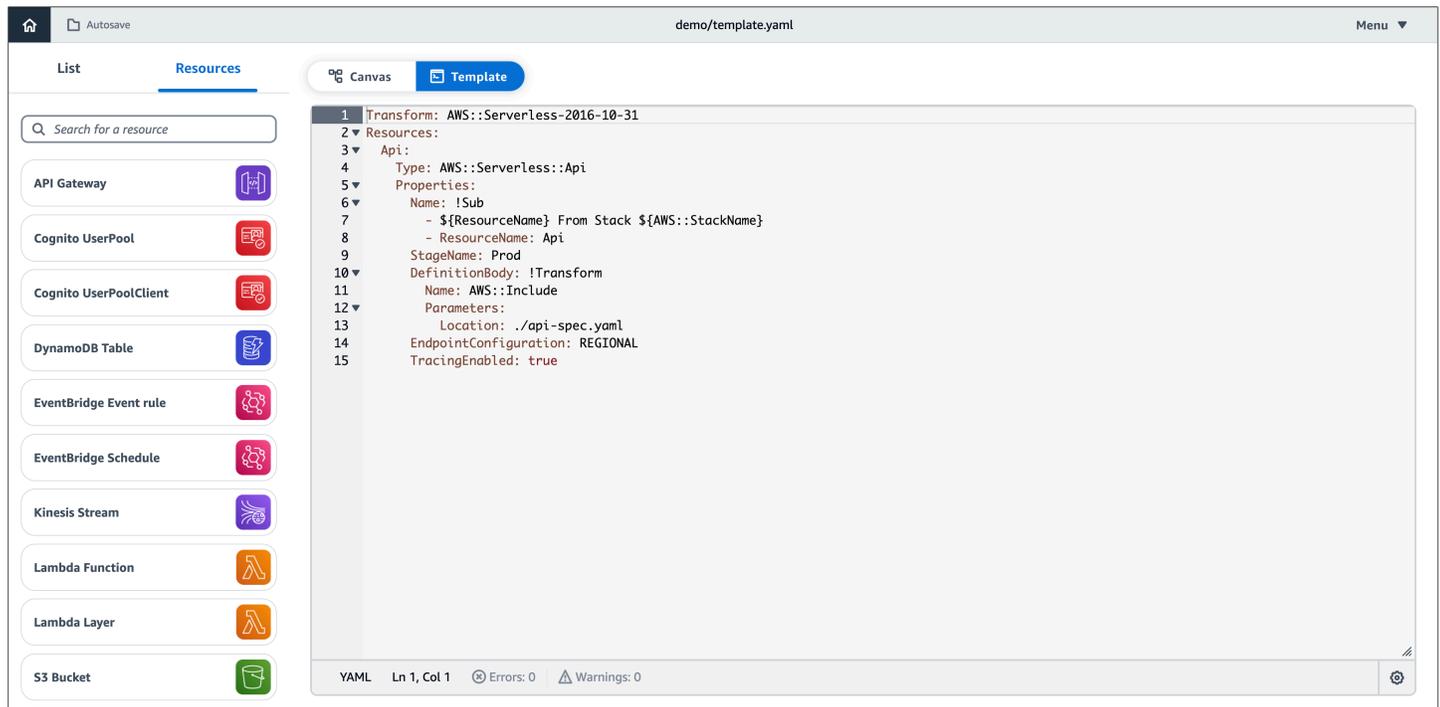
Dadurch wird das folgende Verzeichnis auf unserem lokalen Computer erstellt:

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

Jetzt können Sie die externe Datei auf unserem lokalen Computer konfigurieren. Öffnen Sie mit unserer IDE den Ordner, der `api-spec.yaml` sich in Ihrem Projektordner befindet. Ersetzen Sie den Inhalt durch Folgendes:

```
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
```

In der Infrastructure Composer-Vorlagenansicht können Sie sehen, dass Infrastructure Composer Ihre Vorlage automatisch aktualisiert hat, sodass sie auf die externe Datei verweist.

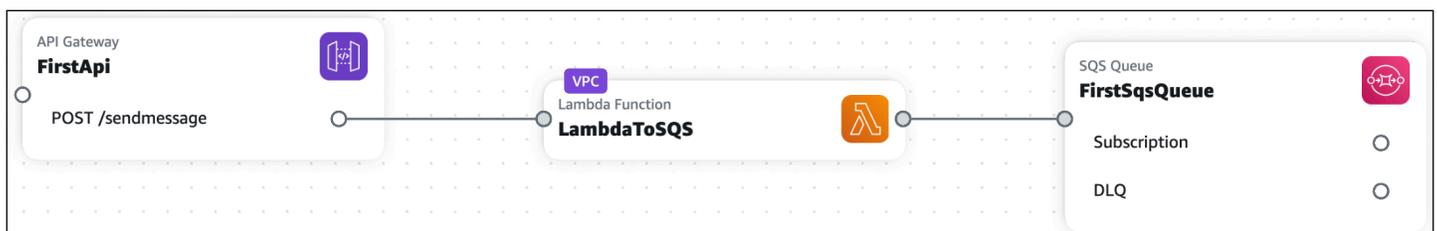


## Integrieren Sie Infrastructure Composer in Amazon Virtual Private Cloud (Amazon VPC)

AWS-Infrastruktur-Composer bietet eine Integration mit dem Amazon Virtual Private Cloud (Amazon VPC) -Service. Mit Infrastructure Composer können Sie Folgendes tun:

- Identifizieren Sie die Ressourcen auf Ihrer Leinwand, die sich in einer VPC befinden, anhand eines visuellen VPC-Tags.
- Konfigurieren Sie AWS Lambda Funktionen mithilfe einer VPCs externen Vorlage.

Die folgende Abbildung zeigt ein Beispiel für eine Anwendung mit einer Lambda-Funktion, die mit einer VPC konfiguriert ist.



Weitere Informationen zu Amazon VPC finden Sie unter [Was ist Amazon VPC?](#) im Amazon VPC-Benutzerhandbuch.

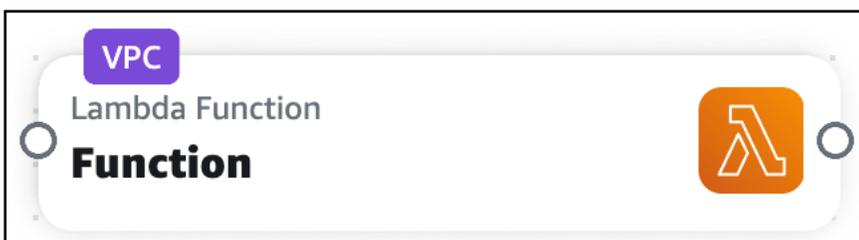
## Themen

- [Identifizieren Sie Infrastructure Composer-Ressourcen und zugehörige Informationen in einer VPC](#)
- [Lambda-Funktionen mit extern VPCs in Infrastructure Composer konfigurieren](#)
- [Parameter in importierten Vorlagen für eine externe VPC mit Infrastructure Composer](#)
- [Hinzufügen neuer Parameter zu importierten Vorlagen mit Infrastructure Composer](#)
- [Konfigurieren Sie eine Lambda-Funktion und eine VPC, die in einer anderen Vorlage definiert sind, mit Infrastructure Composer](#)

## Identifizieren Sie Infrastructure Composer-Ressourcen und zugehörige Informationen in einer VPC

Um Infrastructure Composer in Amazon VPC zu integrieren, müssen Sie zunächst die Ressourcen in einer VPC und die Informationen identifizieren, die für den Abschluss einer Integration erforderlich sind. Dazu gehören auch Konfigurationsinformationen zu Sicherheitsgruppen, Subnetzkennungen, Parametertypen, SSM-Typen und statischen Werttypen.

Infrastructure Composer visualisiert Ressourcen in einer VPC mithilfe eines VPC-Tags. Dieses Tag wird auf Karten auf der Arbeitsfläche angewendet. Das Folgende ist ein Beispiel für eine Lambda-Funktion mit einem VPC-Tag:



VPC-Tags werden auf Karten auf der Arbeitsfläche angewendet, wenn Sie wie folgt vorgehen:

- Konfigurieren Sie eine Lambda-Funktion mit einer VPC in Infrastructure Composer.
- Importieren Sie eine Vorlage, die Ressourcen enthält, die mit einer VPC konfiguriert wurden.

## Sicherheitsgruppen- und Subnetzkennungen

Eine Lambda-Funktion kann mit mehreren Sicherheitsgruppen und Subnetzen konfiguriert werden. Um eine Sicherheitsgruppe oder ein Subnetz für eine Lambda-Funktion zu konfigurieren, geben Sie einen Wert und einen Typ an.

- Wert — Ein Bezeichner für die Sicherheitsgruppe oder das Subnetz. Die akzeptierten Werte variieren je nach Typ.
- Typ — Die folgenden Wertetypen sind zulässig:
  - Parametername
  - AWS Systems Manager (SSM) Parameterspeicher
  - Statischer Wert

## Parametertyp

Der `Parameters` Abschnitt einer AWS CloudFormation Vorlage kann verwendet werden, um Ressourceninformationen in mehreren Vorlagen zu speichern. Weitere Informationen zu Parametern finden Sie unter [Parameter](#) im AWS CloudFormation Benutzerhandbuch.

Für den Parametertyp können Sie einen Parameternamen angeben. Im folgenden Beispiel geben wir einen `PrivateSubnet1` Parameternamen an:

**Subnet IDs**  
List of VPC subnet identifiers

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

Wenn Sie einen Parameternamen angeben, definiert Infrastructure Composer ihn im `Parameters` Abschnitt Ihrer Vorlage. Anschließend verweist Infrastructure Composer auf den Parameter in Ihrer Lambda-Funktionsressource. Im Folgenden wird ein Beispiel gezeigt:

```
...
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-Typ

Der SSM Parameter Store bietet einen sicheren, hierarchischen Speicher für die Verwaltung von Konfigurationsdaten und Geheimnissen. Weitere Informationen finden Sie unter [AWS Systems Manager -Parameterspeicher](#) im Benutzerhandbuch für AWS Systems Manager .

Für den SSM-Typ können Sie die folgenden Werte angeben:

- Dynamischer Verweis auf einen Wert aus dem SSM-Parameterspeicher.
- Logische ID einer in Ihrer Vorlage definierten `AWS::SSM::Parameter` Ressource.

### Dynamische Referenz

Sie können mithilfe einer dynamischen Referenz im folgenden Format auf einen Wert aus dem SSM-Parameterspeicher verweisen: `{{resolve:ssm:reference-key}}`. Weitere Informationen finden Sie unter [SSM-Parameter](#) im AWS CloudFormation Benutzerhandbuch.

Infrastructure Composer erstellt den Infrastrukturcode zur Konfiguration Ihrer Lambda-Funktion mit dem Wert aus dem SSM-Parameterspeicher. Im Folgenden wird ein Beispiel gezeigt:

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

### Logical ID

Sie können anhand der logischen ID auf eine `AWS::SSM::Parameter` Ressource in derselben Vorlage verweisen.

Im Folgenden finden Sie ein Beispiel für eine `AWS::SSM::Parameter` Ressource mit dem Namen `PrivateSubnet1Parameter`, die die Subnetz-ID für `PrivateSubnet1` speichert:

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

Im Folgenden finden Sie ein Beispiel dafür, wie dieser Ressourcenwert von der logischen ID für die Lambda-Funktion bereitgestellt wird:

### Subnet IDs

List of VPC subnet identifiers

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

Infrastructure Composer erstellt den Infrastrukturcode zur Konfiguration Ihrer Lambda-Funktion mit dem SSM-Parameter:

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

## Typ des statischen Werts

Wenn eine Sicherheitsgruppe oder ein Subnetz bereitgestellt wird AWS CloudFormation, wird ein ID-Wert erstellt. Sie können diese ID als statischen Wert angeben.

Für den statischen Wertetyp sind die folgenden Werte gültig:

- Geben Sie für Sicherheitsgruppen den `anGroupId`. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter [Rückgabewerte](#). Im Folgenden wird ein Beispiel gezeigt: `sg-0b61d5c742dc2c773`.
- Geben Sie für Subnetze den `SubnetId` an. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter [Rückgabewerte](#). Im Folgenden wird ein Beispiel gezeigt: `subnet-01234567890abcdef`.

Infrastructure Composer erstellt den Infrastrukturcode, um Ihre Lambda-Funktion mit dem statischen Wert zu konfigurieren. Im Folgenden wird ein Beispiel gezeigt:

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

## Verwenden mehrerer Typen

Für Sicherheitsgruppen und Subnetze können Sie mehrere Typen zusammen verwenden. Im Folgenden finden Sie ein Beispiel, das drei Sicherheitsgruppen für eine Lambda-Funktion konfiguriert, indem Werte verschiedener Typen bereitgestellt werden:

### 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 verweist auf alle drei Werte unter der SecurityGroupIds Eigenschaft:

```

...
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
```

## Lambda-Funktionen mit extern VPCs in Infrastructure Composer konfigurieren

Verwenden Sie die erweiterte Komponentenkarte Lambda Function, um mit der Konfiguration einer Lambda-Funktion mit einer VPC zu beginnen, die auf einer anderen Vorlage definiert ist. Diese Karte stellt eine Lambda-Funktion dar, die den `AWS::Serverless::Function` Ressourcentyp AWS Serverless Application Model (AWS SAM) verwendet.

So konfigurieren Sie eine Lambda-Funktion mit einer VPC aus einer externen Vorlage

1. Erweitern Sie im Bereich mit den Eigenschaften der Lambda-Funktion den Dropdownbereich VPC-Einstellungen (erweitert).
2. Wählen Sie Zu externer VPC zuweisen aus.
3. Geben Sie Werte für die Sicherheitsgruppen und Subnetze an, die für die Lambda-Funktion konfiguriert werden sollen. Details dazu finden Sie unter [Sicherheitsgruppen- und Subnetzbezeichnungen](#).
4. Speichern Sie Ihre Änderungen.

## Parameter in importierten Vorlagen für eine externe VPC mit Infrastructure Composer

Wenn Sie eine vorhandene Vorlage mit Parametern importieren, die für die Sicherheitsgruppen und Subnetze einer externen VPC definiert sind, bietet Infrastructure Composer eine Dropdownliste, aus der Sie Ihre Parameter auswählen können.

Im Folgenden finden Sie ein Beispiel für den `Parameters` Abschnitt einer importierten Vorlage:

```
...
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
...

```

Bei der Konfiguration einer externen VPC für eine neue Lambda-Funktion auf der Leinwand sind diese Parameter in einer Dropdownliste verfügbar. Im Folgenden wird ein Beispiel gezeigt:

### Subnet IDs

List of VPC subnet identifiers

Value	Type
<input style="width: 90%; border: none;" type="text" value="🔍  "/>	<div style="border: 1px solid #ccc; padding: 2px 5px; display: inline-block;">Parameter ▼</div>
VPCSubnets	
VPCSubnet	

## Einschränkungen beim Import von Listenparametertypen

Normalerweise können Sie für jede Lambda-Funktion mehrere Sicherheitsgruppen- und Subnetzkennungen angeben. Wenn Ihre vorhandene Vorlage Listenparametertypen wie `List<AWS::EC2::SecurityGroup::Id>` oder `List<AWS::EC2::Subnet::Id>` enthält, können Sie nur einen Bezeichner angeben.

Weitere Informationen zum Typ von Parameterlisten finden Sie im AWS CloudFormation Benutzerhandbuch unter [Unterstützte AWS spezifische Parametertypen](#).

Im Folgenden finden Sie ein Beispiel für eine Vorlage, die `VPCSecurityGroups` als Listenparametertyp definiert ist:

```

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

```

...

Wenn Sie in Infrastructure Composer den `VPCSecurityGroups` Wert als Sicherheitsgruppenkennung für eine Lambda-Funktion auswählen, wird die folgende Meldung angezeigt:

### Security group IDs

List of VPC security group identifiers

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

[Add new item](#)

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

Diese Einschränkung ist darauf zurückzuführen, dass die `SubnetIds` Eigenschaften `SecurityGroupIds` und eines `AWS::Lambda::Function VpcConfig` Objekts jeweils nur eine Liste von Zeichenkettenwerten akzeptieren. Da ein einzelner Listenparametertyp eine Liste von Zeichenfolgen enthält, kann er das einzige Objekt sein, das bereitgestellt wird, wenn er angegeben wird.

Im Folgenden finden Sie ein Beispiel für Listenparametertypen, wie sie in der Vorlage definiert werden, wenn sie mit einer Lambda-Funktion konfiguriert sind:

```

...
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

```

## Hinzufügen neuer Parameter zu importierten Vorlagen mit Infrastructure Composer

Wenn Sie eine vorhandene Vorlage mit definierten Parametern importieren, können Sie auch neue Parameter erstellen. Anstatt einen vorhandenen Parameter aus der Dropdownliste auszuwählen, geben Sie einen neuen Typ und Wert an. Im Folgenden finden Sie ein Beispiel, das einen neuen Parameter mit dem Namen `MySecurityGroup` erstellt:

### Security group IDs

List of VPC security group identifiers

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

Für alle neuen Werte, die Sie im Bereich Ressourceneigenschaften für die Lambda-Funktion angeben, definiert Infrastructure Composer sie in einer Liste unter den SubnetIds Eigenschaften SecurityGroupIds oder einer Lambda-Funktion. Im Folgenden wird ein Beispiel gezeigt:

```

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

```

```
- !Ref VPCSubnet
```

Wenn Sie aus einer externen Vorlage auf die logische ID eines Listenparametertyps verweisen möchten, empfehlen wir Ihnen, die Vorlagenansicht zu verwenden und Ihre Vorlage direkt zu ändern. Die logische ID eines Listenparametertyps sollte immer als einzelner Wert und als einziger Wert angegeben werden.

```
...
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
```

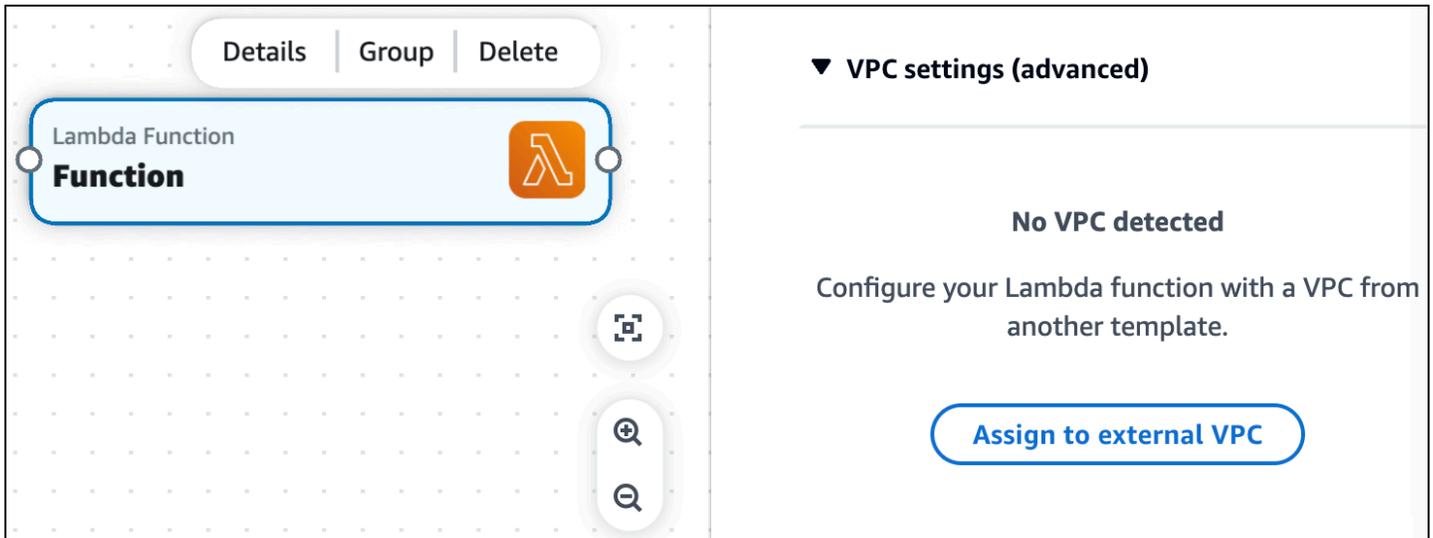
## Konfigurieren Sie eine Lambda-Funktion und eine VPC, die in einer anderen Vorlage definiert sind, mit Infrastructure Composer

In diesem Beispiel konfigurieren wir eine Lambda-Funktion in Infrastructure Composer mit einer VPC, die auf einer anderen Vorlage definiert ist.

Wir beginnen damit, eine mit Lambda Function erweiterte Komponentenkarte auf die Leinwand zu ziehen.



Als Nächstes öffnen wir den Bereich mit den Ressourceneigenschaften der Karte und erweitern den Drop-down-Bereich VPC-Einstellungen (erweitert).



Als Nächstes wählen wir Zu externer VPC zuweisen, um mit der Konfiguration einer VPC anhand einer externen Vorlage zu beginnen.

In diesem Beispiel verweisen wir auf eine Sicherheitsgruppen-ID und eine Subnetz-ID. Diese Werte werden erstellt, wenn die Vorlage, die die VPC definiert, bereitgestellt wird. Wir wählen den Typ Statischer Wert und geben den Wert unseres IDs ein. Wenn wir fertig sind, wählen wir Speichern.

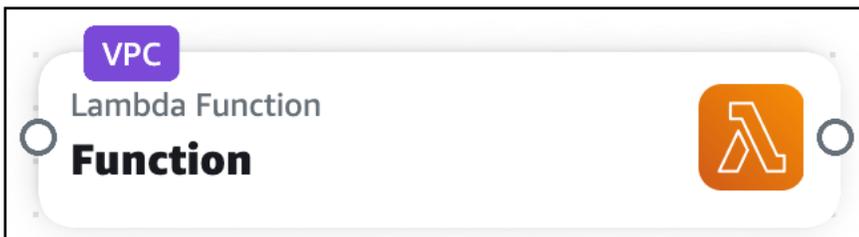
**Security group IDs**  
List of VPC security group identifiers

**Value**   **Type**  ▼

**Subnet IDs**  
List of VPC subnet identifiers

**Value**   **Type**  ▼

Nachdem unsere Lambda-Funktion mit unserer VPC konfiguriert ist, wird das VPC-Tag auf unserer Karte angezeigt.



Infrastructure Composer hat den Infrastrukturcode zur Konfiguration unserer Lambda-Funktion mit der Sicherheitsgruppe und dem Subnetz der externen VPC erstellt.

```

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}
```

# Stellen Sie Ihre serverlose Infrastructure Composer-Anwendung in der AWS Cloud bereit

Wird verwendet AWS-Infrastruktur-Composer , um einsatzbereite serverlose Anwendungen zu entwerfen. Verwenden Sie für die Bereitstellung einen beliebigen kompatiblen Dienst. AWS CloudFormation Wir empfehlen die Verwendung von [AWS Serverless Application Model \(AWS SAM\)](#).

AWS SAM ist ein Open-Source-Framework, das Entwicklertools zum Erstellen und Ausführen serverloser Anwendungen bereitstellt. AWS Mit AWS SAM der Kurzsyntax deklarieren Entwickler AWS CloudFormation Ressourcen und spezialisierte serverlose Ressourcen, die während der Bereitstellung in Infrastruktur umgewandelt werden.

## Wichtige Konzepte AWS SAM

Vor der Verwendung ist es wichtig AWS SAM, dass Sie sich mit einigen der grundlegenden Konzepte vertraut machen.

- [AWS SAM Funktionsweise](#): Dieses Thema, das sich im AWS Serverless Application Model Entwicklerhandbuch befindet, enthält wichtige Informationen zu den Hauptkomponenten, mit denen Sie Ihre Serverless-Anwendung erstellen: AWS SAM CLI, das AWS SAM Projekt und die AWS SAM Vorlage.
- [Verwendung von AWS Serverless Application Model \(AWS SAM\)](#): Dieses Thema, das sich im AWS Serverless Application Model Entwicklerhandbuch befindet, bietet einen allgemeinen Überblick über die Schritte, die Sie ausführen müssen, AWS SAM um Ihre Anwendung in der AWS Cloud bereitzustellen.

Beim Entwerfen Ihrer Anwendung in Infrastructure Composer können Sie den `sam sync` Befehl verwenden, um AWS SAM CLI erkennt automatisch lokale Änderungen und stellt diese Änderungen für bereit AWS CloudFormation. Weitere Informationen finden Sie unter [Sam Sync verwenden](#) im AWS Serverless Application Model Entwicklerhandbuch.

## Nächste Schritte

Informationen [Richten Sie sich für die Bereitstellung mit dem AWS SAM CLI und Infrastructure Composer](#) zur Vorbereitung der Bereitstellung Ihrer Anwendung finden Sie unter.

# Richten Sie sich für die Bereitstellung mit dem AWS SAM CLI und Infrastructure Composer

Um Ihre Anwendung mit bereitzustellen AWS SAM, müssen Sie zuerst das installieren und darauf zugreifen AWS CLI und das AWS SAM CLI. Die Themen in diesem Abschnitt enthalten Einzelheiten dazu.

## Installieren Sie das AWS CLI

Wir empfehlen die Installation und Einrichtung des AWS CLI vor der Installation des AWS SAM CLI. Anweisungen finden [Sie im AWS Command Line Interface Benutzerhandbuch unter Installation oder Aktualisierung AWS CLI auf die neueste Version von](#).

### Note

Nach der Installation des AWS CLI, müssen Sie AWS Anmeldeinformationen konfigurieren. Weitere Informationen finden Sie unter [Schnellinstallation](#) im AWS Command Line Interface Benutzerhandbuch.

## Installieren Sie den AWS SAM CLI

Um das zu installieren AWS SAM CLI, siehe [Installation des AWS SAM CLI](#) im Entwicklerhandbuch für AWS Serverless Application Model .

## Greifen Sie auf AWS SAM CLI

Wenn Sie Infrastructure Composer von verwenden AWS Management Console, haben Sie die folgenden Optionen zur Verwendung von AWS SAM CLI.

### Aktivieren Sie den lokalen Synchronisierungsmodus

Im lokalen Synchronisierungsmodus wird Ihr Projektordner, einschließlich der AWS SAM Vorlage, automatisch auf Ihrem lokalen Computer gespeichert. Infrastructure Composer strukturiert Ihr Projektverzeichnis so, dass es AWS SAM erkannt wird. Sie können das ausführen AWS SAM CLI aus dem Stammverzeichnis Ihres Projekts.

Weitere Hinweise zum lokalen Synchronisierungsmodus finden Sie unter [Synchronisieren und speichern Sie Ihr Projekt lokal in der Infrastructure Composer-Konsole](#).

## Exportieren Sie Ihre Vorlage

Sie können Ihre Vorlage auf Ihren lokalen Computer exportieren. Führen Sie dann den AWS SAM CLI aus dem übergeordneten Ordner, der die Vorlage enthält. Sie können die `--template-file` Option auch mit jedem verwenden AWS SAM CLI Befehl und geben Sie den Pfad zu Ihrer Vorlage an.

Verwenden Sie Infrastructure Composer aus dem AWS Toolkit for Visual Studio Code

Sie können Infrastructure Composer aus dem Toolkit for VS Code verwenden, um Infrastructure Composer auf Ihren lokalen Computer zu bringen. Verwenden Sie dann Infrastructure Composer und AWS SAM CLI von VS Code.

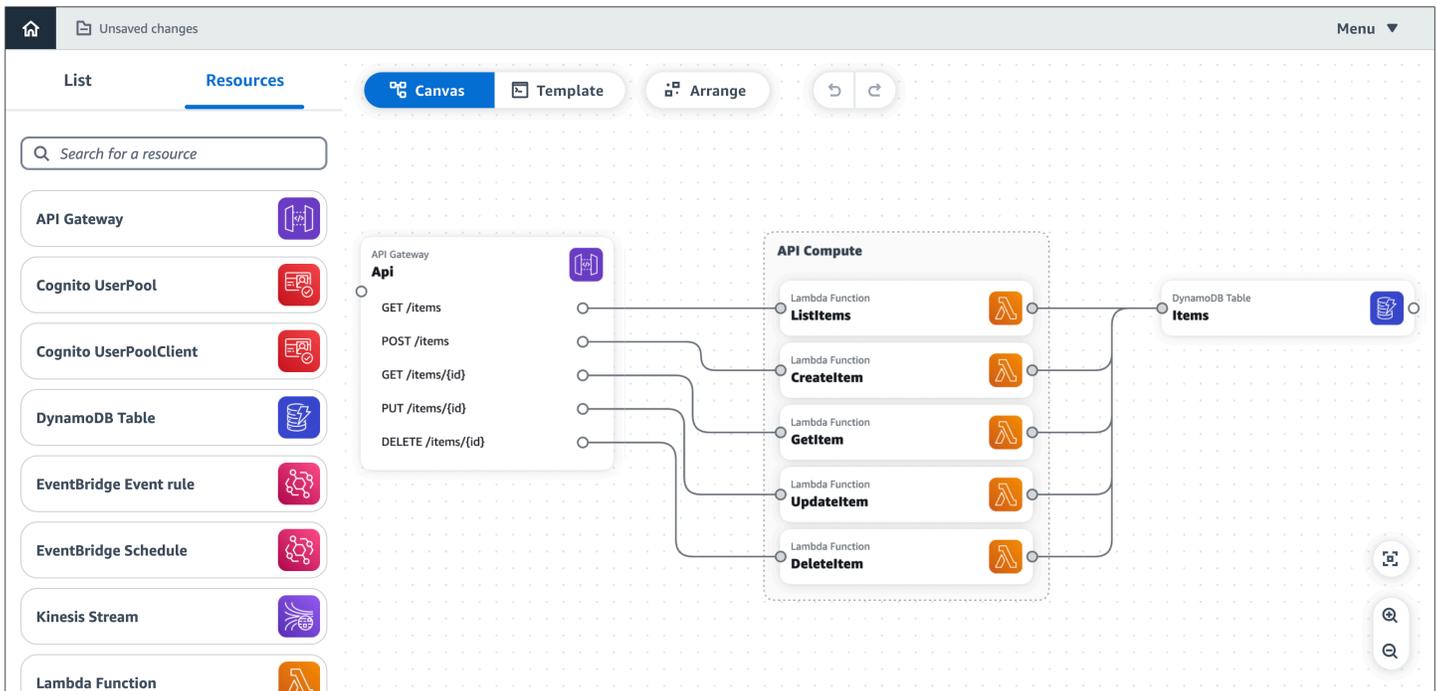
## Nächste Schritte

Informationen zur Bereitstellung Ihrer Anwendung finden Sie unter [Verwenden Sie Infrastructure Composer mit AWS SAM , um sie zu erstellen und bereitzustellen](#).

## Verwenden Sie Infrastructure Composer mit AWS SAM , um sie zu erstellen und bereitzustellen

Nachdem Sie den Vorgang abgeschlossen haben [Richten Sie sich für die Bereitstellung mit dem AWS SAM CLI und Infrastructure Composer](#), können Sie Ihre Anwendung mit einem AWS SAM Infrastructure Composer bereitstellen. Dieser Abschnitt enthält ein Beispiel, in dem detailliert beschrieben wird, wie Sie dies tun können. Anweisungen zur [Bereitstellung Ihrer Anwendung mit finden Sie auch unter Deploy your application and resources](#) with AWS SAM AWS SAM im AWS Serverless Application Model Developer Guide.

Dieses Beispiel zeigt Ihnen, wie Sie die Infrastructure Composer-Demoanwendung erstellen und bereitstellen. Die Demo-Anwendung verfügt über die folgenden Ressourcen:



### Note

- Weitere Informationen zur Demo-Anwendung finden Sie unter. [Laden und ändern Sie das Infrastructure Composer-Demoprojekt](#)
- In diesem Beispiel verwenden wir Infrastructure Composer mit aktivierter lokaler Synchronisierung.

1. Verwenden Sie den `sam build` Befehl, um die Anwendung zu erstellen.

```
$ 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
```

Der AWS SAM CLI erstellt das `./aws-sam` Verzeichnis im Projektordner. Dieses Verzeichnis enthält Build-Artefakte für die Lambda-Funktionen der Anwendung. Hier ist eine Ausgabe des Projektverzeichnis:

```
.
### 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. Jetzt ist die Anwendung bereit, bereitgestellt zu werden. Wir werden verwenden `sam deploy --guided`. Dadurch wird Ihre Anwendung anhand einer Reihe von Eingabeaufforderungen für die Bereitstellung vorbereitet.

```
$ 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]:

```

Das AWS SAM CLI zeigt eine Zusammenfassung dessen, was bereitgestellt wird:

```

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

```

Der AWS SAM CLI stellt die Anwendung bereit, indem zunächst ein AWS CloudFormation Changeset erstellt wird:

```

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

```

Dann das AWS SAM CLI stellt die Anwendung bereit:

```

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
	ListItemsRole		
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-
-----

```

Schließlich wird eine Meldung angezeigt, die Sie darüber informiert, dass die Bereitstellung erfolgreich war:

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

## Verwenden Sie Infrastructure Composer mit AWS SAM , um einen Stack zu löschen

Dieses Beispiel zeigt Ihnen, wie Sie einen AWS CloudFormation Stack mit dem `sam delete` Befehl löschen.

Geben Sie den Befehl `sam delete` in den AWS SAM CLI und bestätigen Sie, ob Sie den Stapel und die Vorlage löschen möchten:

```

$ 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-Infrastruktur-Composer Problembekämpfung

Die Themen in diesem Abschnitt enthalten Anleitungen zur Behebung von Fehlermeldungen bei der Verwendung von AWS-Infrastruktur-Composer.

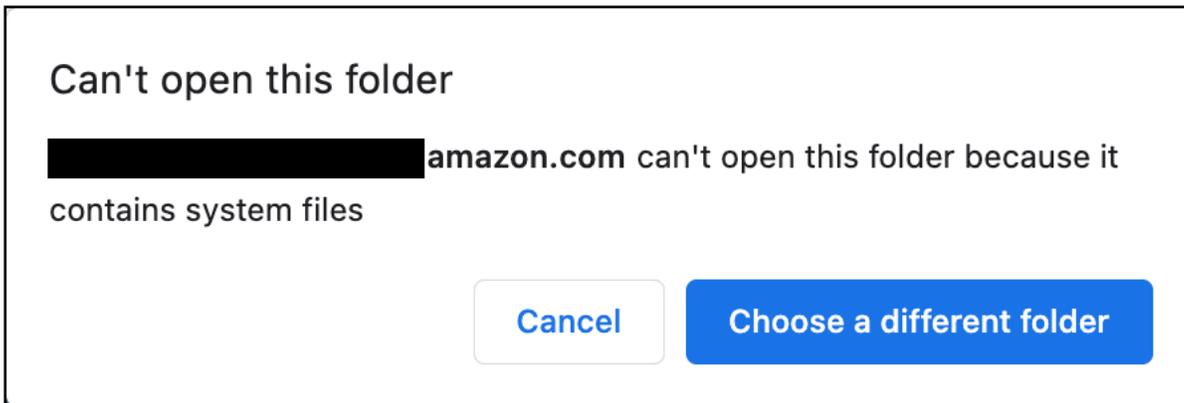
Themen

- [Fehlermeldungen](#)

## Fehlermeldungen

„Dieser Ordner kann nicht geöffnet werden“

Fehlerbeispiel:



Mögliche Ursache: Infrastructure Composer kann im lokalen Synchronisierungsmodus nicht auf ein sensibles Verzeichnis zugreifen.

Weitere Informationen zu diesem Fehler finden Sie unter [Data Infrastructure Composer erhält Zugriff auf](#).

Versuchen Sie, eine Verbindung zu einem anderen lokalen Verzeichnis herzustellen oder Infrastructure Composer zu verwenden, während die lokale Synchronisierung deaktiviert ist.

## „Inkompatible Vorlage“

Beispielfehler: Beim Laden eines neuen Projekts in Infrastructure Composer wird Folgendes angezeigt:

Mögliche Ursache: Ihr Projekt enthält eine extern referenzierte Datei, die in Infrastructure Composer nicht unterstützt wird.

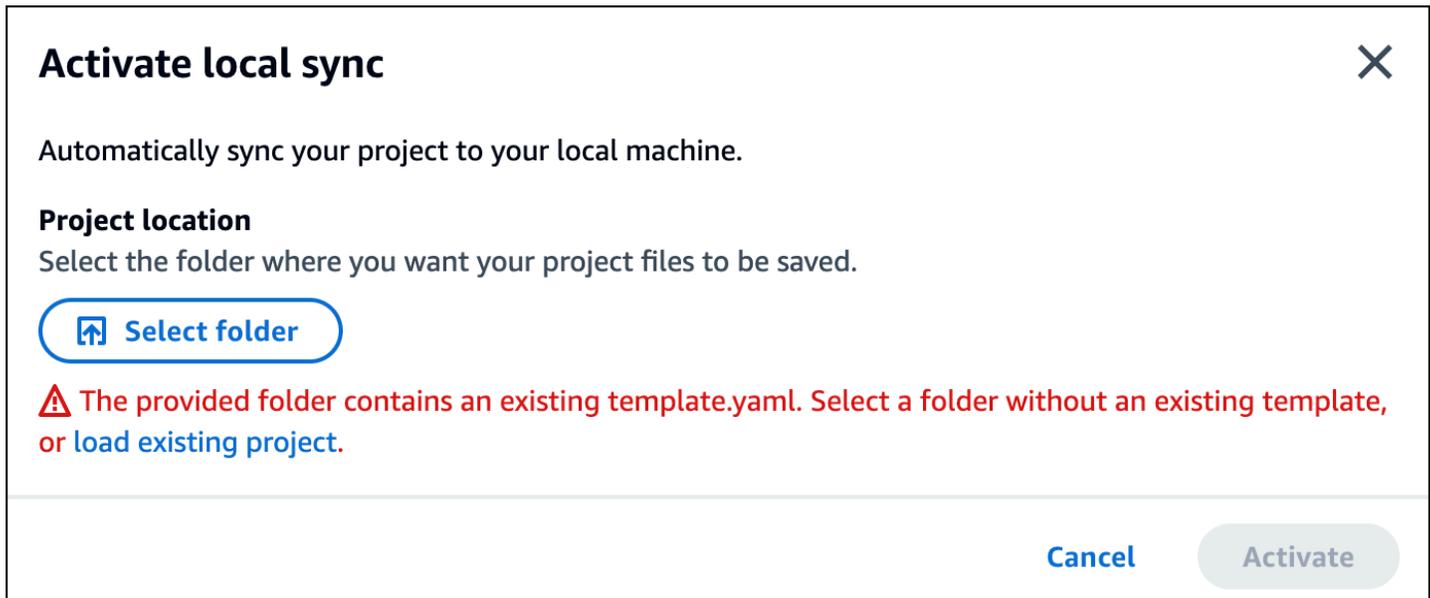
Weitere Informationen zu unterstützten externen Dateien in Infrastructure Composer finden Sie unter [Verweisen Sie auf externe Dateien](#).

Mögliche Ursache: Ihr Projekt ist mit einer externen Datei in einem anderen lokalen Verzeichnis verknüpft.

Verschieben Sie Ihre extern referenzierte Datei in ein Unterverzeichnis des Verzeichnisses, das Sie für den lokalen Synchronisierungsmodus von Infrastructure Composer ausgewählt haben.

„Der bereitgestellte Ordner enthält eine vorhandene Datei `template.yaml`“

Beim Versuch, die lokale Synchronisierung zu aktivieren, wird der folgende Fehler angezeigt:



Mögliche Ursache: Ihr ausgewählter Ordner enthält bereits eine `template.yaml`-Datei.

Wählen Sie ein anderes Verzeichnis aus, das keine Anwendungsvorlage enthält, oder erstellen Sie ein neues Verzeichnis.

„Ihr Browser ist nicht berechtigt, Ihr Projekt in diesem Ordner zu speichern...“

Mögliche Ursache: Infrastructure Composer kann im lokalen Synchronisierungsmodus nicht auf ein sensibles Verzeichnis zugreifen.

Weitere Informationen zu diesem Fehler finden Sie unter [Data Infrastructure Composer erhält Zugriff auf](#).

Versuchen Sie, eine Verbindung zu einem anderen lokalen Verzeichnis herzustellen, oder verwenden Sie Infrastructure Composer mit deaktivierter lokaler Synchronisierung.

# Sicherheit in AWS-Infrastruktur-Composer

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#). Weitere Informationen zu den Compliance-Programmen, die für gelten AWS-Infrastruktur-Composer, finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#).
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Infrastructure Composer anwenden können. In den folgenden Themen erfahren Sie, wie Sie Infrastructure Composer konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere AWS Dienste verwenden können, die Sie bei der Überwachung und Sicherung Ihrer Infrastructure Composer-Ressourcen unterstützen.

Themen

- [Datenschutz in AWS-Infrastruktur-Composer](#)
- [AWS Identity and Access Management für AWS-Infrastruktur-Composer](#)
- [Konformitätsvalidierung für AWS-Infrastruktur-Composer](#)
- [Resilienz in AWS-Infrastruktur-Composer](#)

## Datenschutz in AWS-Infrastruktur-Composer

Das [Modell der AWS gemeinsamen Verantwortung](#) und gilt für den Datenschutz in AWS Infrastructure Composer. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz

der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden zu schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS benötigt TLS 1.2 und empfiehlt TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Infrastructure Composer oder anderen Geräten arbeiten und die Konsole, die API oder AWS-Services verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

**Note**

Alle Daten, die Sie in Infrastructure Composer eingeben, werden ausschließlich zu dem Zweck verwendet, Funktionen innerhalb von Infrastructure Composer bereitzustellen und Projektdateien und Verzeichnisse zu generieren, die lokal auf Ihrem Computer gespeichert werden. Infrastructure Composer speichert, speichert oder überträgt keine dieser Daten.

## Datenverschlüsselung

Infrastructure Composer verschlüsselt keine Kundeninhalte, da Daten nicht gespeichert, gespeichert oder übertragen werden.

### Verschlüsselung im Ruhezustand

Infrastructure Composer verschlüsselt keine Kundeninhalte, da Daten nicht gespeichert, gespeichert oder übertragen werden.

### Verschlüsselung während der Übertragung

Infrastructure Composer verschlüsselt keine Kundeninhalte, da Daten nicht gespeichert, gespeichert oder übertragen werden.

## Schlüsselverwaltung

Infrastructure Composer unterstützt keine Schlüsselverwaltung, da Kundeninhalte nicht gespeichert, gespeichert oder übertragen werden.

## Datenschutz für den Datenverkehr zwischen Netzwerken

Infrastructure Composer generiert keinen Datenverkehr mit lokalen Clients und Anwendungen.

## AWS Identity and Access Management für AWS-Infrastruktur-Composer

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Infrastructure

Composer-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

## Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Wie AWS-Infrastruktur-Composer funktioniert mit IAM](#)

## Zielgruppe

Infrastructure Composer erfordert mindestens schreibgeschützten Zugriff auf die AWS Management Console. Jeder Benutzer mit dieser Autorisierung kann alle Funktionen von Infrastructure Composer nutzen. Granularer Zugriff auf bestimmte Funktionen von Infrastructure Composer wird nicht unterstützt.

## Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung

von Anforderungen finden Sie unter [AWS Signature Version 4 für API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [AWS Multi-Faktor-Authentifizierung \(MFA\) in IAM](#) im IAM-Benutzerhandbuch.

## AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

## Verbundidentität

Es hat sich bewährt, menschlichen Benutzern, einschließlich Benutzern, die Administratorzugriff benötigen, vorzuschreiben, den Verbund mit einem Identitätsanbieter zu verwenden, um AWS-Services mithilfe temporärer Anmeldeinformationen darauf zuzugreifen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

## IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe einen Namen geben IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

## IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Methoden für die Übernahme einer Rolle](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe](#)

[Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- Servicerolle – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und

gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt](#) werden.

## Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

## Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer Inline-Richtlinie wählen, finden Sie unter [Auswählen zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

## Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

## Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF  
Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter [Richtlinien zur Servicesteuerung](#) im AWS Organizations Benutzerhandbuch.
- **Ressourcenkontrollrichtlinien (RCPs)** — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter [Resource Control Policies \(RCPs\)](#) im AWS Organizations Benutzerhandbuch.

- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

## Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

## Wie AWS-Infrastruktur-Composer funktioniert mit IAM

AWS-Infrastruktur-Composer erfordert mindestens schreibgeschützten Zugriff auf die AWS Management Console. Jeder Benutzer mit dieser Autorisierung kann alle Funktionen von Infrastructure Composer nutzen. Granularer Zugriff auf bestimmte Funktionen von Infrastructure Composer wird nicht unterstützt.

Wenn Sie Ihre Projektvorlage und die Dateien für bereitstellen AWS CloudFormation, müssen Sie über die erforderlichen Berechtigungen verfügen. Weitere Informationen finden Sie unter [Steuern des Zugriffs mit AWS Identity and Access Management](#) im AWS CloudFormation Benutzerhandbuch.

Die folgende Tabelle zeigt, mit AWS-Infrastruktur-Composer welchen IAM-Funktionen Sie arbeiten können.

IAM-Feature	Unterstützung für Infrastructure Composer
<a href="#">Identitätsbasierte Richtlinien</a>	Nein
<a href="#">Ressourcenbasierte Richtlinien</a>	Nein
<a href="#">Richtlinienaktionen</a>	Nein

IAM-Feature	Unterstützung für Infrastructure Composer
<a href="#">Richtlinienressourcen</a>	Nein
<a href="#">Bedingungsschlüssel für die Richtlinie</a>	Nein
<a href="#">ACLs</a>	Nein
<a href="#">ABAC (Tags in Richtlinien)</a>	Nein
<a href="#">Temporäre Anmeldeinformationen</a>	Ja
<a href="#">Hauptberechtigungen</a>	Nein
<a href="#">Servicerollen</a>	Nein
<a href="#">Serviceverknüpfte Rollen</a>	Nein

Einen allgemeinen Überblick darüber, wie Infrastructure Composer und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

## Identitätsbasierte Richtlinien für Infrastructure Composer

Unterstützt identitätsbasierte Richtlinien: Nein

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

## Ressourcenbasierte Richtlinien in Infrastructure Composer

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

## Richtlinienaktionen für Infrastructure Composer

Unterstützt politische Aktionen: Nein

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Infrastructure Composer-Aktionen finden Sie unter [Von AWS Infrastructure Composer definierte Aktionen](#) in der Serviceautorisierungsreferenz.

## Richtlinienressourcen für Infrastructure Composer

Unterstützt politische Ressourcen: Nein

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (\*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der Infrastructure Composer-Ressourcentypen und ihrer Eigenschaften ARNs finden Sie unter [Von AWS Infrastructure Composer definierte Ressourcen](#) in der Serviceautorisierungsreferenz. Informationen zu den Aktionen, mit denen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von AWS Infrastructure Composer definierte Aktionen](#).

## Bedingungsschlüssel für Richtlinien für Infrastructure Composer

Unterstützt dienstspezifische Richtlinien-Bedingungsschlüssel: Nein

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Infrastructure Composer-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für AWS Infrastructure Composer](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von AWS Infrastructure Composer definierte Aktionen](#).

## ACLs in Infrastructure Composer

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

## ABAC mit Infrastructure Composer

Unterstützt ABAC (Tags in Richtlinien): Nein

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das

Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

## Verwenden temporärer Anmeldeinformationen mit Infrastructure Composer

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln von einer Benutzerrolle zu einer IAM-Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt

langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

Sie können temporäre Anmeldeinformationen verwenden, um über den auf Infrastructure Composer zuzugreifen AWS Management Console. Ein Beispiel finden Sie im IAM-Benutzerhandbuch unter [Aktivieren des benutzerdefinierten Identity-Broker-Zugriffs auf die AWS Konsole](#).

## Serviceübergreifende Prinzipalberechtigungen für Infrastructure Composer

Unterstützt Forward Access Sessions (FAS): Nein

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

## Servicerollen für Infrastructure Composer

Unterstützt Servicerollen: Nein

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

### Warning

Durch das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von Infrastructure Composer beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn Infrastructure Composer Sie dazu anleitet.

## Dienstbezogene Rollen für Infrastructure Composer

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

## Konformitätsvalidierung für AWS-Infrastruktur-Composer

Informationen darüber, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt](#), finden Sie unter [Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Compliance und Governance im Bereich Sicherheit](#) – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- [Referenz für berechnete HIPAA-Services](#) – Listet berechnete HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National

Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.

- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Dies AWS-Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerelementreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

## Resilienz in AWS-Infrastruktur-Composer

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter [AWS Globale](#) Infrastruktur.

Alle Daten, die Sie in Infrastructure Composer eingeben, werden ausschließlich zu dem Zweck verwendet, Funktionen innerhalb von Infrastructure Composer bereitzustellen und Projektdateien und Verzeichnisse zu generieren, die lokal auf Ihrem Computer gespeichert werden. Infrastructure Composer speichert oder speichert keine dieser Daten.

# Dokumentenverlauf für Infrastructure Composer

In der folgenden Tabelle werden wichtige Dokumentationsversionen für Infrastructure Composer beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

- Letzte Aktualisierung der Dokumentation: 30. November 2023

Änderung	Beschreibung	Datum
<a href="#">Neustrukturierter und aktualisierter Inhalt im gesamten Entwicklerhandbuch</a>	Der Leitfaden wurde neu organisiert und neu strukturiert, um die Auffindbarkeit und Benutzerfreundlichkeit zu verbessern. Aktualisierte und verbesserte Titel. Bei der Einführung von Themen und Konzepten wurden zusätzliche Details bereitgestellt.	1. August 2024
<a href="#">Dokumentation für die Verwendung von Infrastructure Composer im CloudFormation Konsolenmodus hinzugefügt und der Infrastructure Composer Developer Guide neu strukturiert.</a>	AWS-Infrastruktur-Composer kann jetzt im AWS CloudFormation Konsolenmodus verwendet werden. Weitere Informationen finden Sie unter <a href="#">Verwenden von Infrastructure Composer im CloudFormation Konsolenmodus</a> . Darüber hinaus wurde ein Großteil des Inhalts des Benutzerhandbuchs neu organisiert, um ein optimiertes Benutzererlebnis zu bieten.	28. März 2024
<a href="#">Dokumentation für die Infrastructure Composer-Integration</a>	AWS-Infrastruktur-Composer aus dem Toolkit for VS Code	30. November 2023

[mit hinzugefügt CodeWhisperer](#)

bietet eine Integration mit Amazon CodeWhisperer. Weitere Informationen finden Sie unter [AWS-Infrastruktur-Composer Mit Amazon verwenden CodeWhisperer](#).

[Es wurde eine Dokumentation für die Bereitstellung Ihrer Anwendung mit Infrastructure Composer aus dem hinzugefügt AWS Toolkit for Visual Studio Code](#)

Verwenden Sie die Synchronisierungsschaltfläche auf der Infrastructure Composer-Arbeitsfläche, um Ihre Anwendung auf dem bereitzustellen AWS Cloud. Weitere Informationen finden Sie unter [Bereitstellen Ihrer Anwendung mit Sam Sync](#).

30. November 2023

[Es wurde eine Dokumentation für Infrastructure Composer aus dem hinzugefügt AWS Toolkit for Visual Studio Code](#)

Sie können jetzt Infrastructure Composer von VS Code mit dem verwenden AWS Toolkit for Visual Studio Code. Weitere Informationen finden Sie unter [Verwenden AWS-Infrastruktur-Composer von AWS Toolkit for Visual Studio Code](#).

30. November 2023

[Integration von Step Functions Workflow Studio hinzugefügt](#)

Starten Sie Step Functions Workflow Studio von der Infrastructure Composer-Arbeitsfläche aus. Weitere Informationen finden Sie unter [Verwenden AWS-Infrastruktur-Composer mit AWS Step Functions](#).

8. November 2023

[Lambda-Konsole und Infrastructure Composer-Integration hinzugefügt](#)

Starten Sie die Infrastructure Composer-Canvas von der Lambda-Konsole aus. Weitere Informationen finden Sie unter [AWS-Infrastruktur-Composer Mit der AWS Lambda Konsole verwenden](#).

14. November 2023

[Amazon VPC wurde als besonderer Service mit Infrastructure Composer hinzugefügt](#)

Infrastructure Composer führt ein VPC-Tag ein, um Ressourcen zu visualisieren, die mit einer VPC konfiguriert wurden. Sie können Lambda-Funktionen auch konfigurieren, wenn sie auf einer externen Vorlage VPCs definiert sind. Weitere Informationen finden Sie unter [Verwenden von Infrastructure Composer mit Amazon VPC](#).

17. Oktober 2023

[Amazon RDS wurde als besonderer Service mit Infrastructure Composer hinzugefügt](#)

Connect Ihre Infrastructure Composer-Anwendung mit einem Amazon RDS-DB-Cluster oder einer Amazon RDS-Instance, die auf einer externen Vorlage definiert ist. Weitere Informationen finden Sie unter [Verwenden von Infrastructure Composer mit Amazon RDS](#).

17. Oktober 2023

[Infrastructure Composer-Unterstützung wurde hinzugefügt, um mit allen AWS CloudFormation Ressourcen zu entwerfen](#)

Wählen Sie eine beliebige AWS CloudFormation Ressource aus der Resource palette aus, mit der Sie Ihre Anwendungen entwerfen möchten. Weitere Informationen finden Sie unter [Arbeiten mit beliebigen AWS CloudFormation Ressourcen](#).

26. September 2023

[Dokumentation für Karten in Infrastructure Composer hinzugefügt](#)

Infrastructure Composer unterstützt mehrere Kartentypen, mit denen Sie Ihre Anwendung entwerfen und erstellen können. Weitere Informationen finden Sie unter [Entwerfen mit Karten in Infrastructure Composer](#).

20. September 2023

[Dokumentation für die Funktionen zum Rückgängigmachen und Wiederherstellen hinzugefügt](#)

Verwenden Sie die Schaltflächen „Rückgängig“ und „Wiederherstellen“ auf der Infrastructure Composer-Arbeitsfläche. Weitere Informationen finden Sie unter [Rückgängig machen und Wiederherstellen](#).

1. August 2023

[Dokumentation für den lokalen Synchronisierungsmodus hinzugefügt](#)

Verwenden Sie den lokalen Synchronisierungsmodus, um Ihr Projekt automatisch zu synchronisieren und auf Ihrem lokalen Computer zu speichern. Weitere Informationen finden Sie unter [Lokaler Synchronisierungsmodus](#).

1. August 2023

[Dokumentation für die Export-Canvas-Funktion hinzugefügt](#)

Verwenden Sie die Export-Canvas-Funktion, um die Leinwand Ihrer Anwendung als Bild auf Ihren lokalen Computer zu exportieren. Weitere Informationen finden Sie unter [Leinwand exportieren](#).

1. August 2023

[Infrastructure Composer-Unterstützung für externe Dateiverweise](#)

Verweisen Sie auf externe Dateien für unterstützte Ressourcen in Infrastructure Composer. Weitere Informationen finden Sie unter [Arbeiten mit Vorlagen, die auf externe Dateien verweisen](#).

17. Mai 2023

[Neue Dokumentation zum Verbinden von Ressourcen](#)

Connect Ressourcen miteinander, um ereignisgesteuerte Beziehungen zwischen Ressourcen in Ihrer Anwendung zu definieren. Weitere Informationen finden Sie unter [Ressourcen mithilfe der Visual Canvas von Infrastructure Composer miteinander verbinden](#).

7. März 2023

[Neue Change Inspector-Funktion](#)

Verwenden Sie den Change Inspector, um Ihre Vorlagencode-Aktualisierungen anzusehen und zu erfahren, was Infrastructure Composer für Sie erstellt. Weitere Informationen finden Sie unter [Codeaktualisierungen mit dem Change Inspector anzeigen](#).

7. März 2023

### [Infrastructure Composer ist jetzt allgemein verfügbar](#)

AWS-Infrastruktur-Composer ist jetzt allgemein verfügbar. Weitere Informationen finden Sie unter [AWS-Infrastruktur-Composer Jetzt allgemein verfügbar — Schnelle visuelle Erstellung serverloser Anwendungen](#).

7. März 2023

### [Die Vorteile des vernetzten Modus wurden erweitert](#)

Verwenden Sie Infrastructure Composer im verbundenen Modus mit Ihrer lokalen IDE, um die Entwicklung zu beschleunigen. Weitere Informationen finden Sie unter [Verwenden von Infrastructure Composer mit Ihrer lokalen IDE](#).

7. März 2023

### [Das Thema zur Verwendung anderer AWS Dienste zur Bereitstellung Ihrer Anwendung wurde aktualisiert](#)

Verwenden Sie Infrastructure Composer, um einsatzbereite serverlose Anwendungen zu entwerfen. Verwenden Sie diese Option AWS SAM , um Ihre serverlose Anwendung bereitzustellen. Weitere Informationen finden Sie unter [Verwenden von Infrastructure Composer mit AWS CloudFormation und AWS SAM](#).

03. März 2023

[Abschnitt „Serverlose Konzepte“ hinzugefügt](#)

Informieren Sie sich über grundlegende Konzepte für serverlose Systeme, bevor Sie Infrastructure Composer verwenden . Weitere Informationen finden Sie unter [Serverlose Konzepte](#).

2. März 2023

[Öffentliche Veröffentlichung](#)

Erste öffentliche Version von Infrastructure Composer.

01. Dezember 2022

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.