開發人員指南

AWS Infrastructure Composer

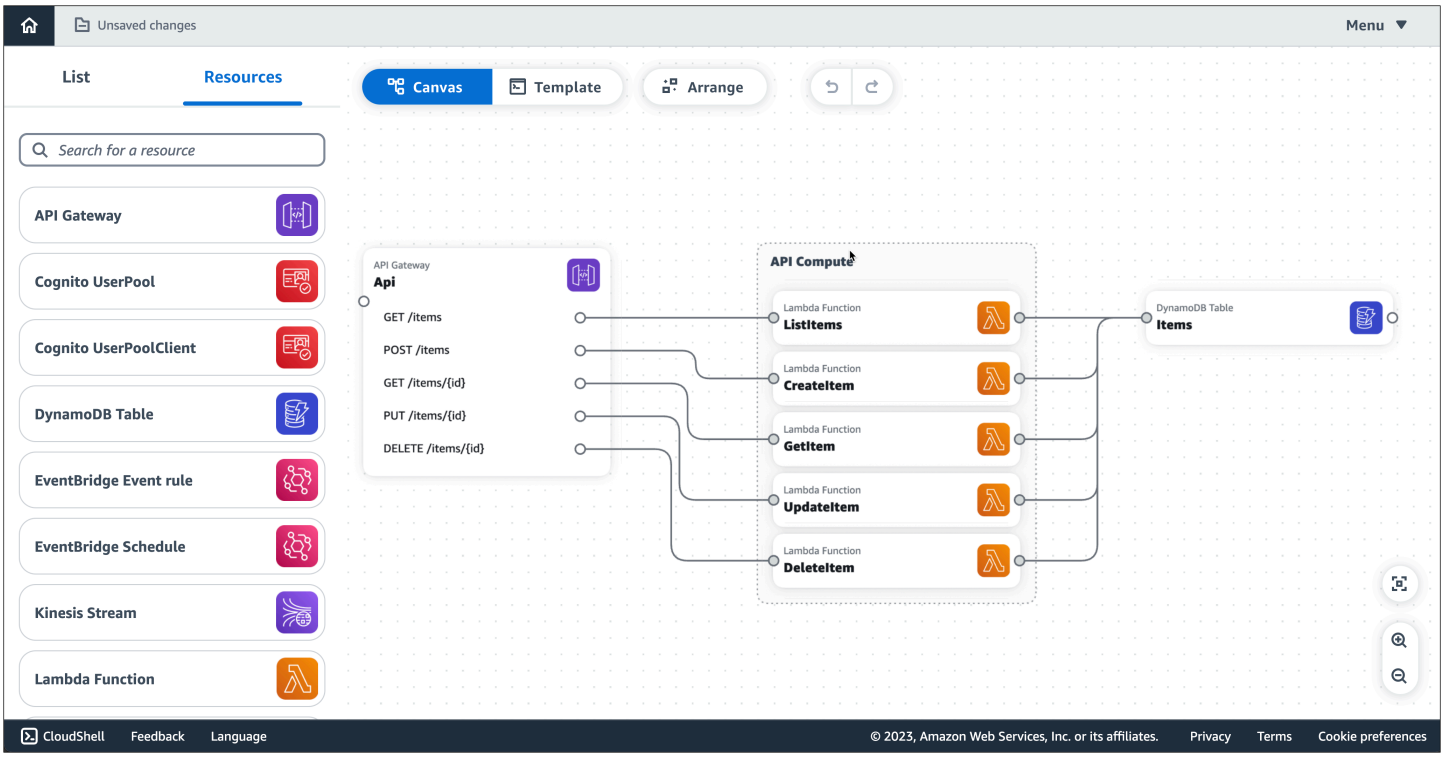
AWS Infrastructure Composer: 開發人員指南

Table of Contents

什麼是 AWS Infrastructure Composer？

AWS Infrastructure Composer 可讓您以視覺化方式編寫現代應用程式 AWS。更具體地說，您可以使用 Infrastructure Composer 從 支援的所有 AWS 服務視覺化、建置和部署現代應用程式， AWS CloudFormation 而不需要成為 的專家 AWS CloudFormation。

當您編寫 AWS CloudFormation 基礎設施時，透過令人滿意的drag-and-drop界面，基礎設施編譯器會將基礎設施建立為程式碼 (IaC) 範本，同時遵循 AWS 最佳實務。下圖顯示在 Infrastructure Composer 的視覺化畫布上拖曳、捨棄、設定和連線資源有多容易。



Infrastructure Composer 可以從 Infrastructure Composer 主控台、 AWS Toolkit for Visual Studio Code和 CloudFormation 主控台模式中使用。

主題

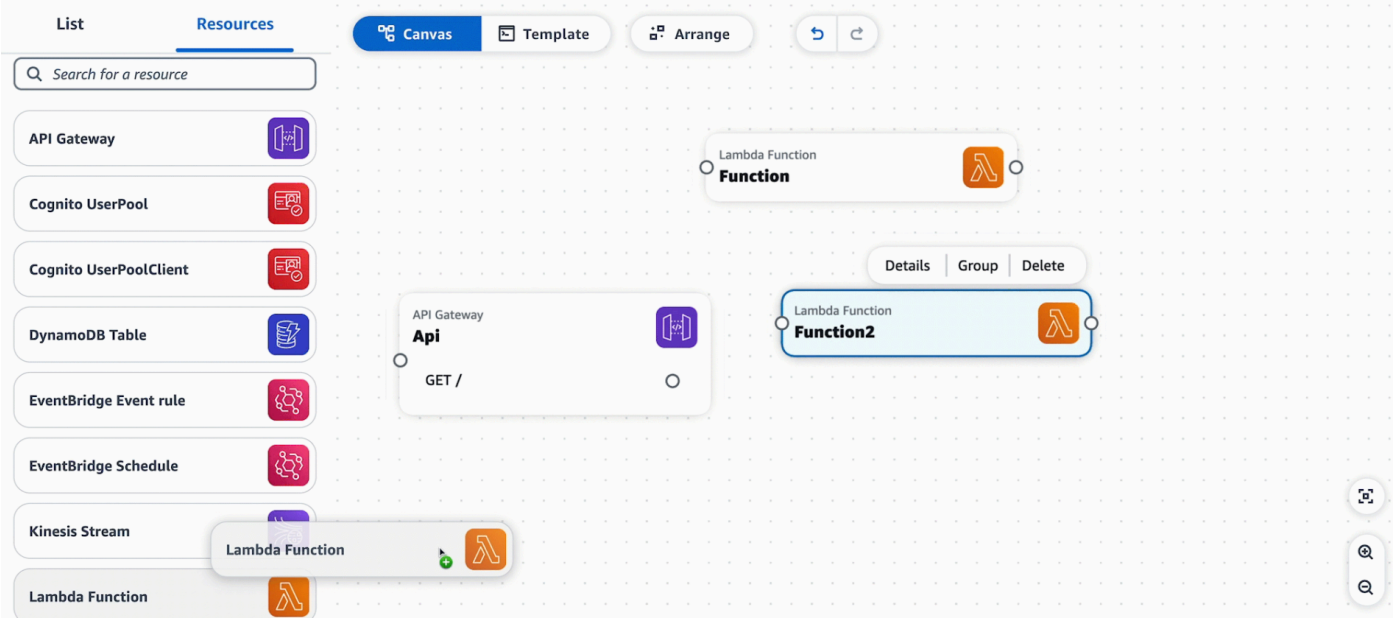
- 編寫您的應用程式架構
- 將基礎設施定義為程式碼 (IaC) 範本
- 與現有的工作流程整合
- 存取 Infrastructure Composer 的方法
- 進一步了解

- 後續步驟
- 的無伺服器概念 AWS Infrastructure Composer

編寫您的應用程式架構

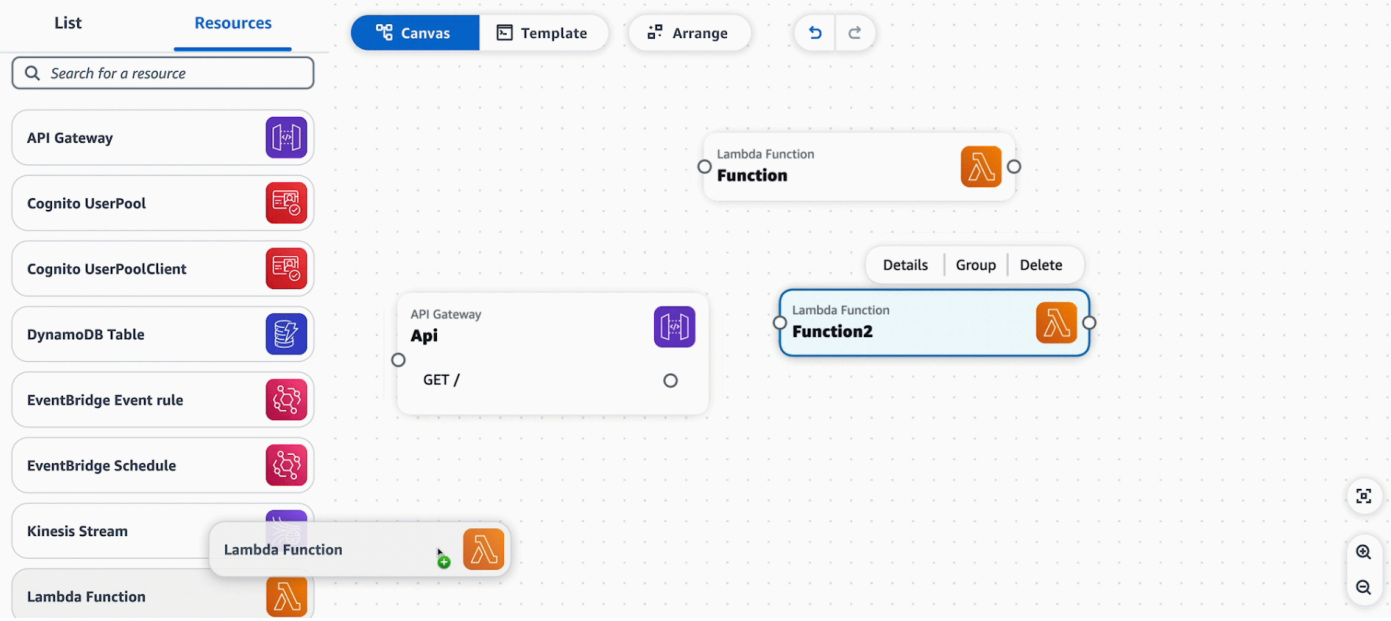
使用卡片建置

在 Infrastructure Composer 畫布上放置卡片，以視覺化和建置您的應用程式架構。



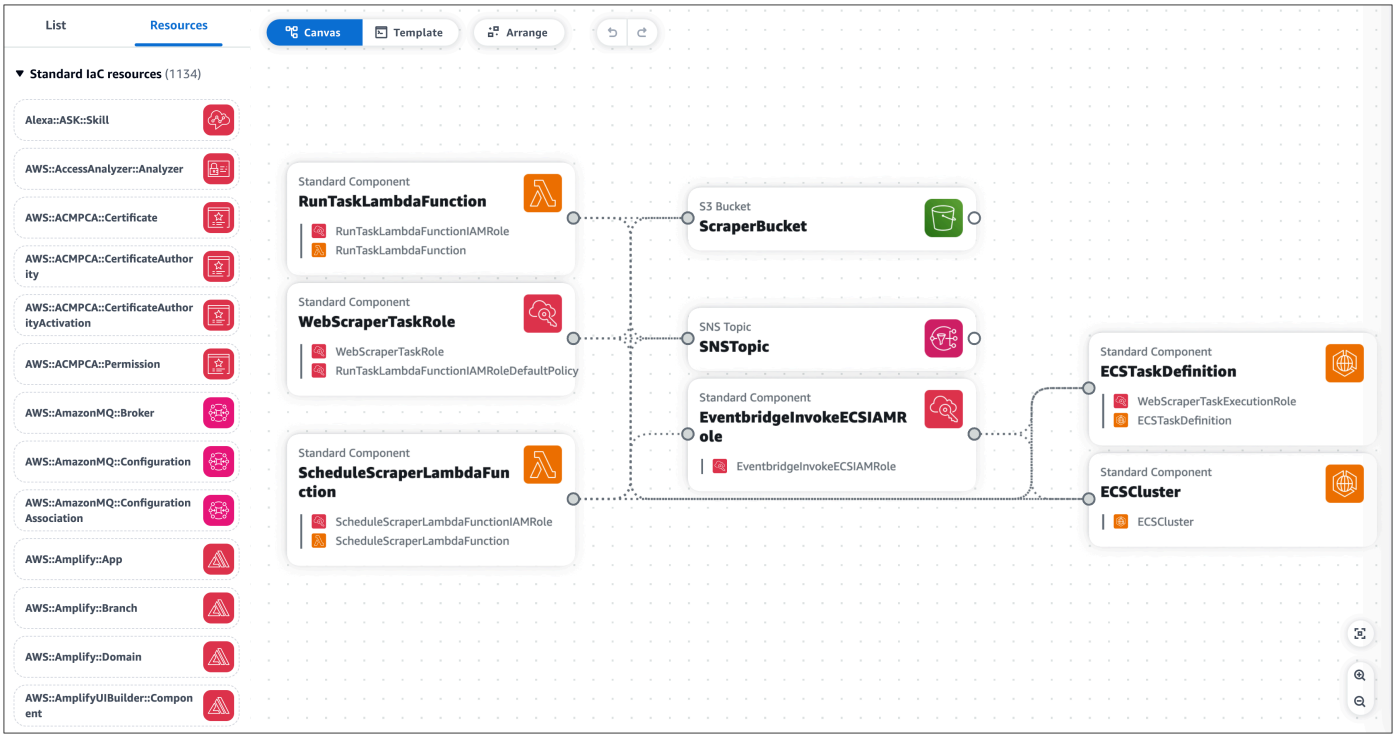
將卡片連接在一起

以視覺化方式將資源連接在一起，藉此設定資源彼此互動的方式。透過精選屬性面板進一步指定其屬性。



使用任何 AWS CloudFormation 資源

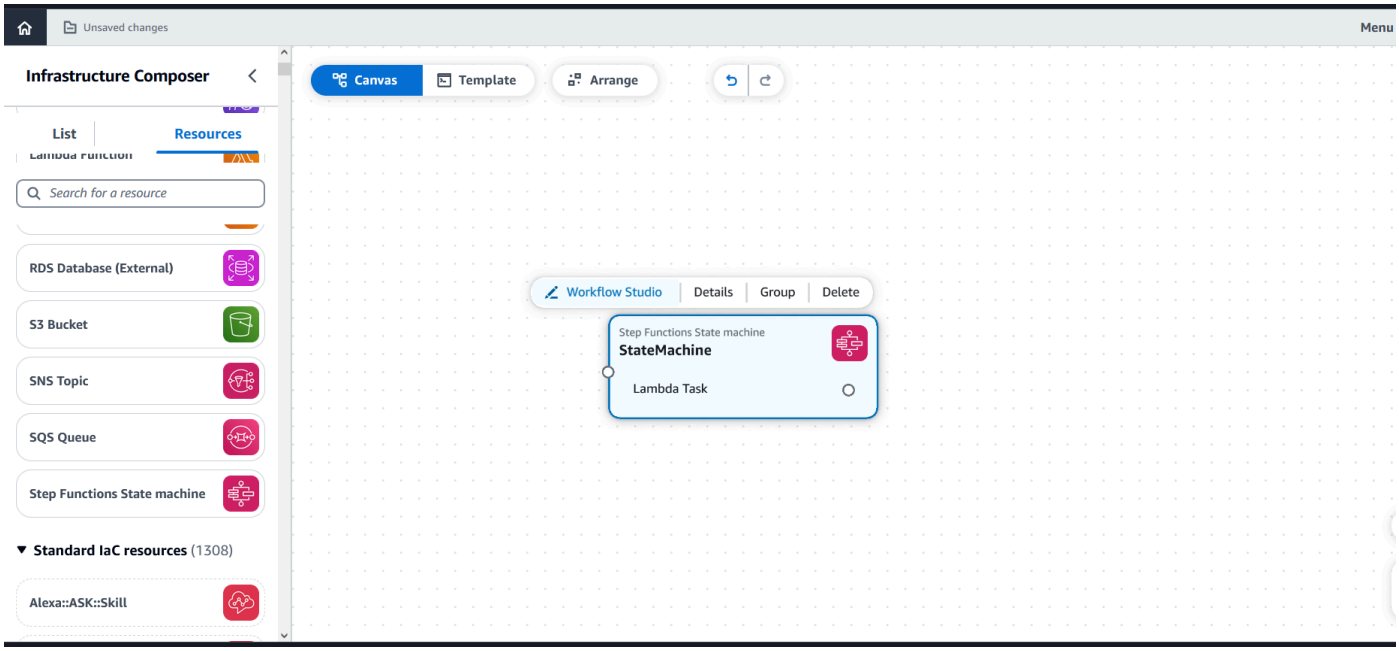
將任何 AWS CloudFormation 資源拖曳到畫布上，以構成您的應用程式架構。Infrastructure Composer 提供啟動的 IaC 範本，可用來指定資源的屬性。如需進一步了解，請參閱 [在 Infrastructure Composer 中設定和修改卡片](#)。



使用 功能存取其他功能 AWS 服務

Infrastructure Composer 功能 AWS 服務 ，在建置應用程式時經常一起使用或一起設定。如需進一步了解，請參閱 與 Amazon VPC 整合。

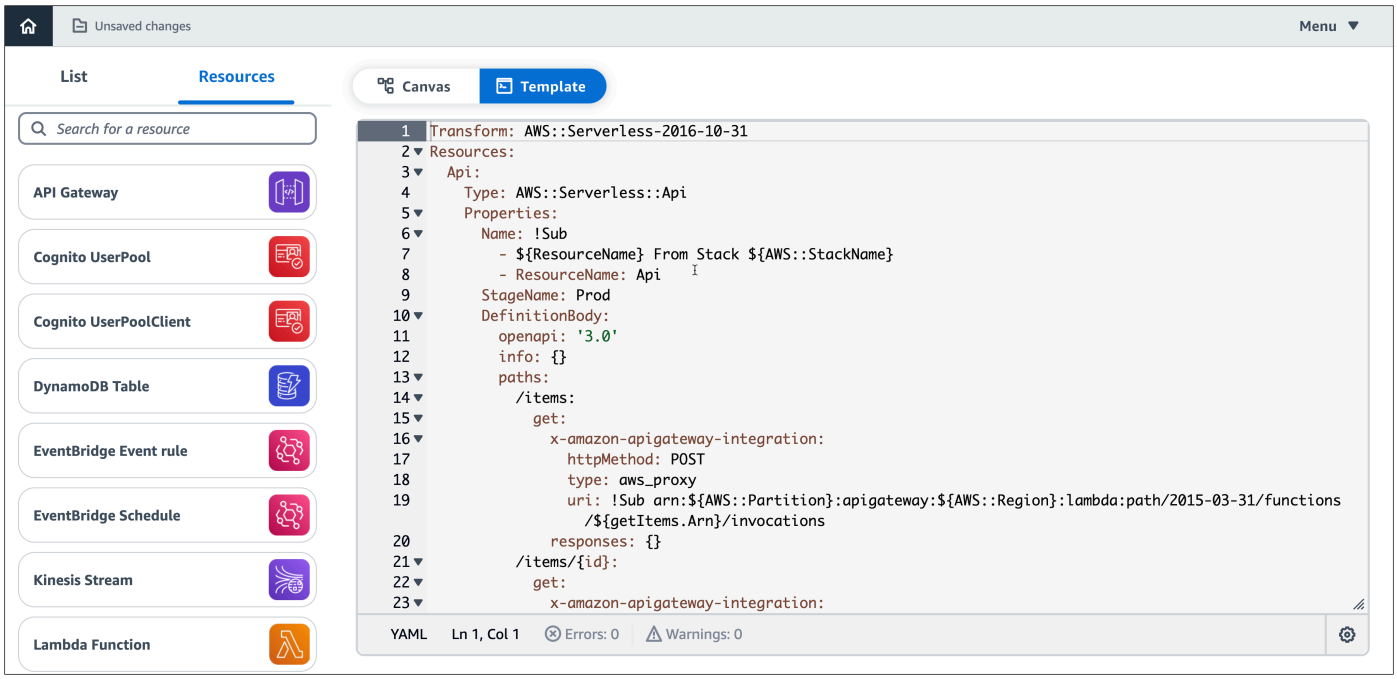
以下是 AWS Step Functions 功能的範例，其提供Workflow Studio直接在 Infrastructure Composer 畫布中啟動 Step Functions 的整合。



將基礎設施定義為程式碼 (IaC) 範本

Infrastructure Composer 會建立您的基礎設施程式碼

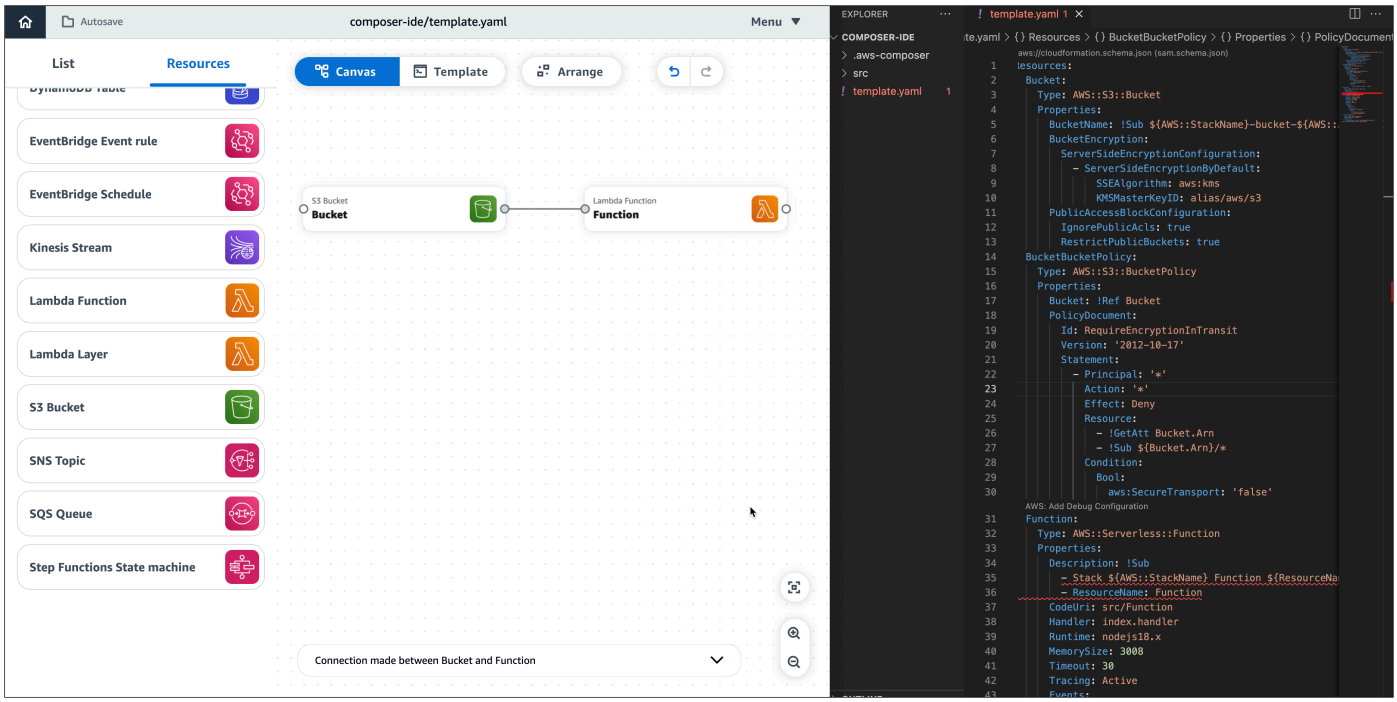
當您編寫時，基礎設施編寫器會自動建立您的 AWS CloudFormation 和 AWS Serverless Application Model (AWS SAM) 範本，並遵循 AWS 最佳實務。您可以直接從 Infrastructure Composer 中檢視和修改範本。Infrastructure Composer 會自動同步視覺化畫布和範本程式碼之間的變更。



與現有的工作流程整合

匯入現有的範本和專案

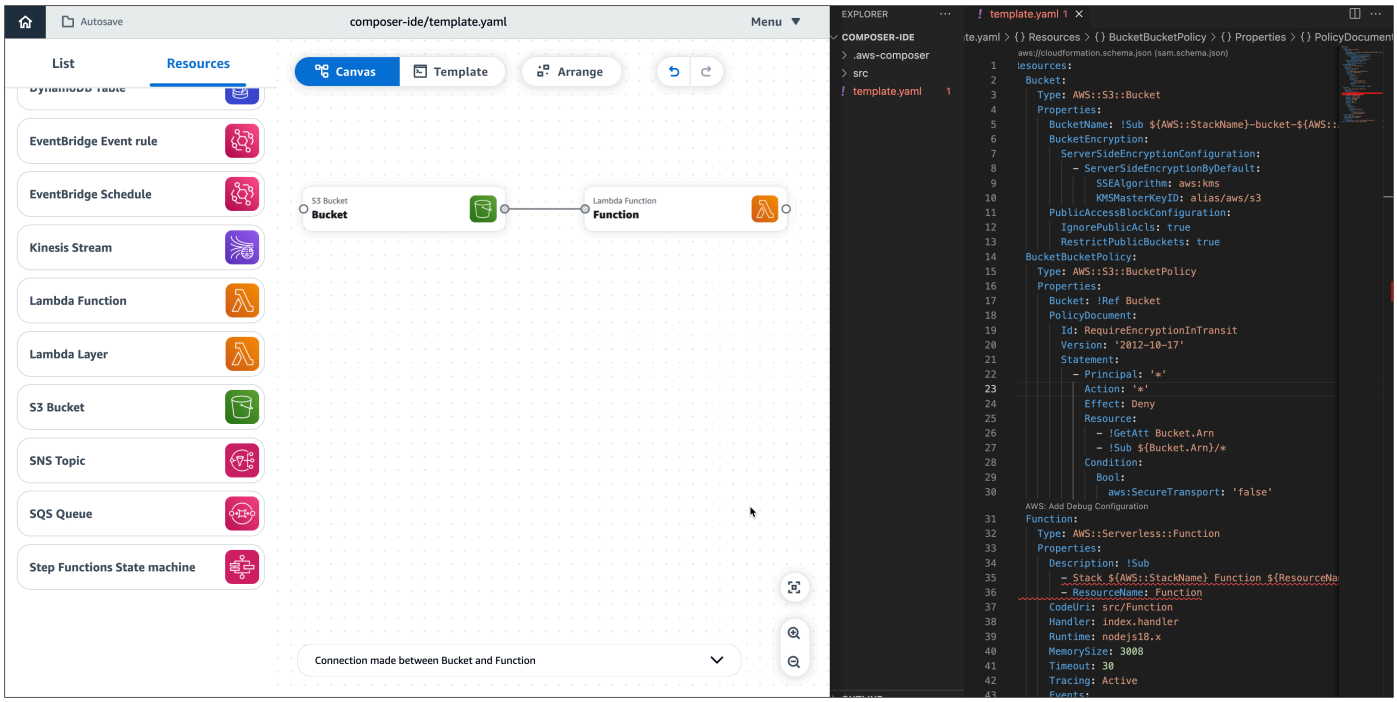
匯入現有的 AWS CloudFormation 和 AWS SAM 範本，以視覺化方式呈現它們，以便更好地了解和修改其設計。匯出您在 Infrastructure Composer 中建立的範本，並將其整合到您現有的工作流程中以進行部署。



存取 Infrastructure Composer 的方法

從 Infrastructure Composer 主控台

透過 Infrastructure Composer 主控台存取 Infrastructure Composer 以快速入門。此外，您可以使用本機同步模式，自動同步並儲存 Infrastructure Composer 與本機機器。



從 AWS CloudFormation 主控台

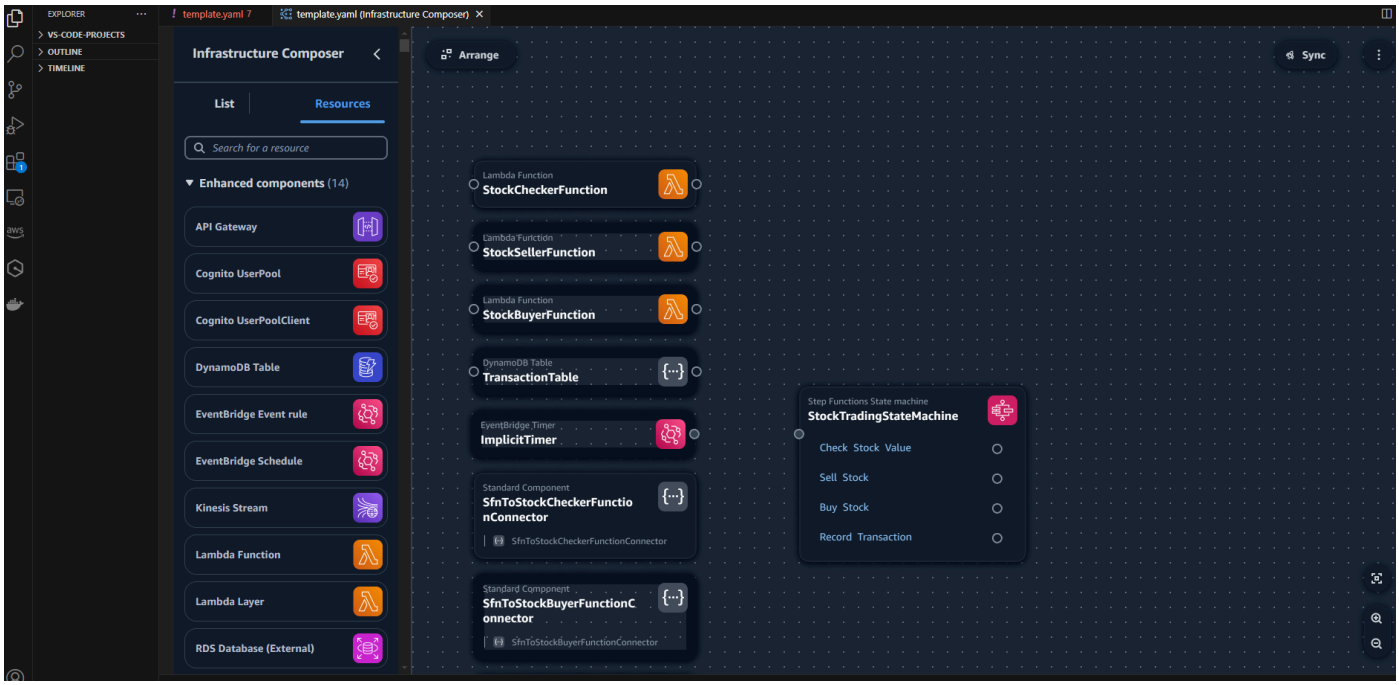
Infrastructure Composer 主控台也支援 CloudFormation 主控台模式，這是 CloudFormation 設計工具與 AWS CloudFormation 堆疊工作流程整合的改善。此新工具現在是視覺化 CloudFormation 範本的建議工具。

從 Lambda 主控台

使用 Infrastructure Composer，您也可以從 Lambda 主控台匯入 Lambda 函數。如需進一步了解，請參閱 從 Lambda 主控台將函數匯入 Infrastructure Composer。

從 AWS Toolkit for Visual Studio Code

透過 Toolkit for VS 程式碼延伸來存取 Infrastructure Composer，將 Infrastructure Composer 帶入您的本機開發環境。



進一步了解

若要繼續了解 Infrastructure Composer，請參閱下列資源：

- Infrastructure Composer 卡
- 視覺化編寫和建立無伺服器應用程式 | 無伺服器辦公時間 – Infrastructure Composer 概觀和示範。

後續步驟

若要設定 Infrastructure Composer，請參閱 Infrastructure Composer 主控台入門。

的無伺服器概念 AWS Infrastructure Composer

使用前，了解基本的無伺服器概念 AWS Infrastructure Composer。

無伺服器概念

事件驅動型架構

無伺服器應用程式包含個別 AWS 服務，例如 AWS Lambda 用於運算的 和用於資料庫管理的 Amazon DynamoDB，每個服務都會執行專門的角色。然後，這些服務會透過事件驅動的架構彼此鬆散整合。若要進一步了解事件驅動架構，請參閱什麼是事件驅動架構？。

基礎設施即程式碼 (IaC)

基礎設施即程式碼 (IaC) 是一種處理基礎設施的方式，與開發人員處理程式碼的方式相同，將相同的嚴格應用程式程式碼開發套用至基礎設施佈建。您可以在範本檔案中定義基礎設施、將其部署到 AWS，並為您 AWS 建立資源。使用 IAC，您可以在程式碼中定義 AWS 您要佈建的內容。如需詳細資訊，請參閱白皮書上的 DevOps 簡介 AWS AWS 中的基礎設施做為程式碼。

無伺服器技術

透過無 AWS 伺服器技術，您可以建置和執行應用程式，而不必管理自己的伺服器。所有伺服器管理都由 完成 AWS，提供許多好處，例如自動擴展和內建高可用性，讓您快速將想法帶入生產環境。使用無伺服器技術，您可以專注於產品的核心，而不必擔心管理和操作伺服器。若要進一步了解無伺服器，請參閱 上的無伺服器 AWS。

如需核心無 AWS 伺服器服務的基本簡介，請參閱 Serverless 101：了解 Serverless Land 的無伺服器服務。

Infrastructure Composer 卡

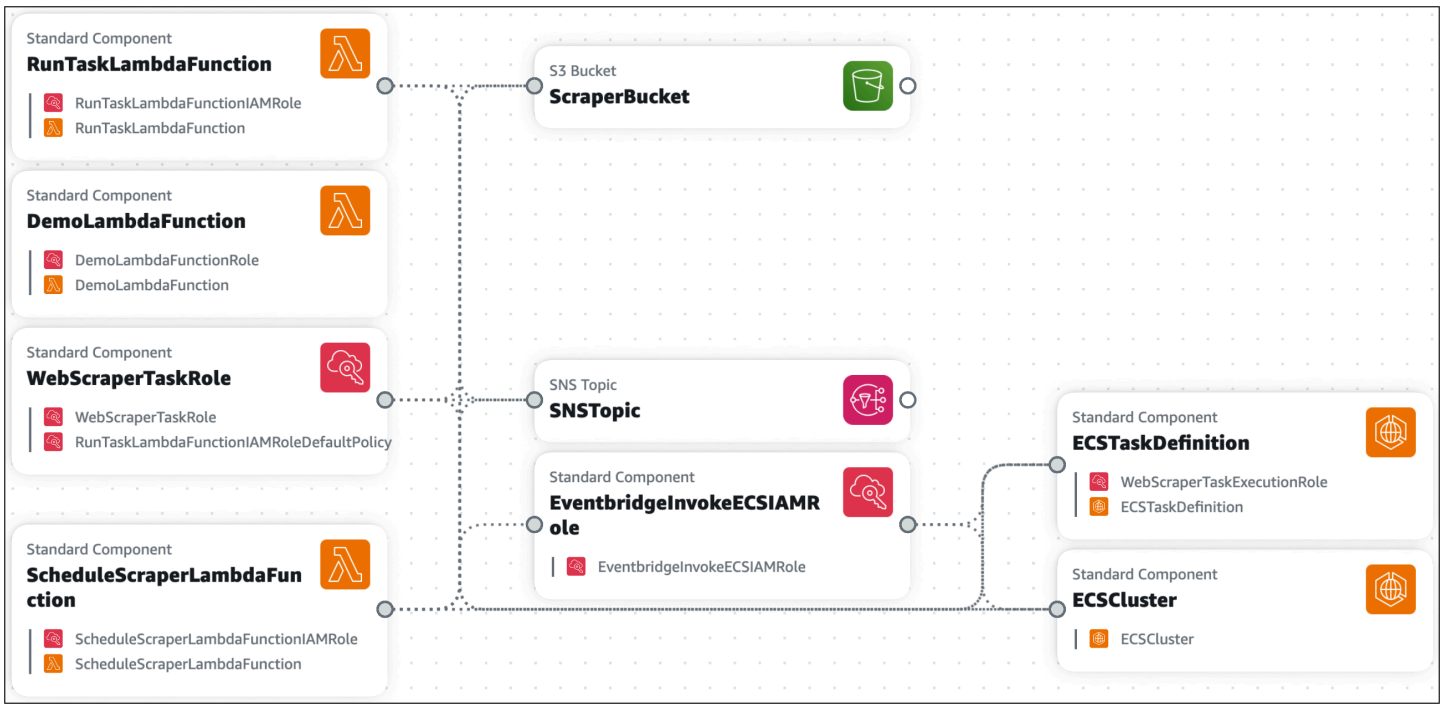
Infrastructure Composer 簡化將基礎設施寫入 AWS CloudFormation 資源程式碼 (IaC) 的程序。若要有效使用 Infrastructure Composer，您應該先了解兩個基本概念：基礎架構 Composer 卡和卡片連線。

在 Infrastructure Composer 中，卡片代表 AWS CloudFormation 資源。卡片有兩種一般類別：

- 增強型元件卡 – 已合併為單一策畫卡 AWS CloudFormation 的資源集合，可增強易用性、功能，且專為各種使用案例而設計。增強型元件卡是 Infrastructure Composer 的資源面板中列出的第一張卡片。
- 標準 IaC 資源卡 – 單一 AWS CloudFormation 資源。每個標準 IaC 資源卡一旦拖曳到畫布上，都會標示為標準元件，並且可以合併為多個資源。

> **Note**
>
> 視卡片而定，標準 IaC 資源卡在拖曳至視覺化畫布後，可能會標示為標準元件卡。這只是表示卡片是一或多個標準 IaC 資源卡的集合。

雖然某些類型的卡片可從資源調色盤取得，但當您將現有 AWS CloudFormation 或 AWS Serverless Application Model (AWS SAM) 範本匯入 Infrastructure Composer 時，卡片也可能出現在畫布上。下圖是匯入應用程式的範例，其中包含各種卡片類型：



主題

- Infrastructure Composer 中的增強型元件卡
- Infrastructure Composer 中的標準元件卡
- Infrastructure Composer 中的卡片連線

Infrastructure Composer 中的增強型元件卡

增強型元件卡由 Infrastructure Composer 建立和管理。每個卡片都包含建置應用程式時經常一起使用 AWS CloudFormation 的資源 AWS。其基礎設施程式碼是由 Infrastructure Composer 按照 AWS 最佳實務建立。增強型元件卡是開始設計應用程式的好方法。

增強型元件卡可從資源面板的增強型元件區段下取得。

增強型元件卡可在 Infrastructure Composer 中完整設定和使用，以設計和建置無伺服器應用程式。我們建議您在設計沒有現有程式碼的應用程式時使用增強型元件卡。

此資料表顯示增強型元件，其中包含卡片特色資源的 AWS CloudFormation or AWS Serverless Application Model (AWS SAM) 範本規格連結：

卡	參考資料
Amazon API Gateway	AWS::Serverless::API
Amazon Cognito UserPool	AWS::Cognito::UserPool
Amazon Cognito UserPoolClient	AWS::Cognito::UserPoolClient
Amazon DynamoDB 資料表	AWS::DynamoDB::Table
Amazon EventBridge 事件規則	AWS::Events::Rule
EventBridge 排程	AWS::Scheduler::Schedule
Amazon Kinesis 串流	AWS::Kinesis::Stream
AWS Lambda 函數	AWS::Serverless::Function
Lambda Layer	AWS::Serverless::LayerVersion
Amazon Simple Storage Service (Amazon S3) 儲存貯體	AWS::S3::Bucket
Amazon Simple Notification Service (Amazon SNS) 主題	AWS::SNS::Topic
Amazon Simple Queue Service (Amazon SQS) 佇列	AWS::SQS::Queue
AWS Step Functions 狀態機器	AWS::Serverless::StateMachine

範例

以下是 S3 儲存貯體增強型元件的範例：

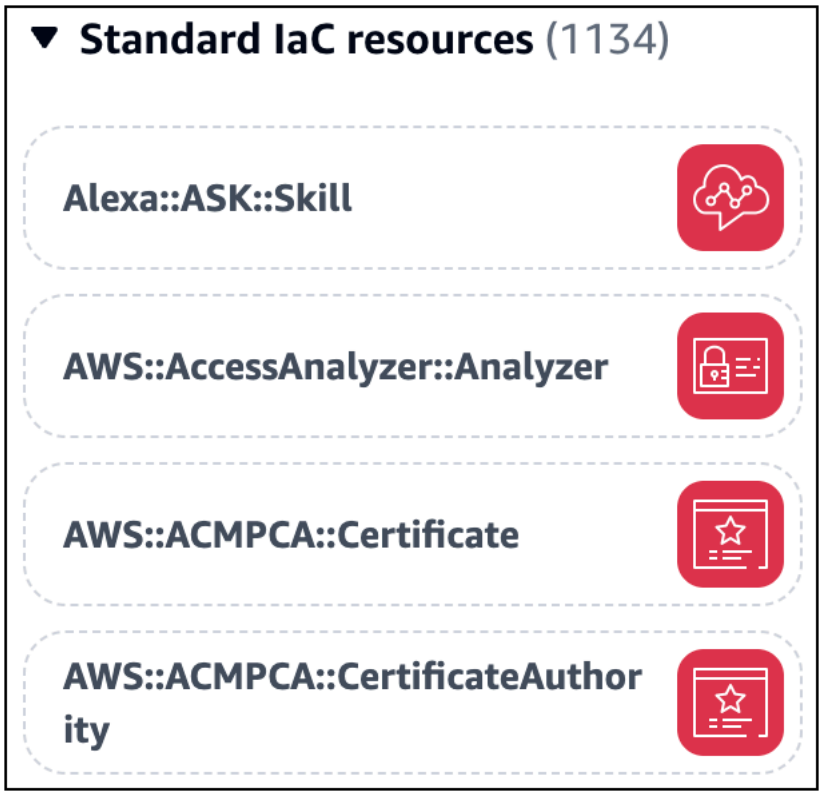
當您將 S3 儲存貯體元件卡拖曳到畫布上並檢視範本時，您會看到下列兩個 AWS CloudFormation 資源新增至範本：

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

S3 儲存貯體增強型元件卡代表 Amazon Simple Storage Service (Amazon S3) 儲存貯體與應用程式中其他服務互動所需的兩種 AWS CloudFormation 資源。

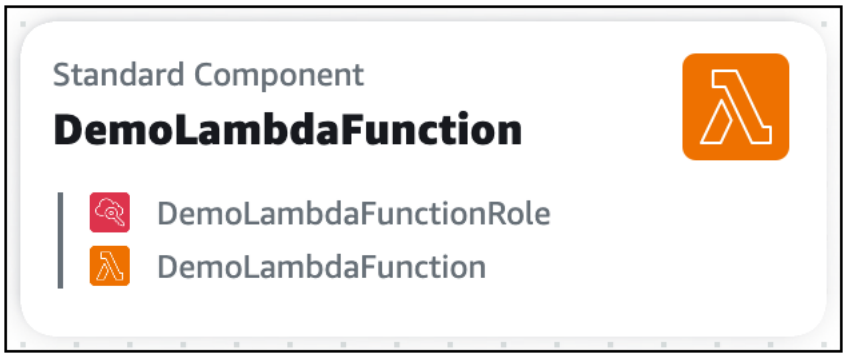
Infrastructure Composer 中的標準元件卡

在將標準元件卡放置在 Infrastructure Composer 的視覺化畫布上之前，它會在 Infrastructure Composer 的資源面板上列為標準 (IaC) 資源卡。標準 (IaC) 資源卡代表單一 AWS CloudFormation 資源。每個標準 IaC 資源卡一旦放置在視覺化畫布上，就會成為標有標準元件的卡片，並且可以合併以代表多個 AWS CloudFormation 資源。

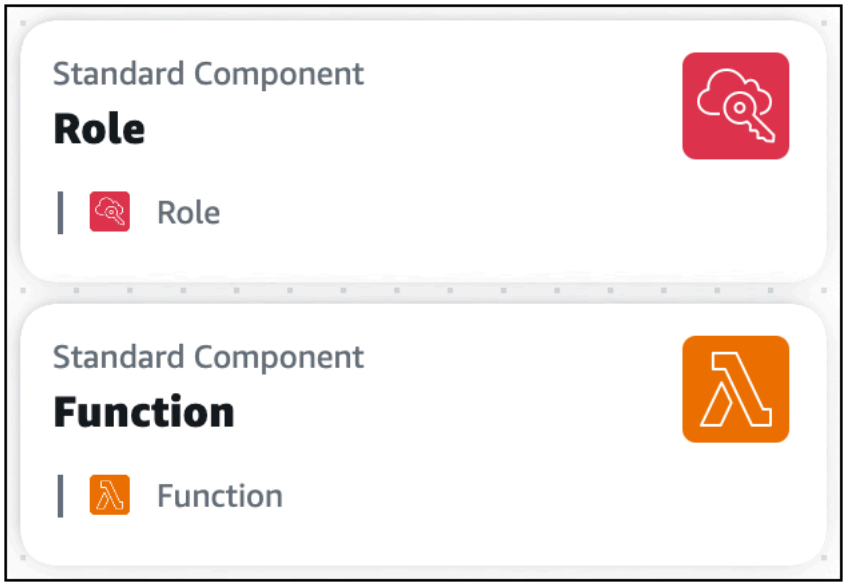


每個標準 IaC 資源卡都可以依其 AWS CloudFormation 資源類型來識別。以下是代表 資源類型的標準 IaC `AWS::ECS::Cluster` AWS CloudFormation 資源卡範例：

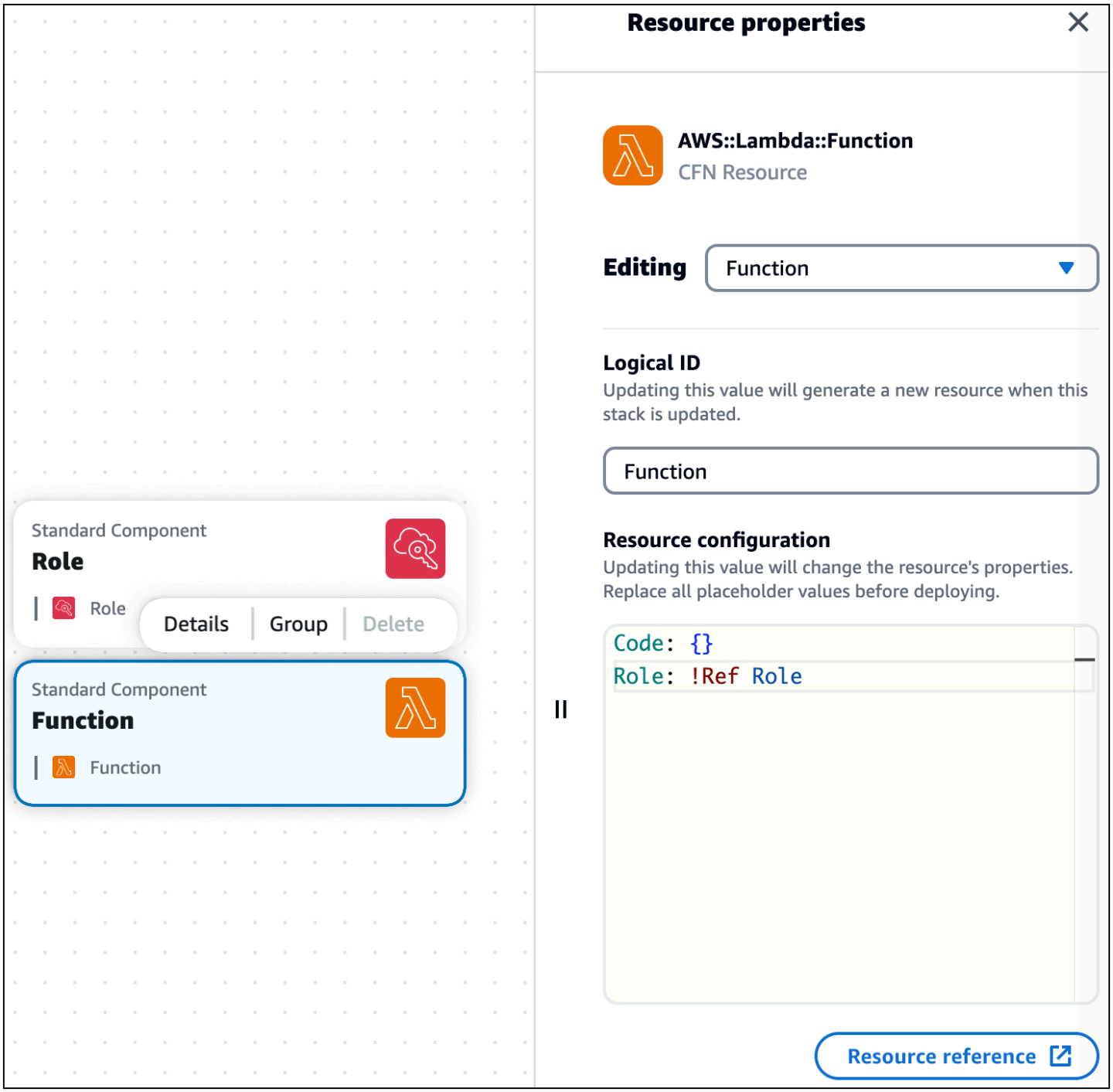
每個標準元件卡都會視覺化其中包含 AWS CloudFormation 的資源。以下是包含兩個標準 IaC 資源的標準元件卡範例：



當您設定標準元件卡的屬性時， Infrastructure Composer 可能會將相關卡結合在一起。例如，以下有兩個標準元件卡：



在代表 `AWS::Lambda::Function` 資源的標準元件卡的資源屬性面板中，我們會依其邏輯 ID 來參考 AWS Identity and Access Management (IAM) 角色：



儲存範本後，兩個標準元件卡會合併為單一標準元件卡。

Infrastructure Composer 中的卡片連線

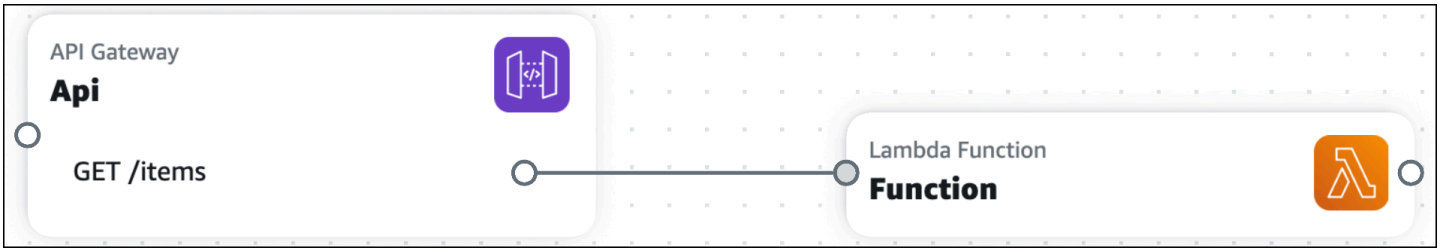
在 中 AWS Infrastructure Composer，兩張卡片之間的連線會以一行視覺方式顯示。這些行代表應用程式中的事件驅動關係。

主題

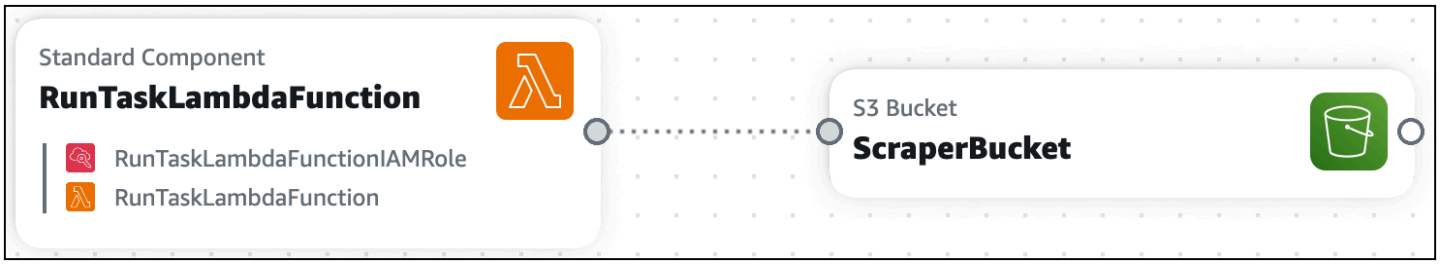
- 卡片之間的連線
- 增強型元件卡之間的連線
- 標準 IaC 資源卡的往返連線

卡片之間的連線

如何將卡片連接在一起，取決於卡片類型。每個增強型卡至少有一個連接器連接埠。若要連接它們，您只需選取一個連接器連接埠，並將其拖曳至另一張卡片的連接埠，基礎設施編譯器就會連接兩個資源，或顯示訊息，指出不支援此組態。



如上所示，增強型元件卡之間的線條是實心的。相反地，標準 IaC 資源卡（也稱為標準元件卡）沒有連接器連接埠。對於這些卡片，您必須在應用程式的範本中指定這些事件驅動關係，而 Infrastructure Composer 會自動偵測其連線，並在卡片之間使用虛線將它們視覺化。

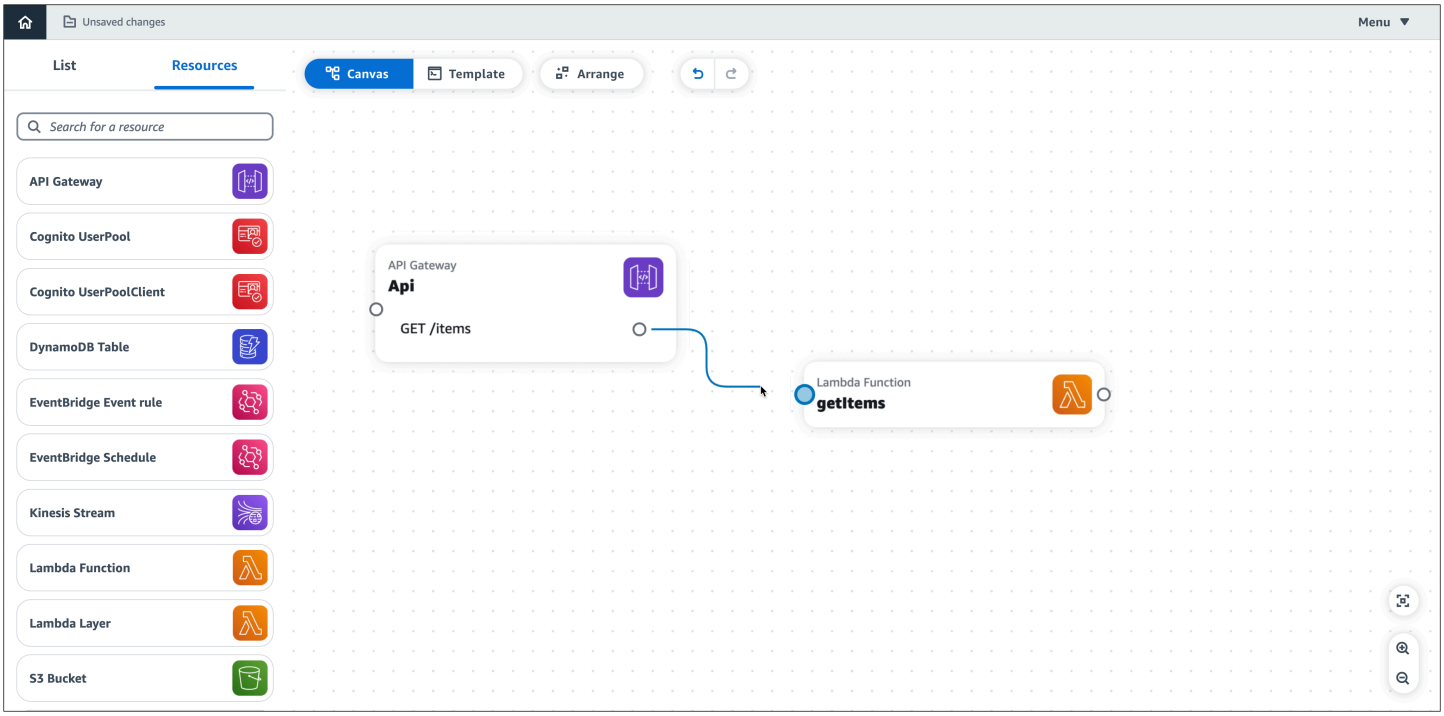


若要進一步了解，請參閱以下各節。

增強型元件卡之間的連線

在 Infrastructure Composer 中，兩條增強型元件卡之間的連線會以實線視覺顯示。這些行代表應用程式中的事件驅動關係。

若要連接兩張卡片，請按一下一張卡片的連接埠，並將其拖曳到另一張卡片的連接埠。



> **Note**
>
> 標準 IaC 資源卡沒有連接器連接埠。對於這些卡片，您必須在應用程式的範本中指定其事件驅動關係，而 Infrastructure Composer 會自動偵測其連線，並在卡片之間使用虛線將它們視覺化。

如需詳細資訊，請參閱在 Infrastructure Composer 的視覺化畫布上連接卡片。

增強型元件卡佈建

兩個卡片之間的連線，以一行視覺方式表示，必要時佈建以下內容：

- AWS Identity and Access Management (IAM) 政策
- 環境變數
- 事件

IAM 政策

當資源需要呼叫其他資源的許可時， Infrastructure Composer 會使用 AWS Serverless Application Model (AWS SAM) 政策範本佈建以資源為基礎的政策。

- 若要進一步了解 IAM 許可和政策，請參閱《IAM 使用者指南》中的存取管理概觀：許可和政策。
- 若要進一步了解 AWS SAM 政策範本，請參閱《 AWS Serverless Application Model 開發人員指南》中的AWS SAM 政策範本。

環境變數

環境變數是暫時值，可以變更這些值來影響資源的行為。必要時，基礎設施編譯器會定義基礎設施程式碼，以在資源之間利用環境變數。

事件

資源可以透過不同類型的事件叫用其他資源。必要時，基礎設施編寫器會定義透過事件類型互動資源所需的基礎設施程式碼。

標準 IaC 資源卡的往返連線

所有 AWS CloudFormation 資源都可以用作資源調色盤中的標準 IaC 資源卡。當您將標準 IaC 資源卡拖曳到畫布上時，標準 IaC 資源卡會成為標準元件卡，這會提示 Infrastructure Composer 在應用程式中為您的資源建立啟動範本。

如需詳細資訊，請參閱Infrastructure Composer 中的標準卡。

Infrastructure Composer 主控台入門

使用本節中的主題來設定 AWS Infrastructure Composer 並了解如何使用視覺化畫布設計應用程式。本節中的導覽和教學課程會顯示在 Infrastructure Composer 主控台中，這是預設的使用者體驗。本節中的主題說明如何完成使用 Infrastructure Composer 的先決條件、使用 Infrastructure Composer 主控台、載入和修改專案，以及建置您的第一個應用程式。

Infrastructure Composer 也可以從 AWS Toolkit for Visual Studio Code 和 CloudFormation 主控台模式中使用。工具之間的體驗通常相同，但每個工具之間有一些差異。如需在這些工具中使用 Infrastructure Composer 的詳細資訊，請參閱您可以在其中使用 Infrastructure Composer。

主題

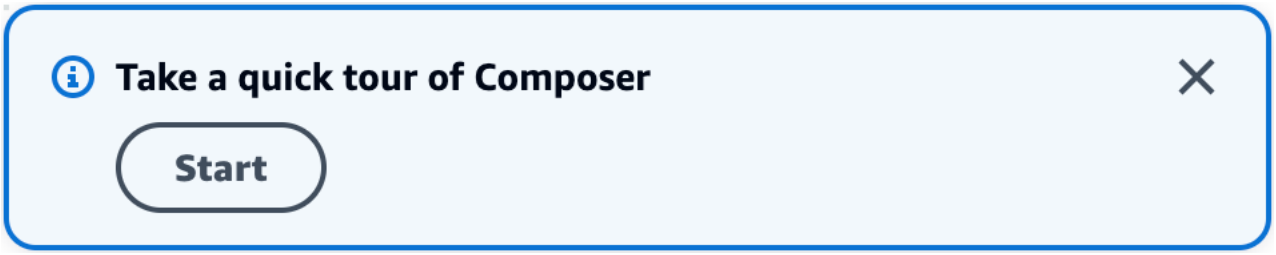
- 在 Infrastructure Composer 主控台中進行導覽
- 載入和修改 Infrastructure Composer 示範專案
- 使用 Infrastructure Composer 建置您的第一個應用程式

在 Infrastructure Composer 主控台中進行導覽

若要了解 AWS Infrastructure Composer 運作方式的一般概念，請參加內建於 Infrastructure Composer 主控台的導覽。如需 Infrastructure Composer 主控台的概觀，請參閱 在 Infrastructure Composer 主控台中進行導覽。如需使用 Infrastructure Composer 的深入指引，請參閱 如何在 中編寫 AWS Infrastructure Composer。

導覽 Infrastructure Composer

1. 登入 Infrastructure Composer 主控台。
2. 在 首頁上，選擇開啟示範。
3. 在右上角的快速導覽 Composer 視窗中，選擇開始。



4. 在撰寫者導覽視窗中，執行下列動作：

- 若要移至下一個步驟，請選擇下一步。
- 若要返回上一個步驟，請選擇上一個。
- 在最後一個步驟中，若要完成導覽，請選擇結束。

導覽提供基本 Infrastructure Composer functonality 的簡短概觀，例如使用、設定和連接卡片。如需詳細資訊，請參閱 如何在 中編寫 AWS Infrastructure Composer。

後續步驟

若要在 Infrastructure Composer 中載入和修改專案，請參閱 載入和修改 Infrastructure Composer 示範專案。

載入和修改 Infrastructure Composer 示範專案

使用此教學課程，透過 Infrastructure Composer 的使用者介面成為家庭，並了解如何載入、修改和儲存 Infrastructure Composer 示範專案。

本教學課程會在 Infrastructure Composer 主控台中完成。完成後，您就可以開始 使用 Infrastructure Composer 建置您的第一個應用程式。

主題

步驟 1：開啟示範

透過建立示範專案開始使用 Infrastructure Composer。

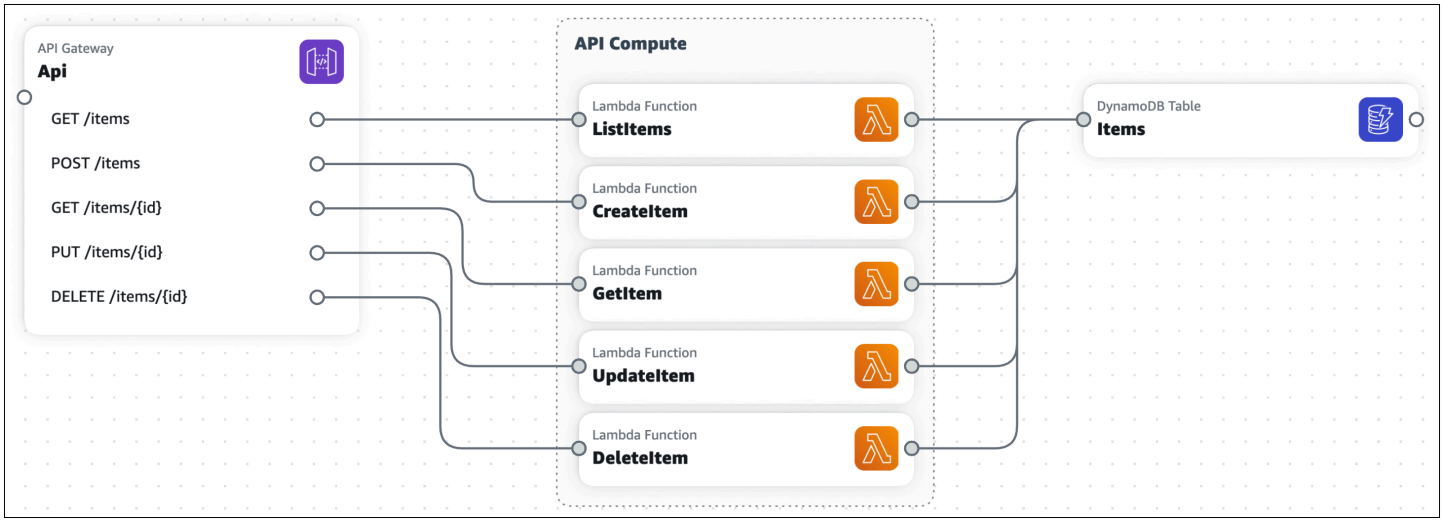
建立示範專案

1. 登入 Infrastructure Composer 主控台。
2. 在 首頁上，選擇開啟示範。

示範應用程式是基本的建立、讀取、刪除和更新 (CRUD) 無伺服器應用程式，其中包括：

- 具有五個路由的 Amazon API Gateway 資源。
- 五個 AWS Lambda 函數。
- Amazon DynamoDB 資料表。

下圖為示範：

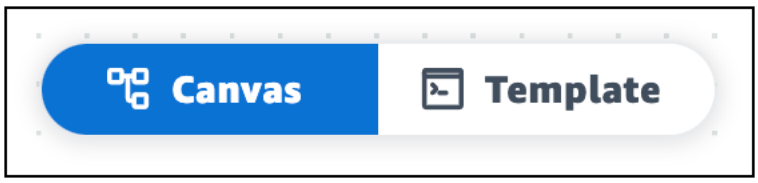


步驟 2：探索 Infrastructure Composer 的視覺化畫布

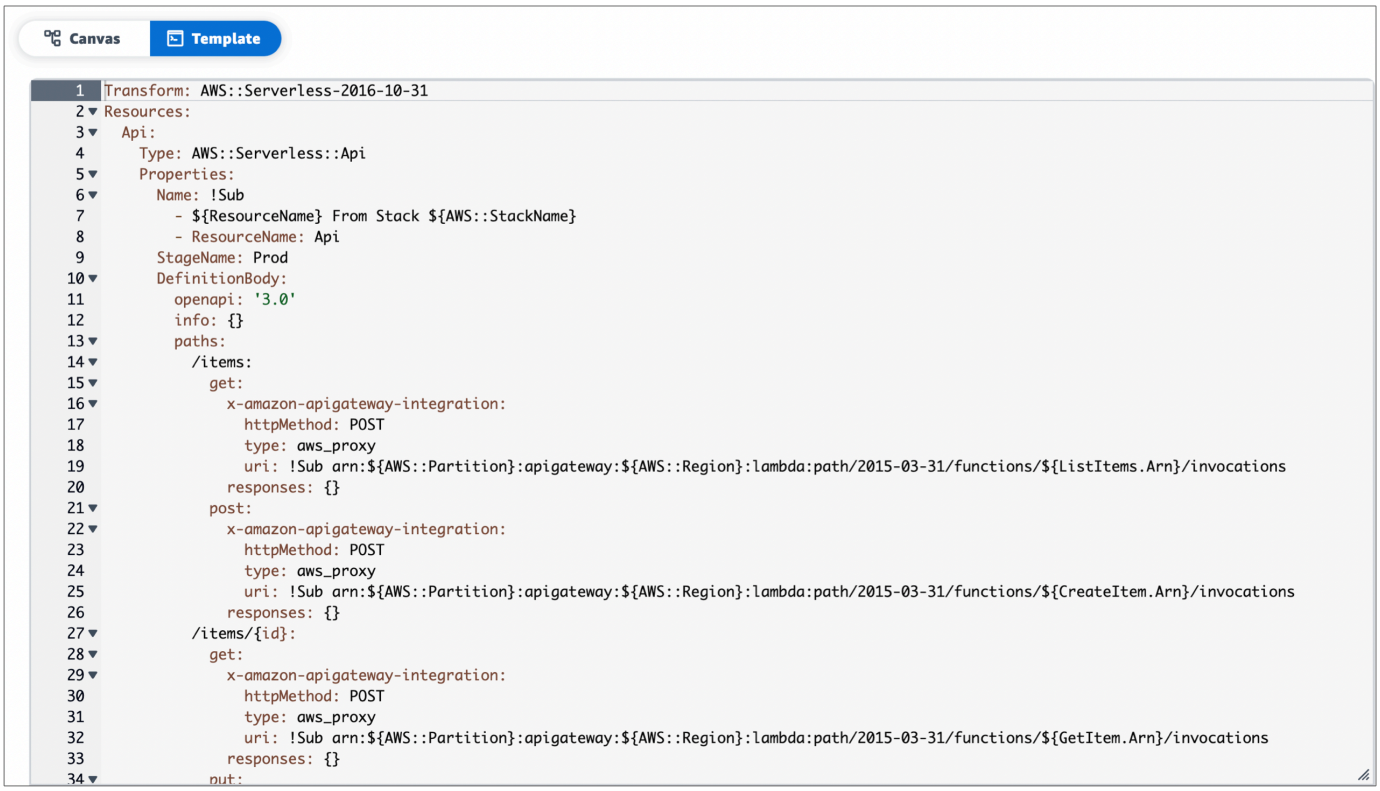
了解視覺化畫布的功能，以建置您的 Infrastructure Composer 示範專案。如需視覺化畫布配置的概觀，請參閱 視覺化概觀。

探索視覺化畫布的功能

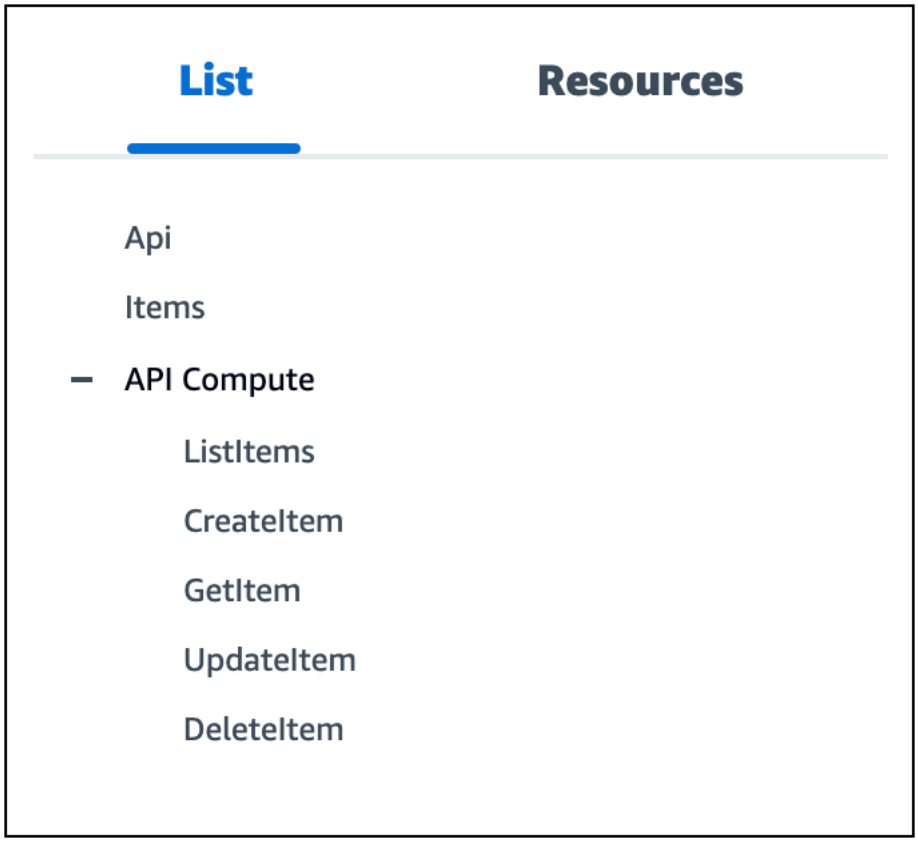
1. 當您開啟新的或現有的應用程式專案時， Infrastructure Composer 會載入畫布檢視，如主檢視區域上方所示。



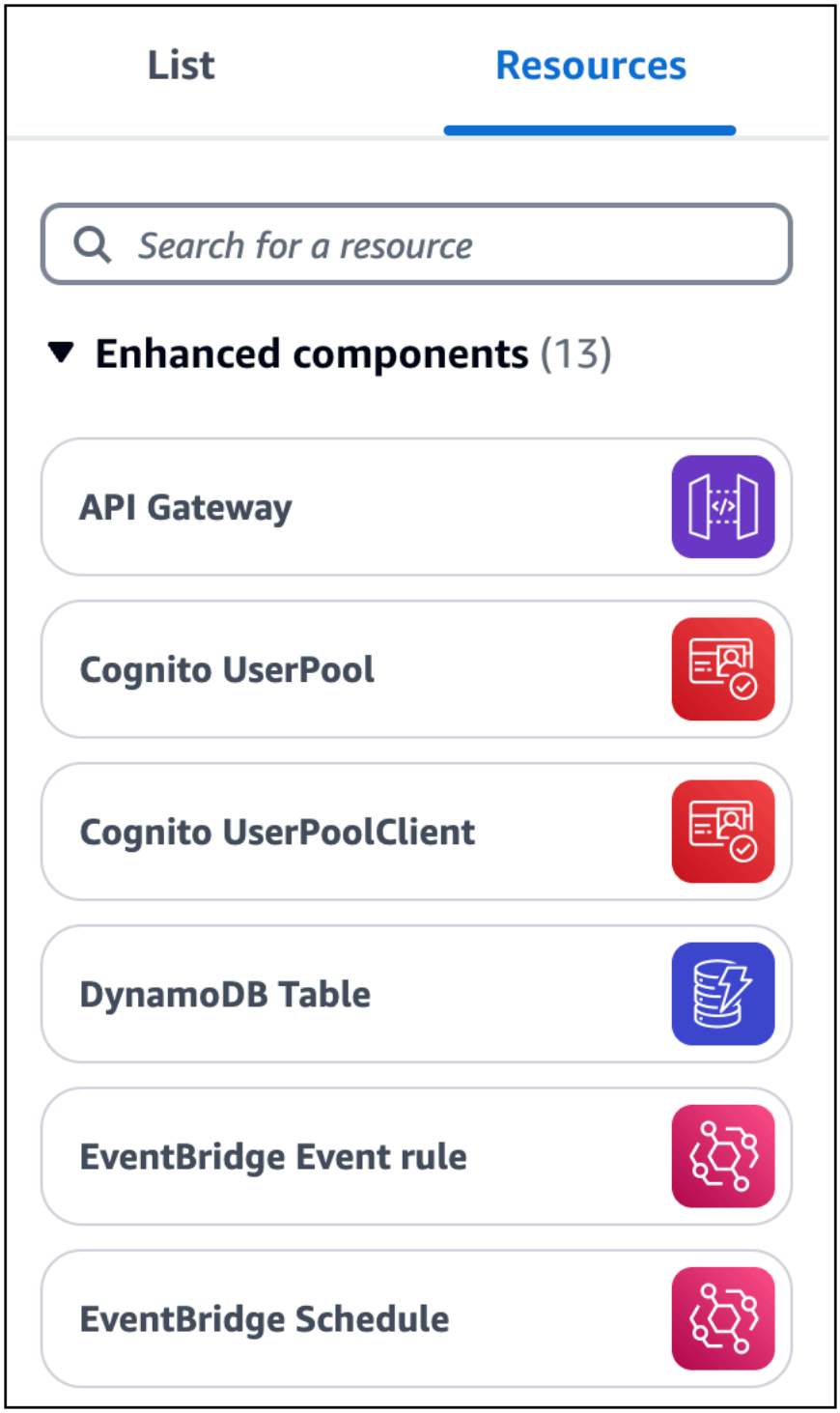
若要在主檢視區域中顯示應用程式的基礎設施代碼，請選擇範本。例如，以下是 Infrastructure Composer 示範專案的 AWS Serverless Application Model (AWS SAM) 範本檢視。



2. 若要再次顯示應用程式的畫布檢視，請選擇畫布。
3. 若要在樹狀檢視中顯示您應用程式的資源，請選擇清單。



4. 若要顯示資源調色盤，請選擇資源。此調色盤具有可用來擴展應用程式架構的卡片。您可以搜尋卡片或捲動清單。



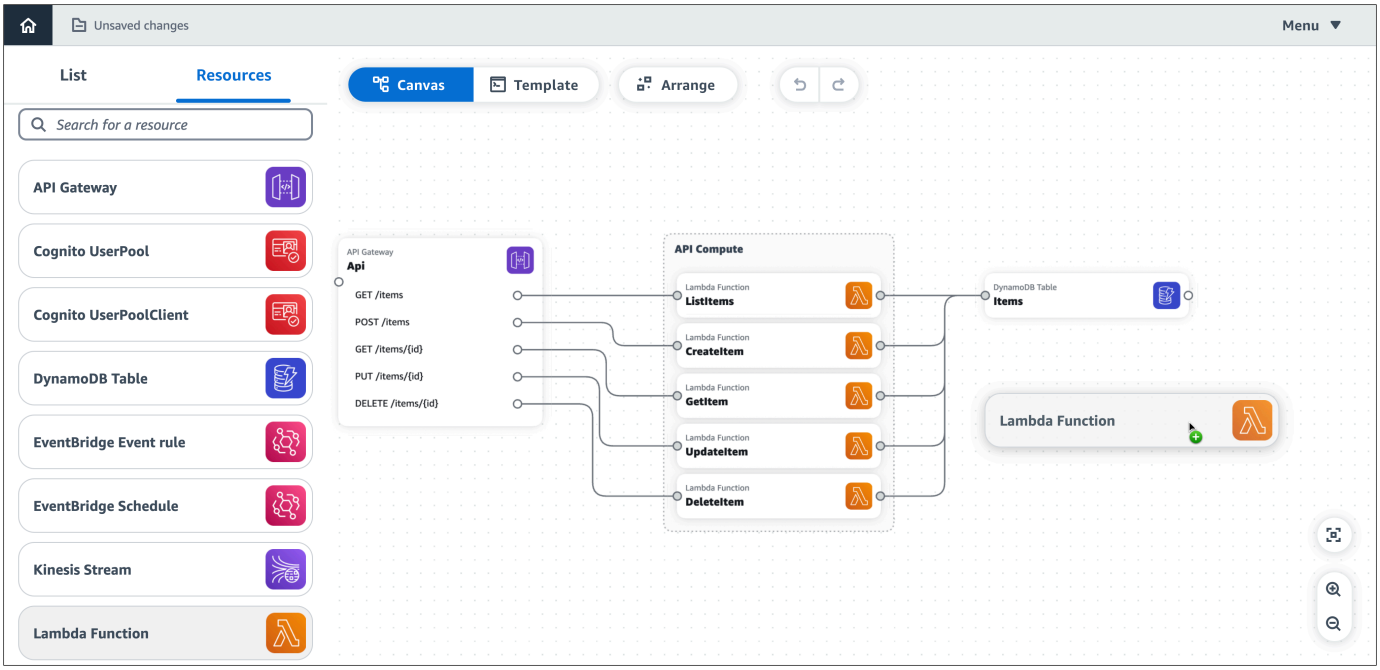
5. 若要在視覺化畫布中移動，請使用基本手勢。如需詳細資訊，請參閱將卡片放在畫布上。

步驟 3：展開您的應用程式架構

在此步驟中，您將透過將 Lambda 函數新增至 DynamoDB 資料表來擴展應用程式架構。

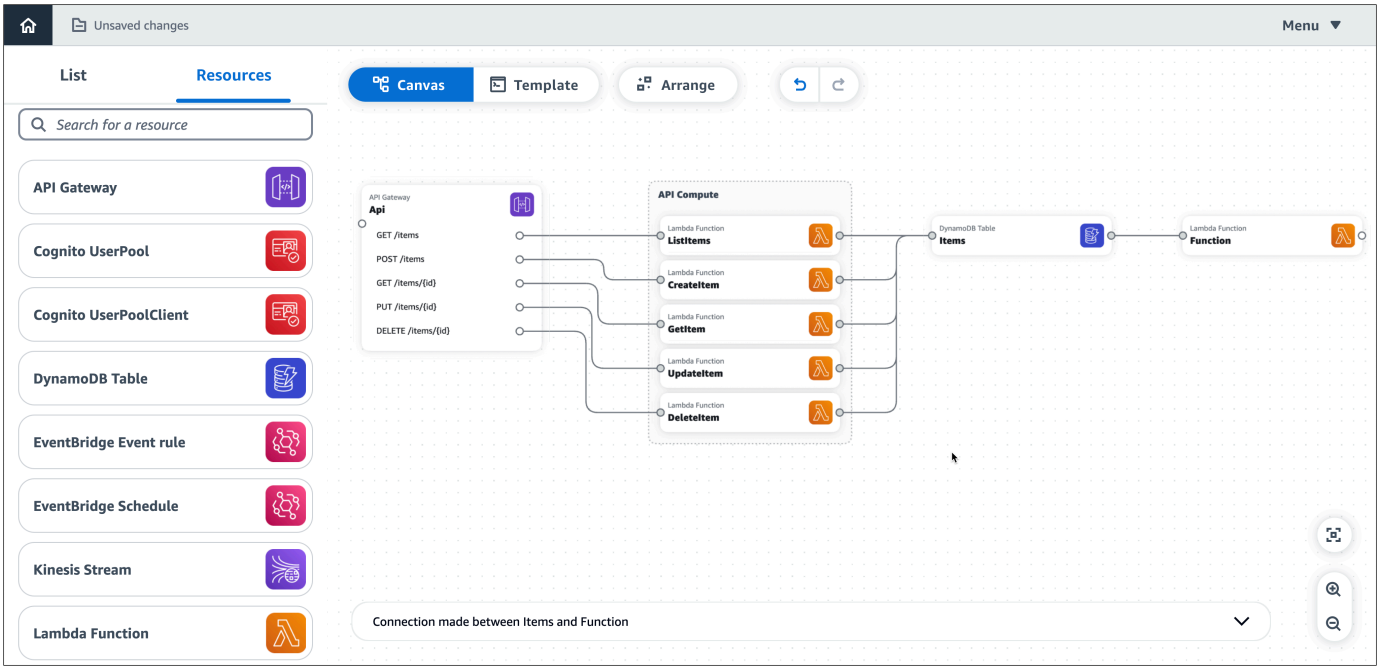
將 Lambda 函數新增至 DynamoDB 資料表

1. 從資源調色盤 (資源)，將 Lambda 函數增強型元件卡拖曳到畫布上 DynamoDB 資料表卡的右側。



2. 將 DynamoDB 資料表連接至 Lambda 函數。若要連接它們，請按一下 DynamoDB 資料表卡的右側連接埠，並將其拖曳至 Lambda 函數卡的左側連接埠。

3. 選擇排列以在畫布檢視中組織卡片。



4. 設定您的 Lambda 函數。若要設定它，請執行下列其中一項操作：

- 在畫布檢視中，修改資源屬性面板上的函數屬性。若要開啟面板，請按兩下 Lambda 函數卡。或者，選取卡片，然後選擇詳細資訊。如需資源屬性面板中所列可設定 Lambda 函數屬性的詳細資訊，請參閱 AWS Lambda 開發人員指南。
- 在範本檢視中，修改函數的程式碼 (`AWS::Serverless::Function`)。Infrastructure Composer 會自動將您的變更同步至畫布。如需範本中 AWS SAM 函數資源的詳細資訊，請參閱AWS SAM 資源和屬性參考中的 AWS::Serverless::Function。

步驟 4：儲存您的應用程式

手動將應用程式範本儲存至本機機器，或啟用本機同步，以儲存您的應用程式。

手動儲存您的應用程式範本

1. 從功能表中，選取儲存 > 儲存範本檔案。
2. 為您的範本提供名稱，然後選擇本機機器上的位置以儲存範本。按下儲存。

如需啟用本機同步的指示，請參閱 在 Infrastructure Composer 主控台中本機同步和儲存您的專案。

後續步驟

若要開始建置您的第一個應用程式，請參閱 使用 Infrastructure Composer 建置您的第一個應用程式。

使用 Infrastructure Composer 建置您的第一個應用程式

在本教學課程中，您會使用 AWS Infrastructure Composer 建置建立、讀取、更新和刪除 (CRUD) 無伺服器應用程式，以管理資料庫中的使用者。

在此教學課程中，我們會在 中使用 Infrastructure Composer AWS Management Console。建議您使用 Google Chrome或 Microsoft Edge，以及全螢幕瀏覽器視窗。

> **您是第一次使用無伺服器嗎？**
>
> 我們建議您對下列主題有基本的了解：
>
> - 事件驅動型架構
> - 基礎設施即程式碼 (IaC)
> - 無伺服器技術

如需進一步了解，請參閱 的無伺服器概念 AWS Infrastructure Composer。

主題

資源屬性參考

建置應用程式時，請使用此表格做為參考，以設定 Amazon API Gateway AWS Lambda 和資源的屬性。

方法	路徑	函數名稱
GET	/項目	getItems
GET	/items/{id}	getItem
PUT	/items/{id}	updateItem
POST	/項目	addItem
DELETE	/items/{id}	deleteItem

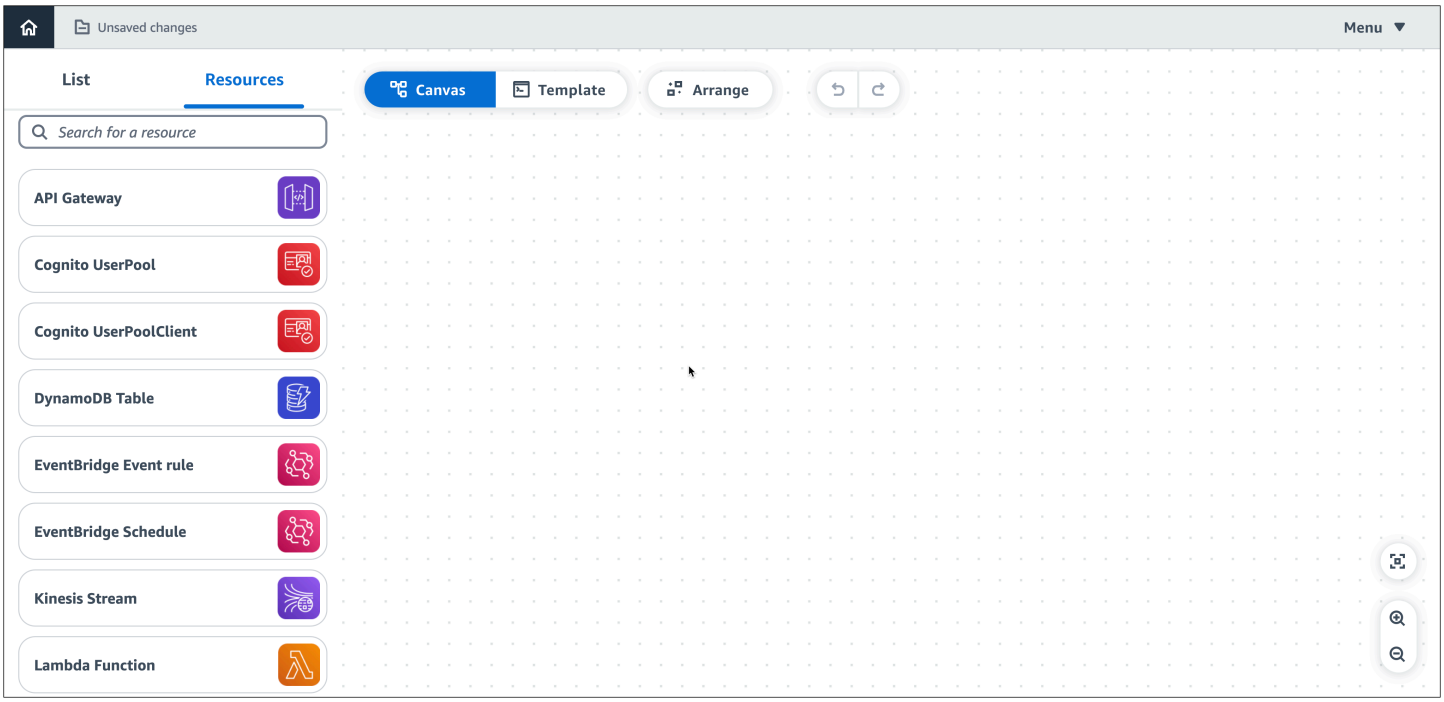
步驟 1：建立您的專案

若要開始使用 CRUD 無伺服器應用程式，請在 Infrastructure Composer 中建立新的專案，並啟用本機同步。

建立新的空白專案

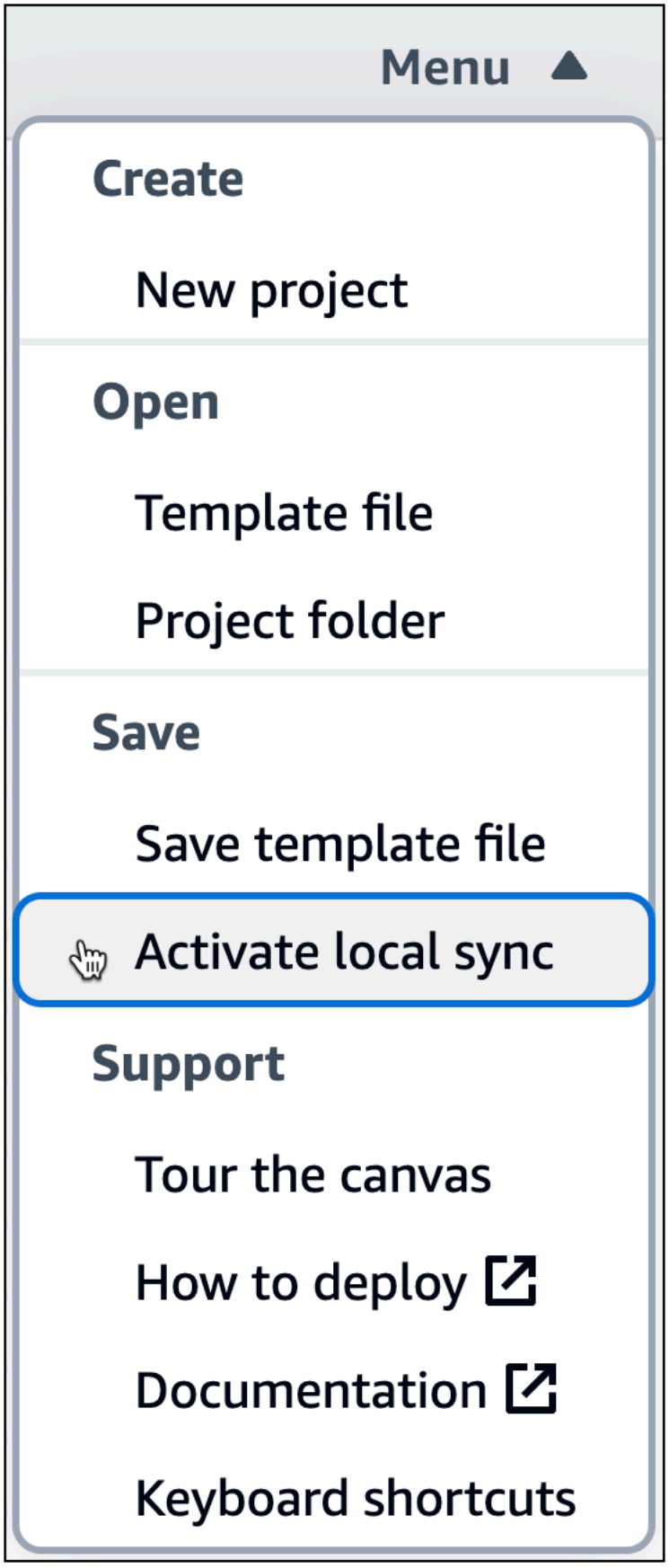
1. 登入 Infrastructure Composer 主控台。
2. 在首頁上，選擇建立專案。

如下圖所示， Infrastructure Composer 會開啟視覺化畫布，並載入啟動 （空白） 應用程式範本。



啟用本機同步

1. 從基礎設施編寫器功能表中，選取儲存 > 啟用本機同步。


Menu
Create
New project
Open
Template file
Project folder
Save
Save template file
Activate local sync
Support
Tour the canvas
How to deploy
Documentation
Keyboard shortcuts

2. 針對專案位置，按下選取資料夾，然後選擇目錄。這是 Infrastructure Composer 會在您設計時儲存和同步範本檔案和資料夾的地方。

   專案位置不得包含現有的應用程式範本。

   > **Note**
   >
   > 本機同步需要支援檔案系統存取 API 的瀏覽器。如需詳細資訊，請參閱Data Infrastructure Composer 存取。

3. 出現允許存取的提示時，請選取檢視檔案。
4. 按下啟用以開啟本機同步。出現儲存變更的提示時，請選取儲存變更。

   啟用時，Autosave 指標會顯示在畫布的左上區域。

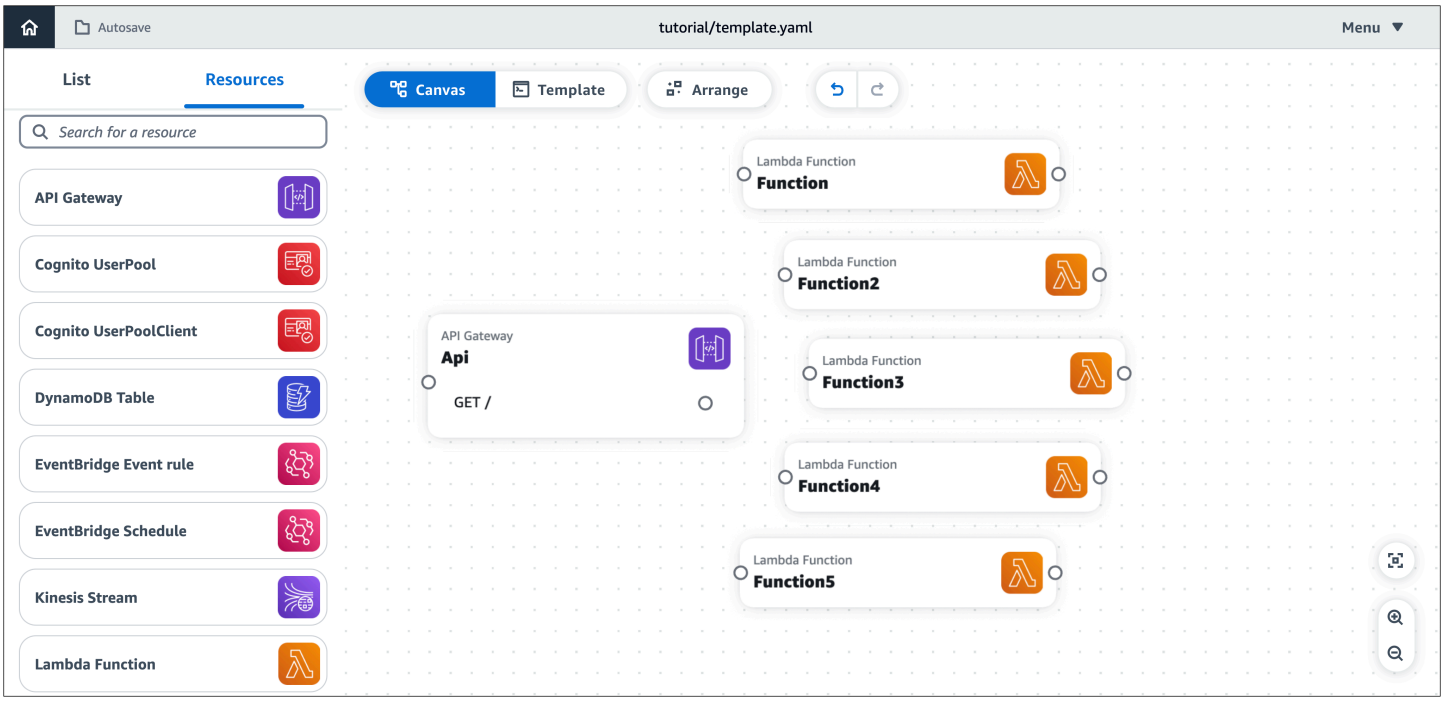
步驟 2：將卡片新增至畫布

從 API Gateway REST API 和五個 Lambda 函數開始，使用增強型元件卡設計您的應用程式架構。

將 API Gateway 和 Lambda 卡新增至畫布

從資源調色盤的增強型元件區段下，執行下列動作：

1. 將 API Gateway 卡拖曳到畫布上。
2. 將 Lambda 函數卡拖曳到畫布上。重複此步驟，直到您將五個 Lambda 函數卡新增至畫布。



步驟 3：設定您的 API Gateway REST API

接下來，在您的 API Gateway 卡中新增五個路由。

將路由新增至 API Gateway 卡

1. 開啟 API Gateway 卡的資源屬性面板。若要開啟面板，請按兩下卡片。或者，選取卡片，然後選擇詳細資訊。
2. 在資源屬性面板的 Routes 下，執行下列動作：

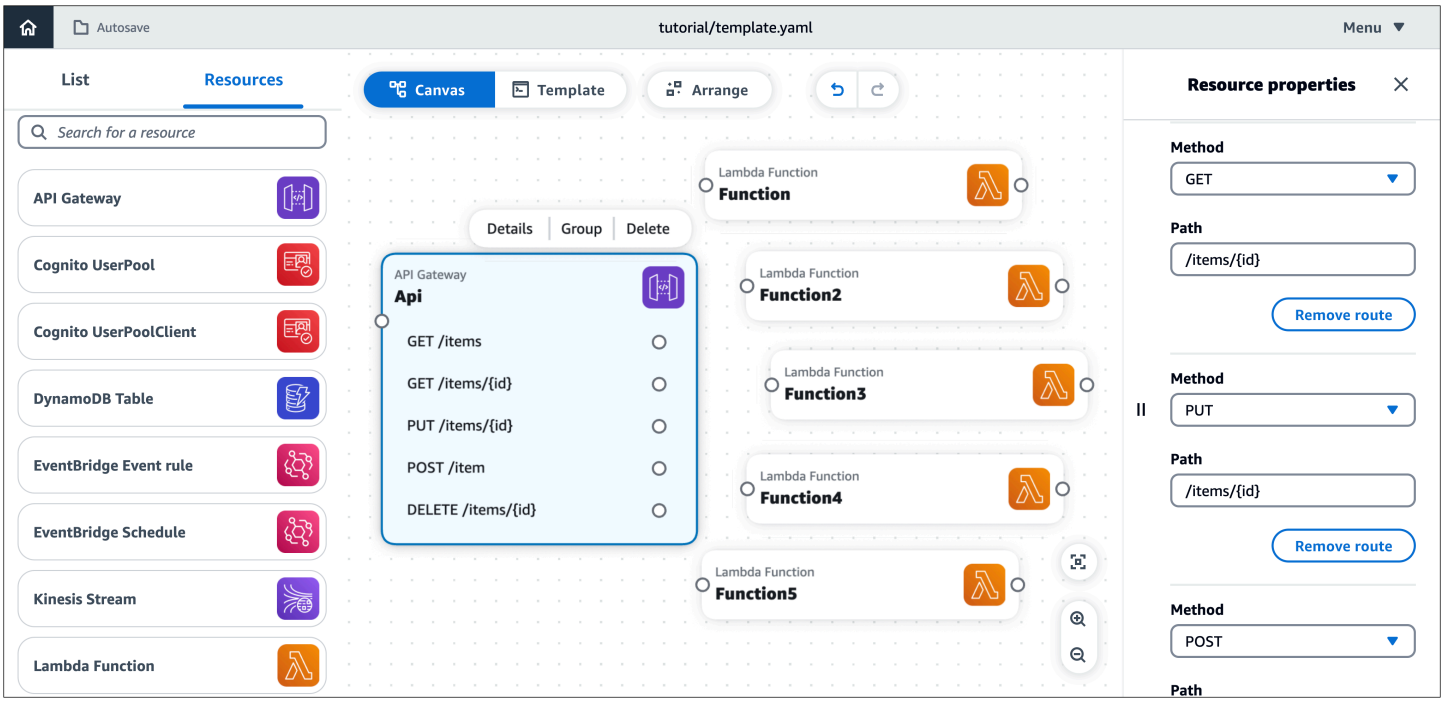
   > **Note**
   >
   > 對於下列每個路由，請使用資源屬性參考表中指定的 HTTP 方法和路徑值。

   a. 針對方法，選擇指定的 HTTP 方法。例如，GET。

   b. 針對路徑，輸入指定的路徑。例如：**`/items`**。

   c. 選擇 Add route (新增路由)。

   d. 重複上述步驟，直到您已新增所有五個指定的路由。
3. 選擇 Save (儲存)。

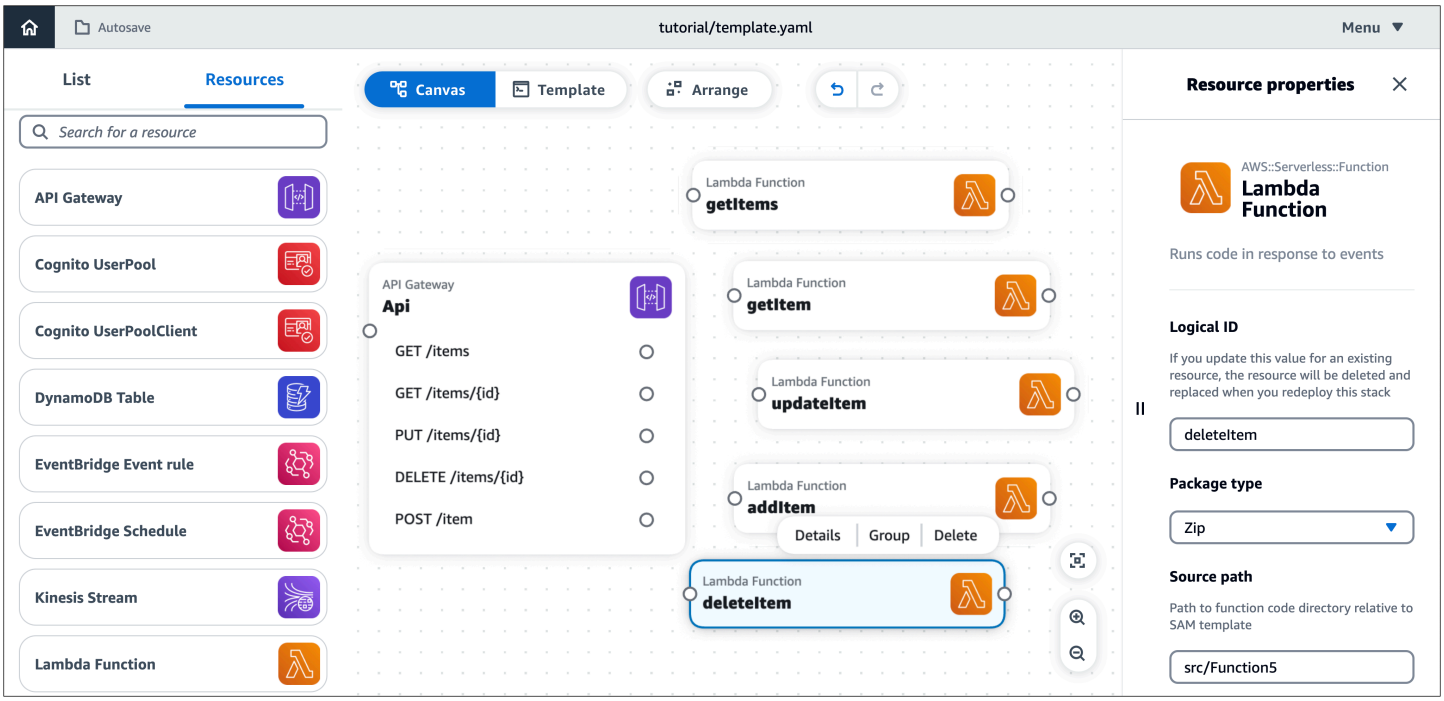


步驟 4：設定 Lambda 函數

依資源屬性參考資料表中指定的方式，命名五個 Lambda 函數。

命名 Lambda 函數

1. 開啟 Lambda 函數卡的資源屬性面板。若要開啟面板，請按兩下卡片。或者，選取卡片，然後選擇詳細資訊。
2. 在資源屬性面板中，針對邏輯 ID，輸入指定的函數名稱。例如：**`getItems`**。
3. 選擇 Save (儲存)。
4. 重複上述步驟，直到您已命名所有五個函數。

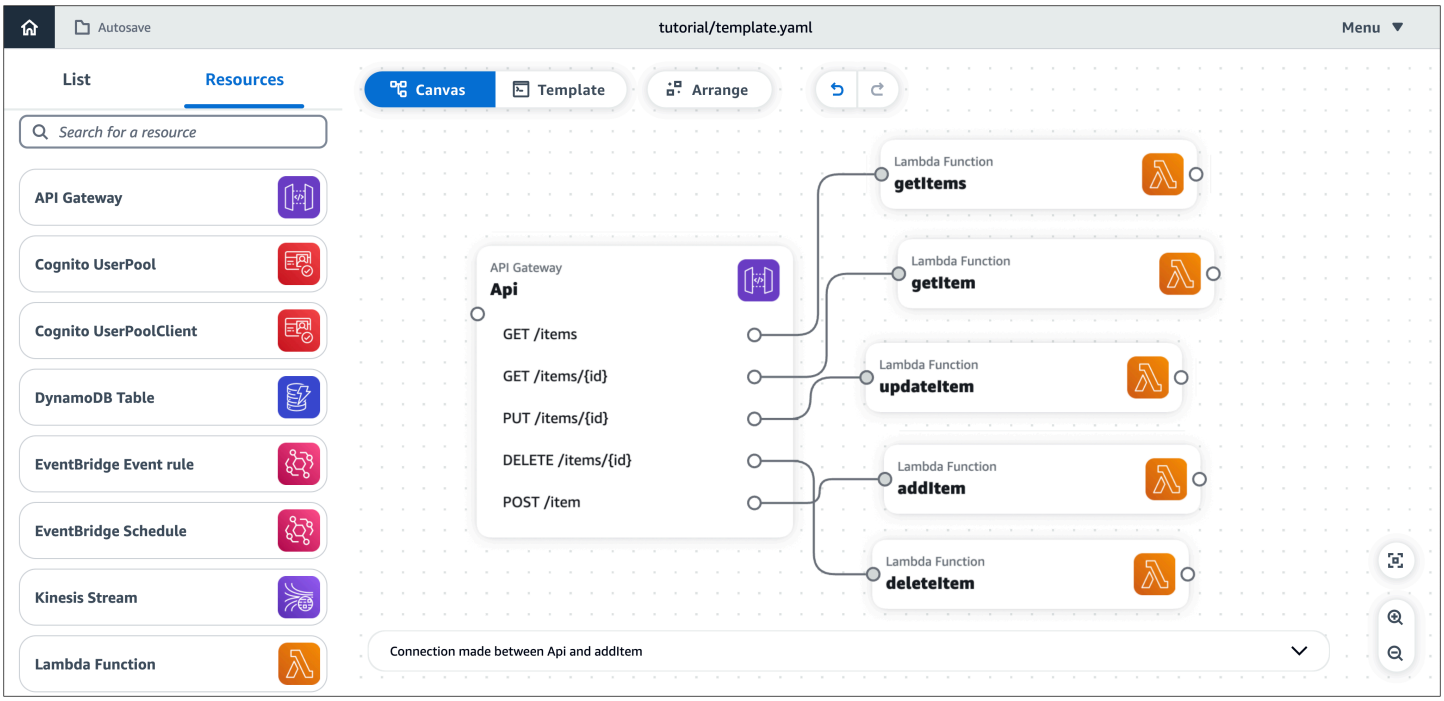


步驟 5：連接您的卡片

將 API Gateway 卡上的每個路由連接到其相關的 Lambda 函數卡，如資源屬性參考表所述。

連接您的卡片

1. 按一下 API Gateway 卡上的右側連接埠，並將其拖曳至指定 Lambda 函數卡的左側連接埠。例如，按一下 GET /items 連接埠，並將其拖曳至 getItems 的左側連接埠。
2. 重複上述步驟，直到您已將 API Gateway 卡上的全部五個路由連接到對應的 Lambda 函數卡。



步驟 6：整理畫布

將 Lambda 函數分組並排列所有卡片，以組織視覺化畫布。

將函數分組

1. 按住 Shift，然後在畫布上選取每個 Lambda 函數卡。
2. 選擇群組。

為您的群組命名

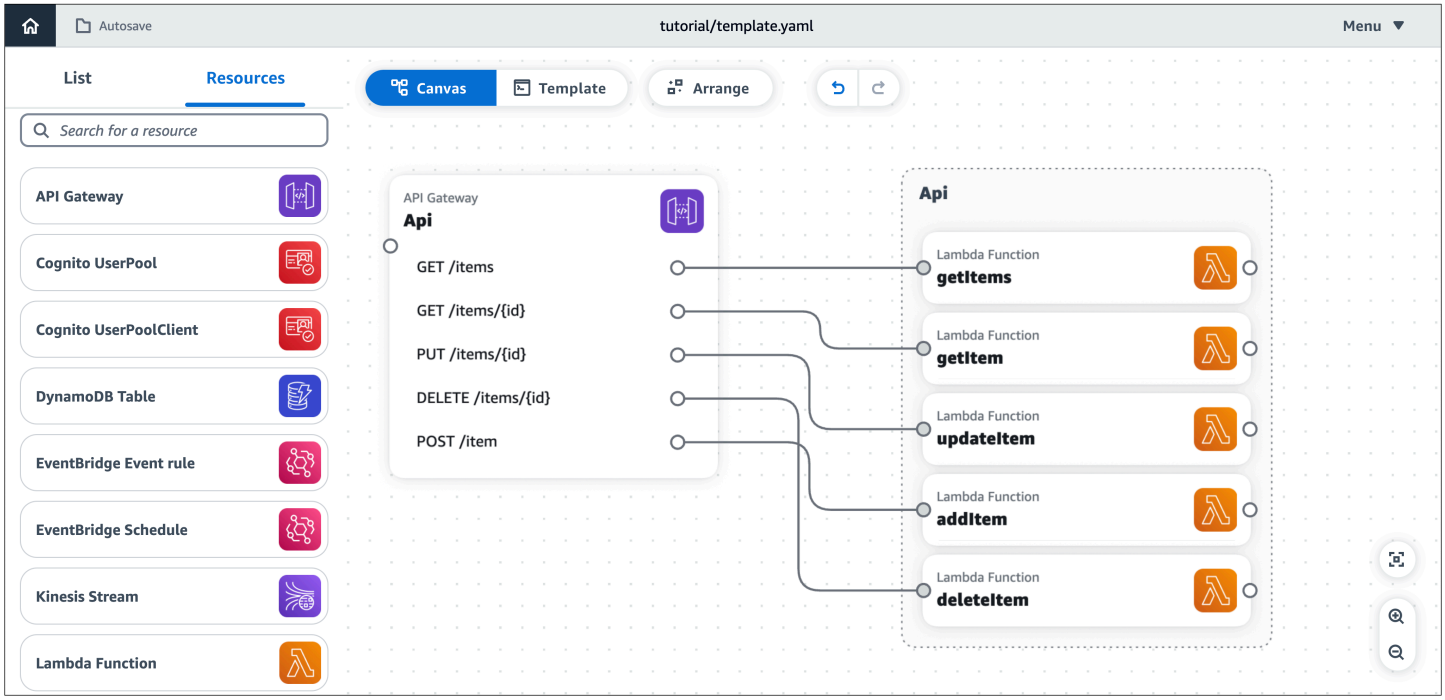
1. 按兩下群組名稱 (群組) 附近的群組頂端。

   群組屬性面板隨即開啟。
2. 在群組屬性面板的群組名稱中，輸入 **API**。
3. 選擇 Save (儲存)。

安排您的卡片

在畫布的主檢視區域上方，選擇安排。

Infrastructure Composer 會排列並對齊視覺化畫布上的所有卡片，包括您的新群組 (API)，如下所示：

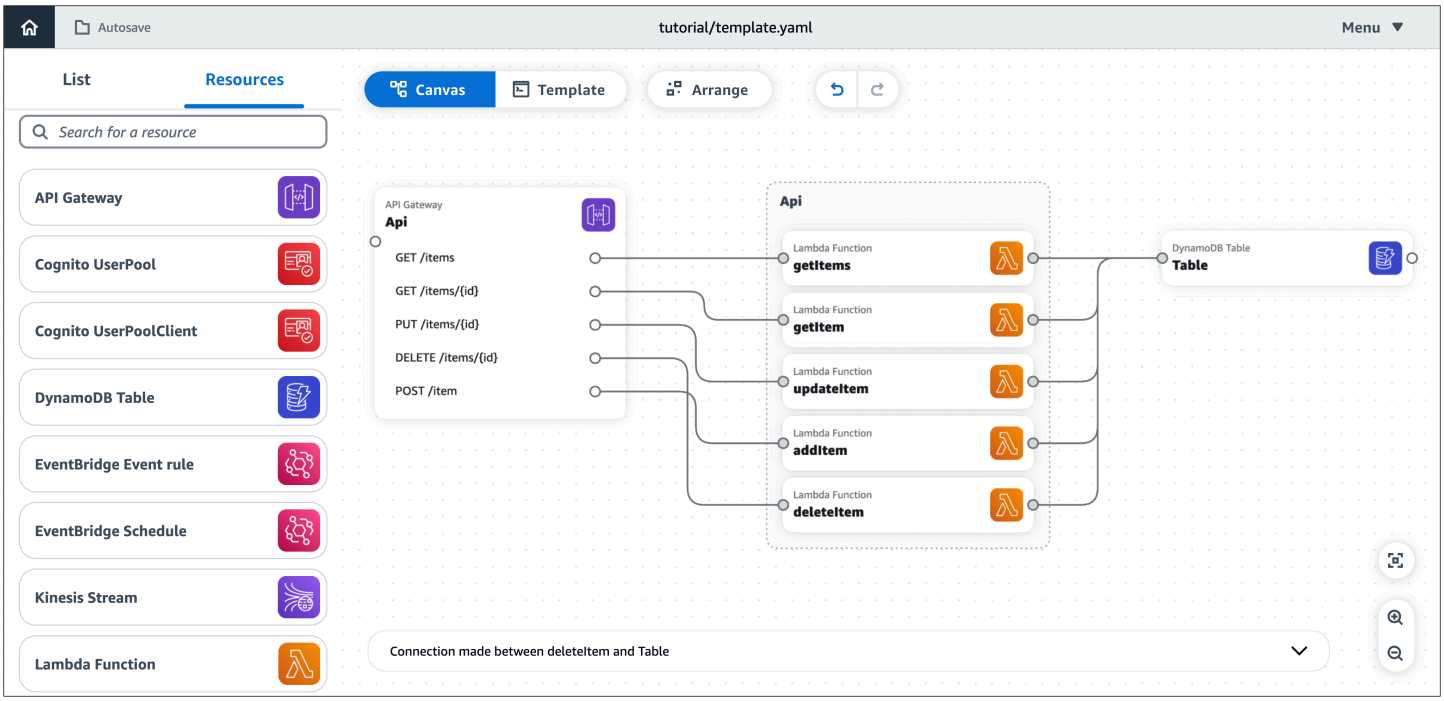


步驟 7：新增並連接 DynamoDB 資料表

現在，將 DynamoDB 資料表新增至您的應用程式架構，並將其連接至您的 Lambda 函數。

新增和連接 DynamoDB 資料表

1. 從資源調色盤 （資源) 的增強元件區段下方，將 DynamoDB 資料表卡拖曳到畫布上。
2. 按一下 Lambda 函數卡上的右側連接埠，並將其拖曳至 DynamoDB 資料表卡的左側連接埠。
3. 重複上一個步驟，直到您已將所有五個 Lambda 函數卡連接到 DynamoDB 資料表卡。
4. （選用） 若要重新組織和重新對齊畫布上的卡片，請選擇安排。

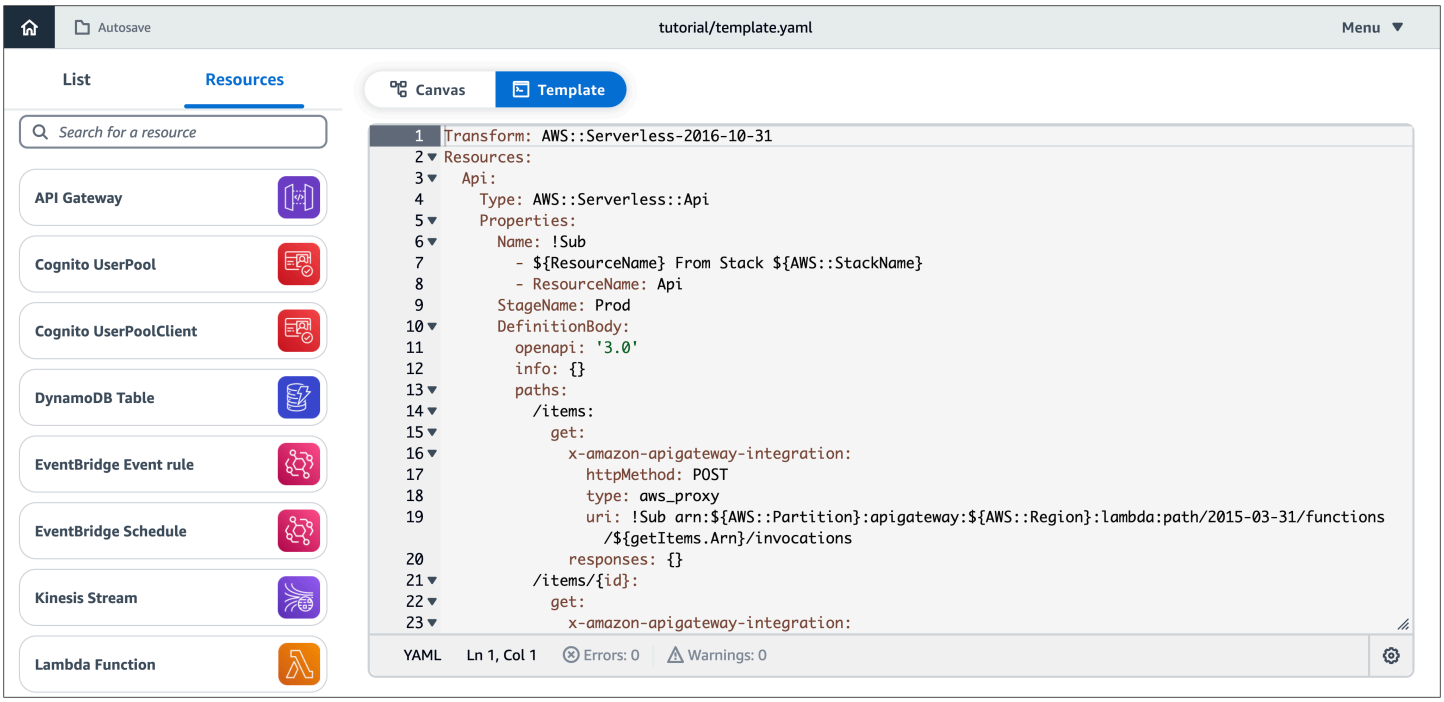


步驟 8：檢閱您的 AWS CloudFormation 範本

恭喜您！您已成功設計準備好部署的無伺服器應用程式。最後，選擇範本來檢閱 Infrastructure Composer 為您自動產生的 AWS CloudFormation 範本。

在 範本中，基礎設施撰寫者已定義下列項目：

- 此`Transform`宣告會將範本指定為 AWS Serverless Application Model (AWS SAM) 範本。如需詳細資訊，請參閱《 AWS Serverless Application Model 開發人員指南》中的AWS SAM 範本結構。
- `AWS::Serverless::Api` 資源，指定您的 API Gateway REST API 及其五個路由。
- 五個`AWS::Serverless::Function`資源，指定 Lambda 函數的組態，包括其環境變數和許可政策。
- `AWS::DynamoDB::Table` 資源，指定您的 DynamoDB 資料表及其屬性。
- `Metadata` 區段，其中包含資源群組 (API) 的相關資訊。如需本節的詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的中繼資料。



步驟 9：整合至您的開發工作流程

使用 Infrastructure Composer 建立的範本檔案和專案目錄，以進行進一步的測試和部署。

- 透過本機同步，您可以將 Infrastructure Composer 連接到本機機器上的 IDE，以加速開發。如需進一步了解，請參閱 將 Infrastructure Composer 主控台與本機 IDE 連線。
- 透過本機同步，您可以使用本機電腦上的 AWS Serverless Application Model 命令列界面 (AWS SAM CLI) 來測試和部署您的應用程式。如需進一步了解，請參閱 將您的 Infrastructure Composer 無伺服器應用程式部署至 AWS 雲端。

後續步驟

您現在可以使用 Infrastructure Composer 建置自己的應用程式。如需使用 Infrastructure Composer 的深入詳細資訊，請參閱 如何在 中編寫 AWS Infrastructure Composer。當您準備好部署應用程式時，請參閱 將您的 Infrastructure Composer 無伺服器應用程式部署至 AWS 雲端。

您可以在其中使用 Infrastructure Composer

您可以在 CloudFormation 主控台模式下，從其主控台、從 AWS Toolkit for Visual Studio Code和在 Infrastructure Composer 中使用 Infrastructure Composer。雖然每個使用案例都略有不同，但整體而言，它們都是類似的體驗。本節提供每個體驗的詳細資訊。

主題使用 AWS Infrastructure Composer 主控台是預設主控台體驗的完整概觀。主題CloudFormation 主控台模式提供與 AWS CloudFormation 堆疊工作流程整合之 Infrastructure Composer 版本的詳細資訊。 AWS Toolkit for Visual Studio Code提供在 VS Code 中存取和使用 Infrastructure Composer 的相關資訊。

主題

- 使用 AWS Infrastructure Composer 主控台
- 在 CloudFormation 主控台模式中使用 Infrastructure Composer
- 從 使用 Infrastructure Composer AWS Toolkit for Visual Studio Code

使用 AWS Infrastructure Composer 主控台

本節提供 AWS Infrastructure Composer 從 Infrastructure Composer 主控台存取和使用 的詳細資訊。這是 Infrastructure Composer 的預設體驗，也是熟悉 Infrastructure Composer 的好方法。您也可以將 Infrastructure Composer 主控台與本機 IDE 整合。如需詳細資訊，請參閱 將 Infrastructure Composer 主控台與本機 IDE 連線。

您也可以從 VS 程式碼中的 AWS Toolkit 存取 Infrastructure Composer，而且您可以使用專門設計用於的 Infrastructure Composer 模式 AWS CloudFormation。

如需使用 Infrastructure Composer 的一般文件，請參閱 如何編寫。

主題

- AWS Infrastructure Composer 主控台視覺效果概觀
- 從 Infrastructure Composer 主控台管理您的專案
- 將 Infrastructure Composer 主控台與本機 IDE 連線
- 允許網頁存取 Infrastructure Composer 中的本機檔案
- 在 Infrastructure Composer 主控台中本機同步和儲存您的專案

- 從 Lambda 主控台將函數匯入 Infrastructure Composer
- 匯出 Infrastructure Composer 視覺化畫布的影像

AWS Infrastructure Composer 主控台視覺效果概觀

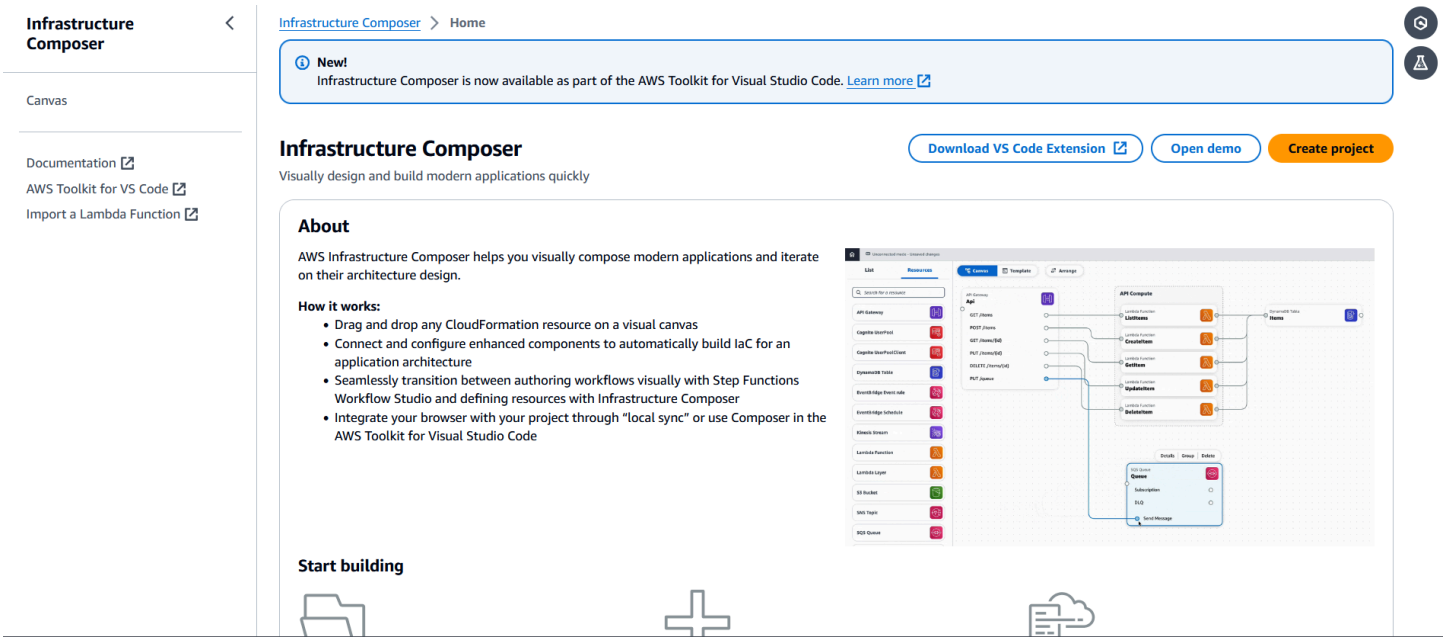
本節提供 AWS Infrastructure Composer 主控台的視覺化概觀。

主題

- 首頁
- 視覺化設計工具與視覺化畫布

首頁

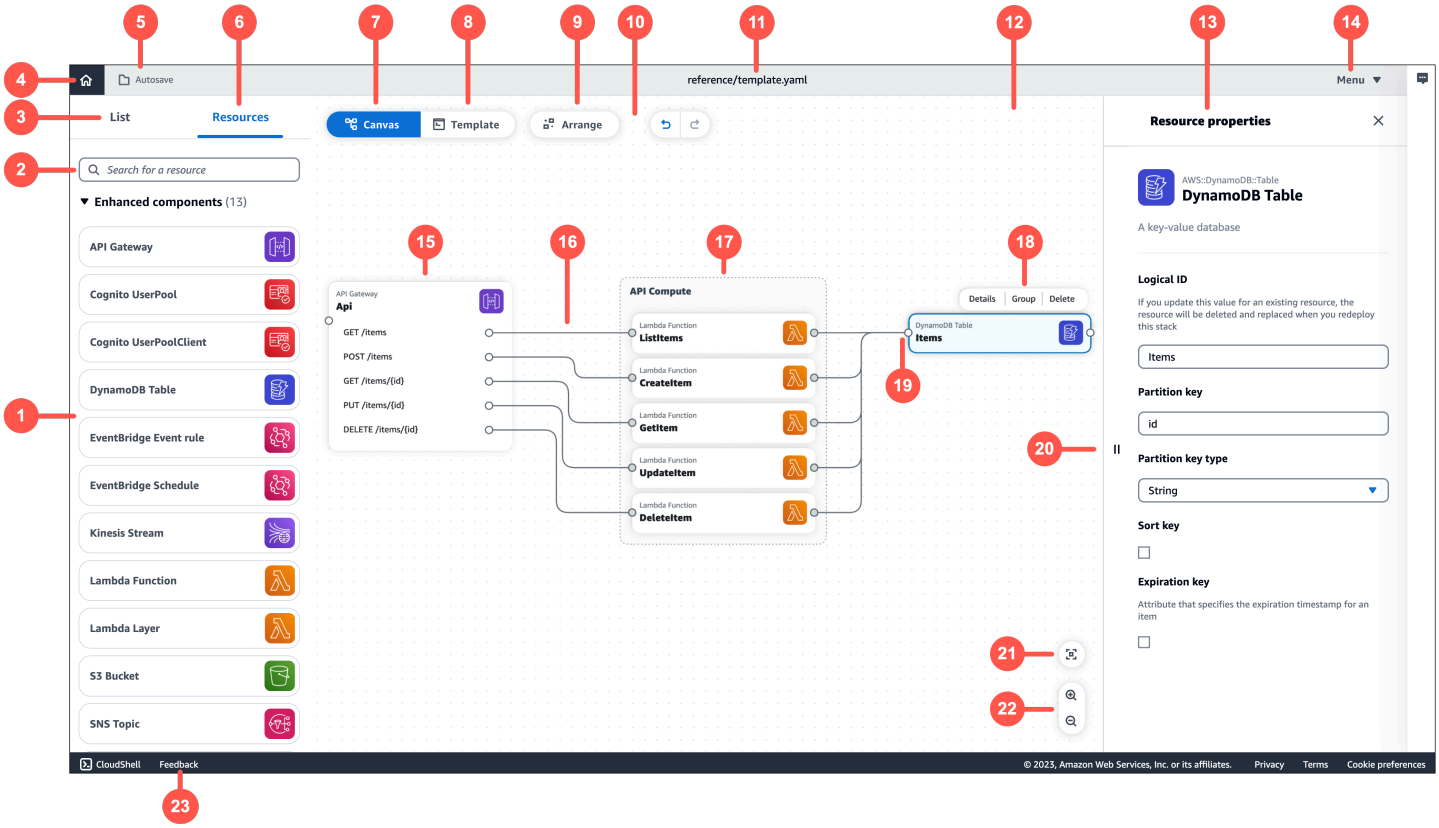
下圖是 Infrastructure Composer 主控台的首頁：



1. 文件 – 前往 Infrastructure Composer 文件。
2. Canvas – 前往畫布並建立或載入專案。
3. 示範 – 開啟 Infrastructure Composer 示範應用程式。
4. 建立專案 – 建立或載入專案。
5. 開始建置 – 快速連結以開始建置應用程式。
6. 意見回饋 – 前往此處提交意見回饋。

視覺化設計工具與視覺化畫布

下圖是 Infrastructure Composer 的視覺化設計工具和視覺化畫布：



1. 資源調色盤 – 顯示您可以使用的卡片。
2. 資源搜尋列 – 搜尋可新增至畫布的卡片。
3. 清單 – 顯示應用程式資源的樹狀檢視。
4. 首頁 – 選取此處前往 Infrastructure Composer 首頁。
5. 儲存狀態 – 指示是否將 Infrastructure Composer 變更儲存至本機機器。狀態包括：
   - 自動儲存 – 本機同步已啟用，您的專案正在自動同步並儲存。
   - 已儲存變更 – 您的應用程式範本會儲存至本機機器。
   - 未儲存的變更 – 您的應用程式範本有未儲存到本機機器的變更。
6. 資源 – 顯示資源調色盤。
7. Canvas – 在主檢視區域中顯示應用程式的畫布檢視。
8. 範本 – 在主檢視區域中顯示應用程式的範本檢視。
9. 排列 – 在畫布中排列應用程式架構。
10. 復原和重做 – 支援時執行復原和重做動作。

11範本名稱 – 指出您正在設計的範本名稱。

12主檢視區域 – 根據您的選擇顯示畫布或範本。

13資源屬性面板 – 顯示已在畫布中選取之卡片的相關屬性。此面板是動態的。顯示的屬性會隨著您設定卡片而變更。

14選單 – 提供一般選項，例如：

- 建立專案
- 開啟範本檔案或專案
- 儲存範本檔案
- 啟用本機同步
- 匯出畫布
- 取得支援
- 鍵盤快速鍵

15.卡片 – 在畫布上顯示卡片的檢視。

16Line – 代表卡片之間的連線。

17群組 – 將選取的卡片分組在一起以用於視覺化組織。

18.卡片動作 – 提供您可以對卡片採取的動作。

a. 詳細資訊 – 調出資源屬性面板。

b. 群組 – 將選取的卡片分組在一起。

c. 刪除 – 從畫布中刪除卡片。

19連接埠 – 連接至其他卡片。

20資源屬性欄位 – 為您的卡片設定的一組精選屬性欄位。

21重新置中 – 在視覺化畫布上重新置中應用程式圖表。

22縮放 – 在畫布上放大和縮小。

23意見回饋 – 前往此處提交意見回饋。

從 Infrastructure Composer 主控台管理您的專案

本主題提供從 Infrastructure Composer 主控台管理專案所執行的基本任務指導。這包括常見的任務，例如建立新專案、儲存專案，以及匯入專案或範本。如果您啟用本機同步模式，也可以載入現有的專案。啟用本機同步模式後，您可以執行下列動作：

- 建立新的專案，其中包含啟動範本和資料夾結構。

- 選擇包含您專案範本和檔案的父資料夾，以載入現有專案。
- 使用 Infrastructure Composer 來管理您的範本和資料夾

使用本機同步模式時， Infrastructure Composer 會自動將專案的範本和資料夾變更儲存至本機機器。如果您的瀏覽器不支援本機同步模式，或者如果您偏好使用基礎設施編譯器，但未啟用本機同步模式，您可以建立新的範本或載入現有的範本。若要儲存變更，您必須將範本匯出至本機機器。

> **Note**
>
> Infrastructure Composer 支援的應用程式包含下列項目：
>
> - 定義基礎設施程式碼的 AWS CloudFormation 或 AWS Serverless Application Model 範本。
> - 資料夾結構，可組織專案檔案，例如 Lambda 函數程式碼、組態檔案和建置資料夾。

主題

在 Infrastructure Composer 主控台中建立新專案

當您建立新的專案時， Infrastructure Composer 會產生啟動範本。當您在畫布上設計應用程式時，您的範本會遭到修改。若要儲存工作，您必須匯出範本或啟用本機同步模式。

建立新專案

1. 登入 Infrastructure Composer 主控台。
2. 在首頁上，選擇建立專案。

> **Note**
>
> 您也可以載入 Infrastructure Composer 中現有的 ，但您必須先啟用本機同步模式。啟用後，請參閱 載入已啟用本機同步的現有 Infrastructure Composer 專案以載入現有專案。

在 Infrastructure Composer 主控台中匯入現有的專案資料夾

使用本機同步模式，您可以匯入現有專案的父資料夾。如果您的專案包含多個範本，您可以選擇要載入的範本。

從首頁匯入現有專案

1. 登入 Infrastructure Composer 主控台。
2. 在首頁上，選擇載入 CloudFormation 範本。
3. 針對專案位置，選擇選取資料夾。選取專案的父資料夾，然後選擇選取。

> **Note**
>
> 如果您沒有收到此提示，您的瀏覽器可能不支援本機同步模式所需的檔案系統存取 API。如需詳細資訊，請參閱允許網頁存取 Infrastructure Composer 中的本機檔案。

4. 當您的瀏覽器提示時，選取檢視檔案。
5. 對於範本檔案，請從下拉式清單中選擇您的範本。如果您的專案包含單一範本，則 Infrastructure Composer 會自動為您選取它。
6. 選擇 Create (建立)。

從畫布匯入現有專案

1. 從畫布中，選擇選單以開啟選單。
2. 在開啟區段中，選擇專案資料夾。

> **Note**
>
> 如果專案資料夾選項無法使用，您的瀏覽器可能不支援本機同步模式所需的檔案系統存取 API。如需詳細資訊，請參閱允許網頁存取 Infrastructure Composer 中的本機檔案。

3. 針對專案位置，選擇選取資料夾。選取專案的父資料夾，然後選擇選取。
4. 當您的瀏覽器提示時，選取檢視檔案。
5. 對於範本檔案，請從下拉式清單中選擇您的範本。如果您的專案包含單一範本，則 Infrastructure Composer 會自動為您選取它。
6. 選擇 Create (建立)。

當您匯入現有的專案資料夾時， Infrastructure Composer 會啟用本機同步模式。對專案範本或檔案所做的變更會自動儲存至本機機器。

在 Infrastructure Composer 主控台中匯入現有的專案範本

當您匯入現有 AWS CloudFormation 或 AWS SAM 範本時， Infrastructure Composer 會自動在畫布上產生應用程式架構的視覺化效果。

您可以從本機機器匯入專案範本。

匯入現有的專案範本

1. 登入 Infrastructure Composer 主控台。
2. 選擇建立專案以開啟空白畫布。
3. 選擇選單以開啟選單。
4. 在開啟區段中，選擇範本檔案。
5. 選取您的範本，然後選擇開啟。

若要儲存範本的變更，您必須匯出範本或啟用本機同步模式。

在 Infrastructure Composer 主控台中儲存現有的專案範本

如果您不使用本機同步模式，則必須匯出範本以儲存變更。如果您已啟用本機同步模式，則不需要手動儲存範本。變更會自動儲存至您的本機機器。

儲存現有的專案範本

1. 從 Infrastructure Composer 畫布中，選擇選單以開啟選單。
2. 在儲存區段中，選擇儲存範本檔案。
3. 為您的範本提供名稱。
4. 選取要儲存範本的位置。
5. 選擇 Save (儲存)。

將 Infrastructure Composer 主控台與本機 IDE 連線

若要將 Infrastructure Composer 主控台與本機整合開發環境 (IDE) 連線，請使用本機同步模式。此模式會自動同步資料，並將資料儲存至本機機器。如需本機同步模式的詳細資訊，請參閱在

Infrastructure Composer 主控台中本機同步和儲存您的專案。如需使用本機同步模式的說明，請參閱在 Infrastructure Composer 主控台中本機同步和儲存您的專案。

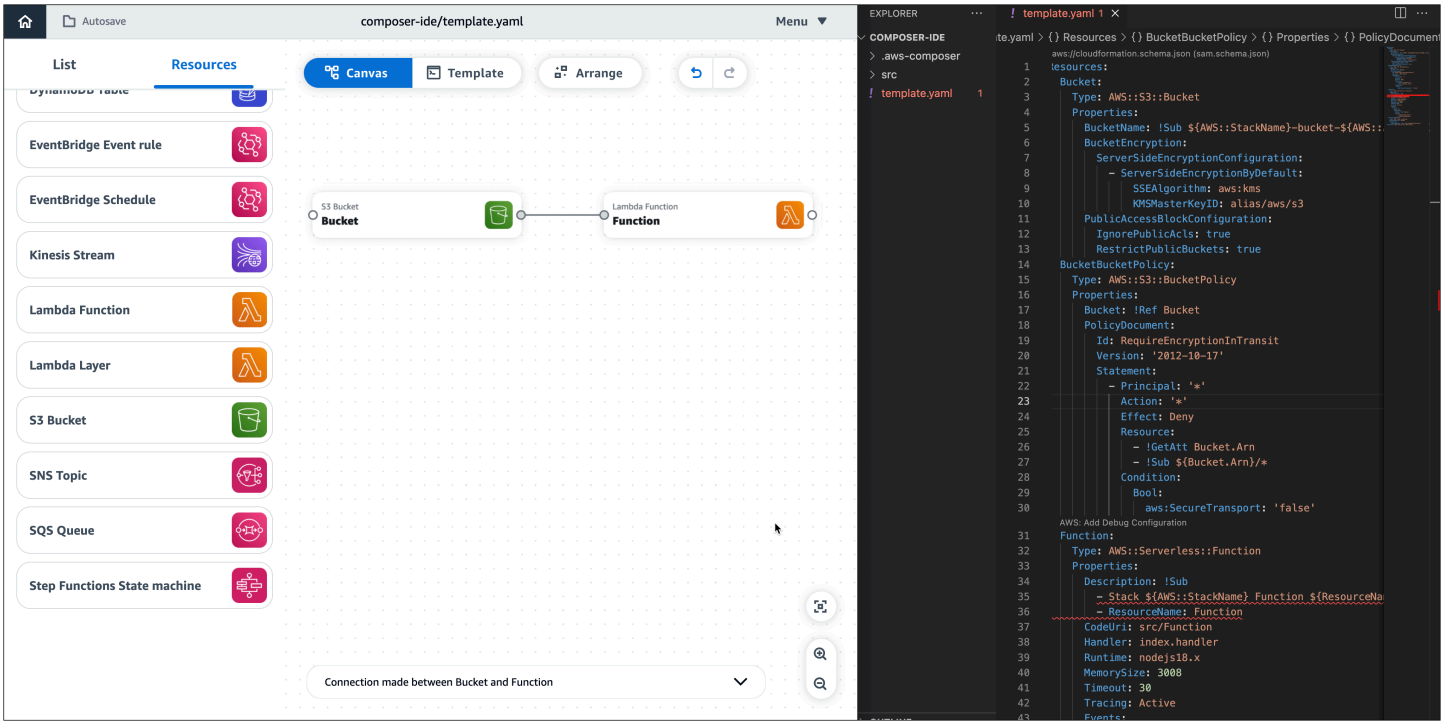
> **Note**
>
> 並非所有瀏覽器都提供啟用本機同步選項。可在 Google Chrome 和 Microsoft Edge 中使用。

將 Infrastructure Composer 與本機 IDE 搭配使用的優勢

當您在 Infrastructure Composer 中設計時，本機範本和專案目錄會自動同步並儲存。

您可以使用本機 IDE 來檢視變更和修改範本。您在本機所做的變更會自動同步至 Infrastructure Composer。

您可以使用 AWS Serverless Application Model 命令列界面 (AWS SAM CLI) 等本機工具來建置、測試、部署您的應用程式等。下列範例示範如何將資源拖放到 Infrastructure Composer 的視覺化畫布上，而畫布則會在本機 IDE 的 AWS SAM 範本中建立標記。



將 Infrastructure Composer 與本機 IDE 整合

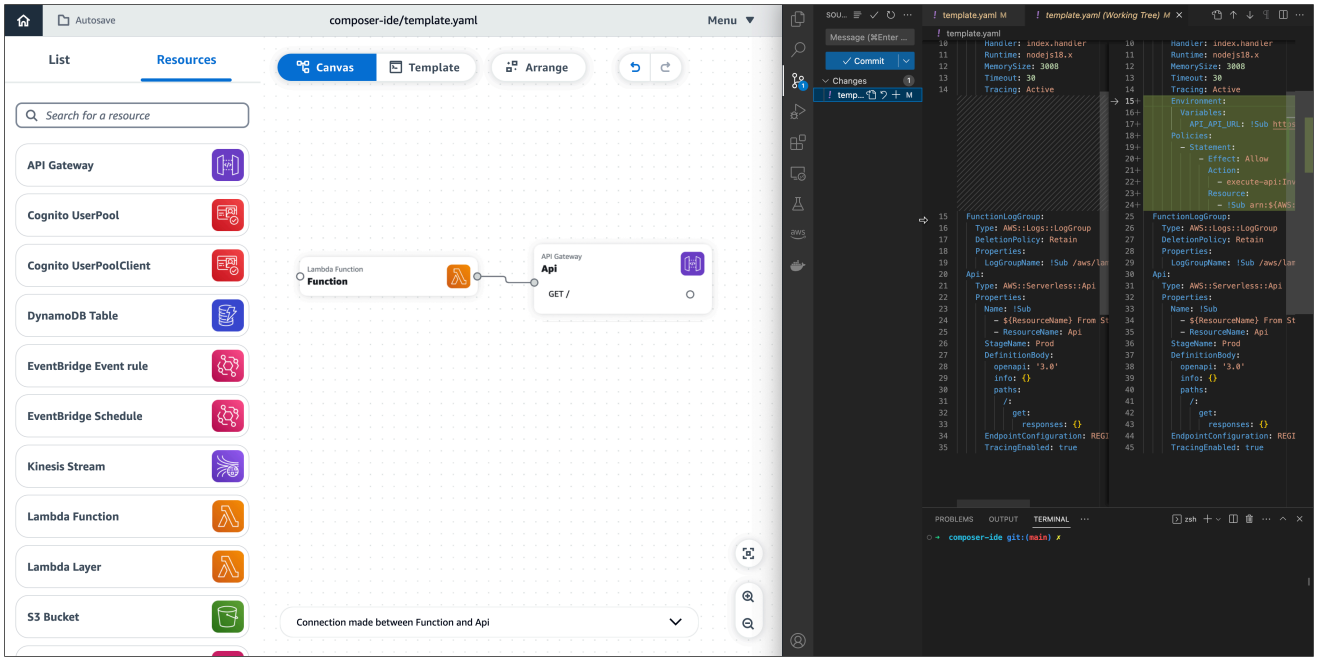
將 Infrastructure Composer 與本機 IDE 整合

1. 在 Infrastructure Composer 中，建立或載入專案，然後選取畫面右上角的選單按鈕，然後選擇啟用本機同步，以啟用本機同步。

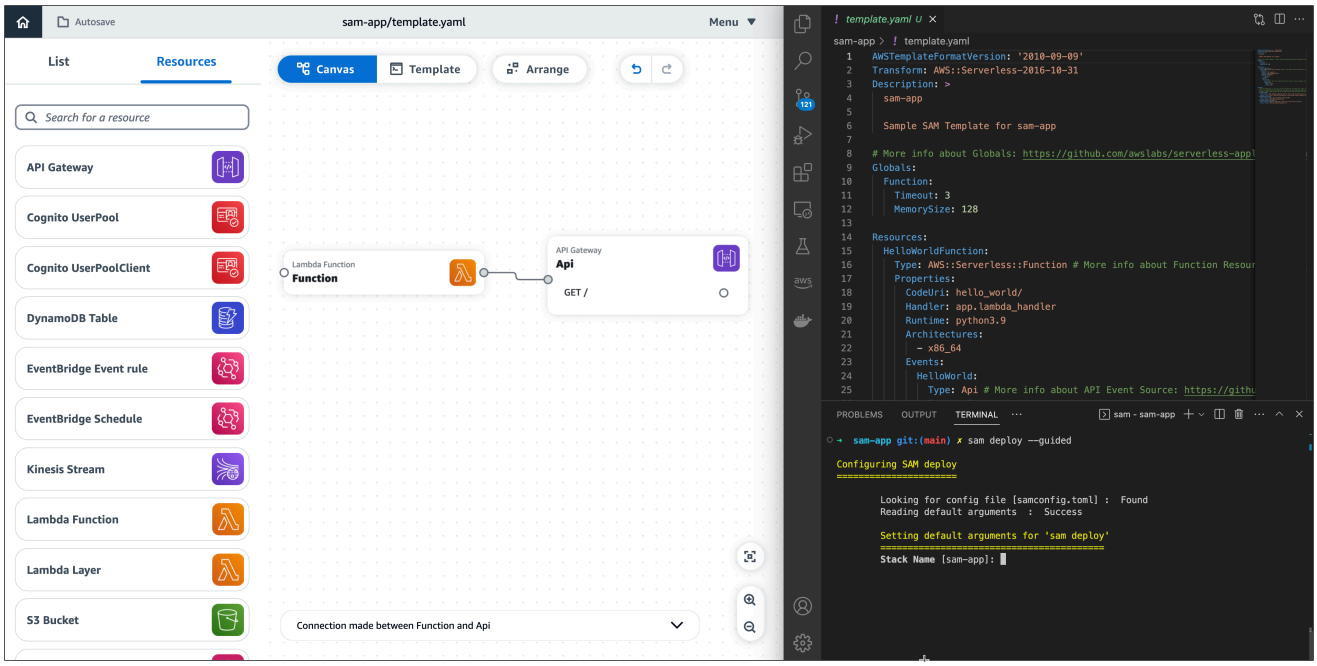
> **Note**
>
> 並非所有瀏覽器都提供啟用本機同步選項。可在 Google Chrome 和 Microsoft Edge 中使用。

2. 在本機 IDE 中，開啟與 Infrastructure Composer 相同的專案資料夾。
3. 將 Infrastructure Composer 與本機 IDE 搭配使用。在 Infrastructure Composer 中進行的更新會自動與本機機器同步。以下是您可以執行的一些動作範例：

   a. 使用您選擇的版本控制系統來追蹤 Infrastructure Composer 正在執行的更新。



   b. 在本機使用 AWS SAM CLI 來建置、測試、部署您的應用程式等。如需進一步了解，請參閱[將您的 Infrastructure Composer 無伺服器應用程式部署至 AWS 雲端]。



允許網頁存取 Infrastructure Composer 中的本機檔案

Infrastructure Composer 主控台支援本機同步模式，以及從 Lambda 主控台匯入函數。若要使用這些功能，需要支援檔案系統存取 API 的 Web 瀏覽器。任何 Google Chrome 和 Microsoft Edge 的最新版本都支援檔案系統存取 API 的所有功能，並且可以在 Infrastructure Composer 中與本機同步模式搭配使用。

檔案系統存取 API 可讓網頁存取您的本機檔案系統，以便讀取、寫入或儲存檔案。此功能預設為關閉，需要透過視覺提示來允許。授予後，此存取權會在您網頁的瀏覽器工作階段期間保留。

若要進一步了解檔案系統存取 API，請參閱：

- mdn Web 文件中的檔案系統存取 API。
- 檔案系統存取 API：簡化對 web.dev 網站中本機檔案的存取。

本機同步模式

當您在 Infrastructure Composer 中設計時，本機同步模式可讓您在本機自動同步和儲存範本檔案和專案資料夾。若要使用此功能，需要支援檔案系統存取 API 的 Web 瀏覽器。

Data Infrastructure Composer 存取

Infrastructure Composer 會取得您允許之專案資料夾的讀取和寫入存取權，以及該專案資料夾的任何子資料夾。此存取權用於建立、更新和儲存您設計時產生的任何範本檔案、專案資料夾和備份目錄。Infrastructure Composer 存取的資料不會用於任何其他用途，也不會存放在本機檔案系統以外的任何位置。

存取敏感資料

檔案系統存取 API 排除或限制存取可能包含敏感資料的特定目錄。如果您選取其中一個要與 Infrastructure Composer 本機同步模式搭配使用的目錄，將會發生錯誤。您可以選擇另一個本機目錄來與 連線，或在停用本機同步的預設模式中使用 Infrastructure Composer。

如需詳細資訊，包括敏感目錄的範例，請參閱 檔案系統存取 W3C 草稿社群群組報告中，授予存取更多或更敏感檔案的使用者。 W3C

如果您使用 Windows Subsystem for Linux (WSL)，檔案系統存取 API 會因為其位於您的Windows系統內，而排除對整個Linux目錄的存取。您可以在停用本機同步的情況下使用 Infrastructure Composer，或設定解決方案，將專案檔案從 WSL目錄同步至 中的工作目錄Windows。然後，將 Infrastructure Composer 本機同步模式與Windows目錄搭配使用。

在 Infrastructure Composer 主控台中本機同步和儲存您的專案

本節提供使用 Infrastructure Composer 的本機同步模式，自動同步並將專案儲存至本機機器的相關資訊。

基於下列原因，建議您使用本機同步：

您可以為新專案啟用本機同步，或載入已啟用本機同步的現有專案。

- 根據預設，您需要在設計時手動儲存應用程式範本。當您進行變更時，請使用本機同步，自動將應用程式範本儲存至本機機器。
- 本機同步會管理專案資料夾、備份資料夾和支援的外部檔案，並自動同步至本機機器。
- 使用本機同步時，您可以將 Infrastructure Composer 與本機 IDE 連線，以加速開發。如需進一步了解，請參閱 將 Infrastructure Composer 主控台與本機 IDE 連線。

本機同步模式儲存的內容

本機同步模式會自動同步並將下列項目儲存至本機機器：

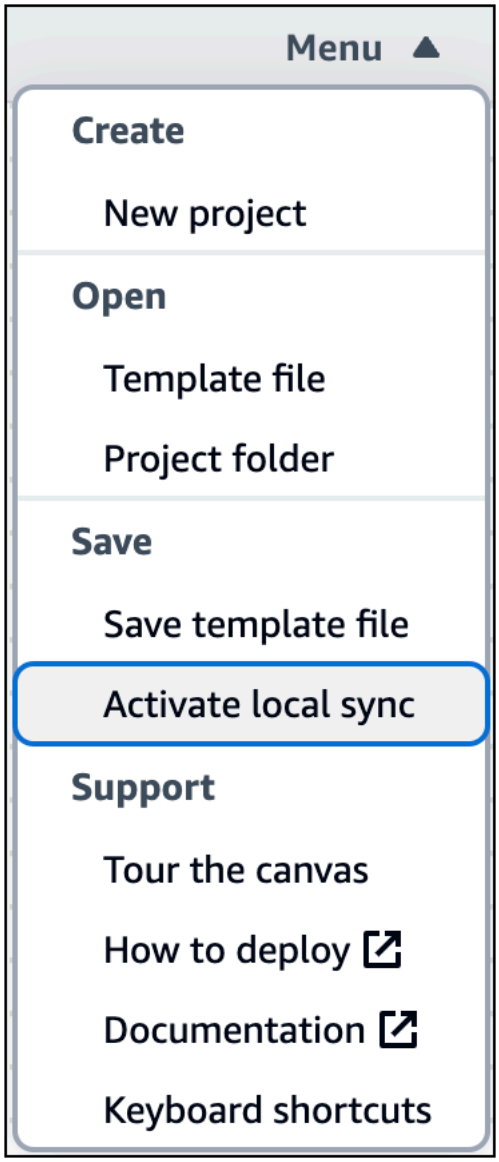
- 應用程式範本檔案 – 包含基礎設施的 AWS CloudFormation 或 AWS Serverless Application Model (AWS SAM) 範本，做為程式碼 (IaC)。
- 專案資料夾 – 一般目錄結構，可組織您的 AWS Lambda 函數。
- 備份目錄 – 名為 的備份目錄`.aws-composer`，建立於專案位置的根目錄。此目錄包含應用程式範本檔案和專案資料夾的備份副本。
- 外部檔案 – 支援的外部檔案，您可以在 Infrastructure Composer 中使用。如需進一步了解，請參閱參考 Infrastructure Composer 中的外部檔案。

瀏覽器要求

本機同步模式需要支援檔案系統存取 API 的瀏覽器。如需詳細資訊，請參閱允許網頁存取 Infrastructure Composer 中的本機檔案。

啟用本機同步模式

本機同步模式預設為停用。您可以透過 Infrastructure Composer 選單啟用本機同步模式。



如需啟用本機同步和現有載入專案的指示，請參閱下列主題：

- 在 Infrastructure Composer 中啟用本機同步
- 載入已啟用本機同步的現有 Infrastructure Composer 專案

在 Infrastructure Composer 中啟用本機同步

若要啟用本機同步，請完成下列步驟：

1. 在基礎設施撰寫者首頁中，選取建立專案。
2. 從基礎設施編寫器功能表中，選取啟用本機同步。

3. 針對專案位置，按下選取資料夾，然後選擇目錄。這是 Infrastructure Composer 會在您設計時儲存和同步範本檔案和資料夾的地方。

> **Note**
>
> 專案位置不得包含現有的應用程式範本。

4. 出現允許存取的提示時，請選取檢視檔案。
5. 按下啟用。出現儲存變更的提示時，請選取儲存變更。

   啟用時，Autosave 指標會顯示在畫布的左上區域。

載入已啟用本機同步的現有 Infrastructure Composer 專案

若要載入已啟用本機同步的現有專案，請完成下列步驟：

1. 在基礎設施撰寫者首頁中，選取載入 AWS CloudFormation 範本。
2. 從基礎設施編寫器功能表中，選取開啟 > 專案資料夾。
3. 針對專案位置，按下選取資料夾，然後選擇專案的根資料夾。
4. 出現允許存取的提示時，請選取檢視檔案。
5. 對於範本檔案，選取您的應用程式範本，然後按建立。
6. 出現儲存變更的提示時，請選取儲存變更。

   啟用時，Autosave 指標會顯示在畫布的左上區域。

從 Lambda 主控台將函數匯入 Infrastructure Composer

Infrastructure Composer 提供與 AWS Lambda 主控台的整合。您可以將 Lambda 函數從 Lambda 主控台匯入 Infrastructure Composer 主控台。然後，使用 Infrastructure Composer 畫布進一步設計您的應用程式架構。

- 此整合需要支援檔案系統存取 API 的瀏覽器。如需詳細資訊，請參閱允許網頁存取 Infrastructure Composer 中的本機檔案。
- 當您將 Lambda 函數匯入 Infrastructure Composer 時，您必須啟用本機同步模式以儲存任何變更。如需詳細資訊，請參閱在 Infrastructure Composer 主控台中本機同步和儲存您的專案。

若要開始使用此整合，請參閱《 AWS Lambda 開發人員指南》中的使用 AWS Lambda 搭配 AWS Infrastructure Composer 。

匯出 Infrastructure Composer 視覺化畫布的影像

本主題說明 AWS Infrastructure Composer 主控台匯出畫布功能。

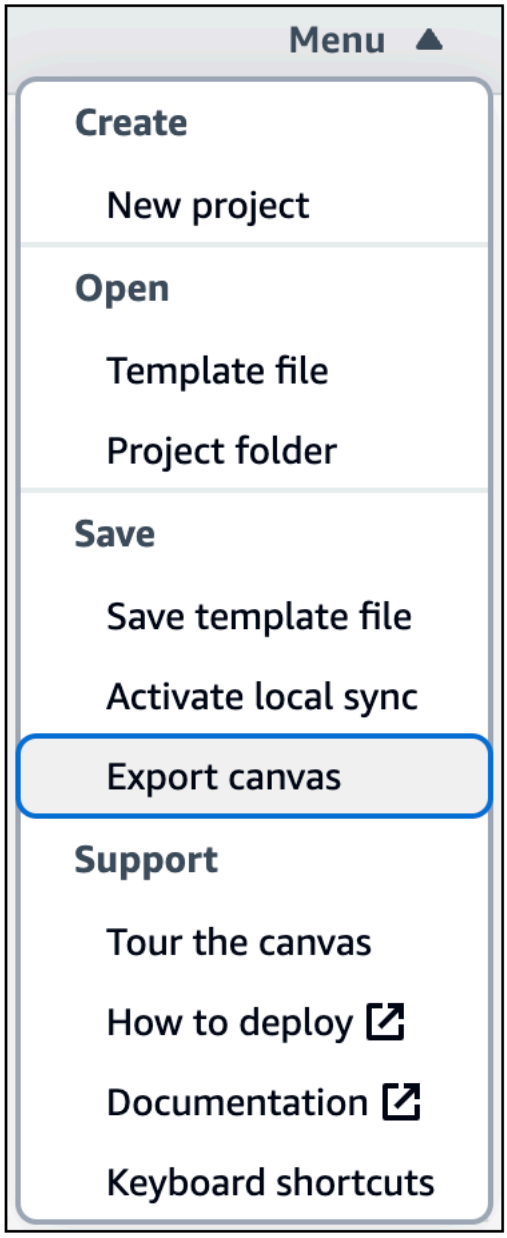
如需所有 Infrastructure Composer 功能的視覺化概觀，請參閱AWS Infrastructure Composer 主控台視覺效果概觀。

關於匯出畫布

匯出畫布功能會將您應用程式的畫布匯出為本機機器的影像。

- Infrastructure Composer 會移除視覺化設計工具 UI 元素，並僅匯出應用程式的圖表。
- 預設的影像檔案格式為 png。
- 檔案會匯出至本機電腦的預設下載位置。

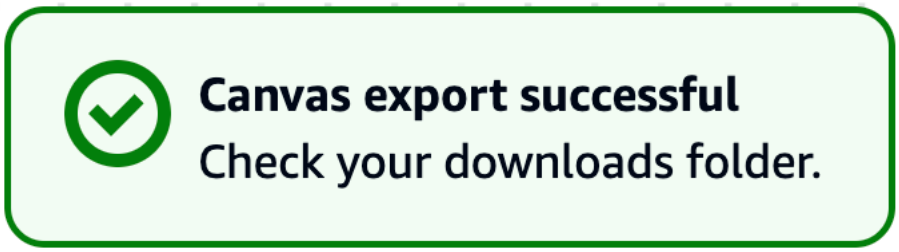
您可以從選單存取匯出畫布功能。



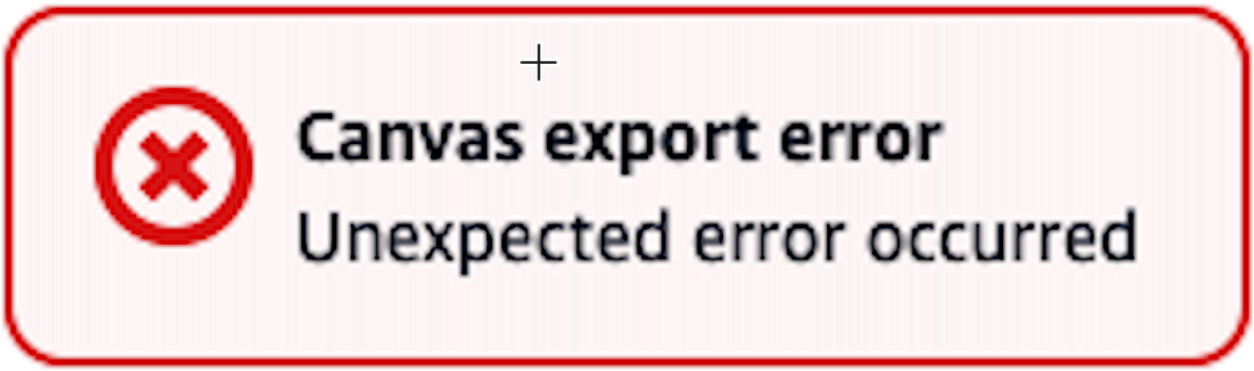
匯出畫布

匯出畫布時，基礎設施編寫器會顯示狀態訊息。

如果匯出成功，您會看到下列訊息：



如果匯出失敗，您會看到錯誤訊息。如果您收到錯誤，請嘗試再次匯出。



在 CloudFormation 主控台模式中使用 Infrastructure Composer

CloudFormation 主控台模式中的 Infrastructure Composer 是視覺化 AWS CloudFormation 範本的建議工具。您也可以使用此工具來建立和編輯 AWS CloudFormation 範本。

此模式與 Infrastructure Composer 主控台有何不同？

CloudFormation 主控台模式中的 Infrastructure Composer 通常具有與預設 Infrastructure Composer 主控台相同的功能，但需要注意一些差異。

- 此模式與 AWS CloudFormation 主控台中的堆疊工作流程整合。這可讓您直接使用 Infrastructure Composer AWS CloudFormation。
- 在 Infrastructure Composer 主控台中本機同步和儲存您的專案不支援自動同步資料並將其儲存至本機電腦的功能。
- Lambda 相關卡 (Lambda 函數和 Lambda Layer) 需要此模式中無法使用的程式碼建置和封裝解決方案。

> **Note**
>
> 這些卡和本機同步可用於 Infrastructure Composer 主控台或 AWS Toolkit for Visual Studio Code。

當您從 AWS CloudFormation 主控台開啟 Infrastructure Composer 時，基礎設施 Composer 會在 CloudFormation 主控台模式中開啟。在此模式中，您可以使用 Infrastructure Composer 來視覺化、建立和更新範本。

如何在 CloudFormation 主控台模式中存取 Infrastructure Composer

CloudFormation 主控台模式中的 Infrastructure Composer 是從 AWS CloudFormation 設計工具升級。我們建議您使用 Infrastructure Composer 來視覺化您的 AWS CloudFormation 範本。您也可以使用此工具來建立和編輯 AWS CloudFormation 範本。

1. 前往 Cloudformation 主控台並登入。
2. 從左側導覽功能表中選取 Infrastructure Composer。這將帶您在 CloudFormation 主控台模式中使用 Infrastructure Composer。

> **Note**
>
> 如需在 CloudFormation 主控台模式中使用 Infrastructure Composer 的資訊，請參閱 在 CloudFormation 主控台模式中使用 Infrastructure Composer。

在 CloudFormation 主控台模式下，視覺化 Infrastructure Composer 中的部署

遵循本主題中的指示，以視覺化已部署的 AWS CloudFormation 堆疊/基礎設施編譯器範本。

1. 前往 AWS CloudFormation 主控台並登入。
2. 選取您要編輯的堆疊。
3. 選取範本索引標籤。
4. 選取 Infrastructure Composer。

Infrastructure Composer 將視覺化您的堆疊/範本。您也可以在這裡進行變更。

在 CloudFormation 主控台模式下的 Infrastructure Composer 中建立新的範本

請遵循本主題中的指示建立新的範本。

1. 前往 AWS CloudFormation 主控台並登入。
2. 從左側導覽功能表中選取 Infrastructure Composer。這將在 CloudFormation 主控台模式中開啟 Infrastructure Composer。

3. 從資源面板拖放、設定和連接您需要的資源 (卡片)。

> **Note**
>
> 如何編寫 如需使用 Infrastructure Composer 的詳細資訊，請參閱 ，並注意 Lambda 相關卡 (Lambda 函數和 Lambda Layer) 需要 CloudFormation 主控台模式中 Infrastructure Composer 中無法使用的程式碼建置和封裝解決方案。這些卡可用於 Infrastructure Composer 主控台或 AWS Toolkit for Visual Studio Code。如需使用這些工具的資訊，請參閱 您可以在其中使用 Infrastructure Composer。

4. 按兩下卡片以使用資源屬性面板來指定卡片的設定方式。
5. 連接您的卡片以指定應用程式的事件驅動工作流程。
6. 選取範本以檢視和編輯您的基礎設施程式碼。變更會自動與您的畫布檢視同步。
7. 您的範本準備好匯出至堆疊後，請選取建立範本。
8. 選取確認並匯出至 CloudFormation 按鈕。這將帶您返回建立堆疊工作流程，並顯示確認範本已成功匯入的訊息。

> **Note**
>
> 只能匯出其中具有資源的範本。

9. 在建立堆疊工作流程中，選取下一步。
10. 提供堆疊名稱、檢閱任何列出的參數，然後選取下一步。

> **Note**
>
> 堆疊名稱必須以字母開頭，且只包含字母、數字、破折號。

11. 提供下列資訊後，請選取下一步：

    - 與堆疊相關聯的標籤
    - 堆疊許可
    - 堆疊的失敗選項

> **Note**
>
> 如需管理堆疊的指引，請參閱 AWS CloudFormation 使用者指南中的AWS CloudFormation 最佳實務。

12. 確認您的堆疊詳細資訊正確、檢查頁面底部的確認，然後選取提交按鈕。

AWS CloudFormation 會根據範本中的資料開始建立堆疊。

在 CloudFormation 主控台模式下更新 Infrastructure Composer 中的現有堆疊

請遵循本主題中的指示來更新現有的 AWS CloudFormation 堆疊。

> **Note**
>
> 如果您的檔案儲存在本機，我們建議您使用 AWS Toolkit for Visual Studio Code。

1. 前往 AWS CloudFormation 主控台並登入。
2. 選取您要編輯的堆疊。
3. 選取更新按鈕。這樣做會帶您前往更新堆疊精靈。
4. 在右側，選取 Infrastructure Composer 中的編輯。
5. 選取下方標示為在 Infrastructure Composer 中編輯的按鈕。這將帶您在 CloudFormation 主控台模式中使用 Infrastructure Composer。
6. 在這裡，您可以從資源面板拖放、設定和連接資源 (卡片)。

> **Note**
>
> 如何編寫 如需使用 Infrastructure Composer 的詳細資訊，請參閱 ，並注意 Lambda 相關卡 (Lambda 函數和 Lambda Layer) 需要 CloudFormation 主控台模式中 Infrastructure Composer 中無法使用的程式碼建置和封裝解決方案。這些卡可用於 Infrastructure Composer 主控台或 AWS Toolkit for Visual Studio Code。如需使用這些工具的資訊，請參閱 您可以在其中使用 Infrastructure Composer。

7. 當您準備好將變更匯出至 時 AWS CloudFormation，請選取更新範本。
8. 選取確認並繼續前往 CloudFormation。這將帶您返回更新堆疊工作流程，並顯示確認範本已成功匯入的訊息。

> **Note**
>
> 只能匯出其中具有資源的範本。

9. 在更新堆疊工作流程中，選取下一步。
10. 檢閱任何列出的參數，然後選取下一步。
11. 提供下列資訊後，請選取下一步：

    - 與堆疊相關聯的標籤
    - 堆疊許可
    - 堆疊的失敗選項

> **Note**
>
> 如需管理堆疊的指引，請參閱 AWS CloudFormation 使用者指南中的AWS CloudFormation 最佳實務。

12. 確認您的堆疊詳細資訊正確、檢查頁面底部的確認，然後選取提交按鈕。

AWS CloudFormation 會根據您在範本中所做的更新開始更新堆疊。

從 使用 Infrastructure Composer AWS Toolkit for Visual Studio Code

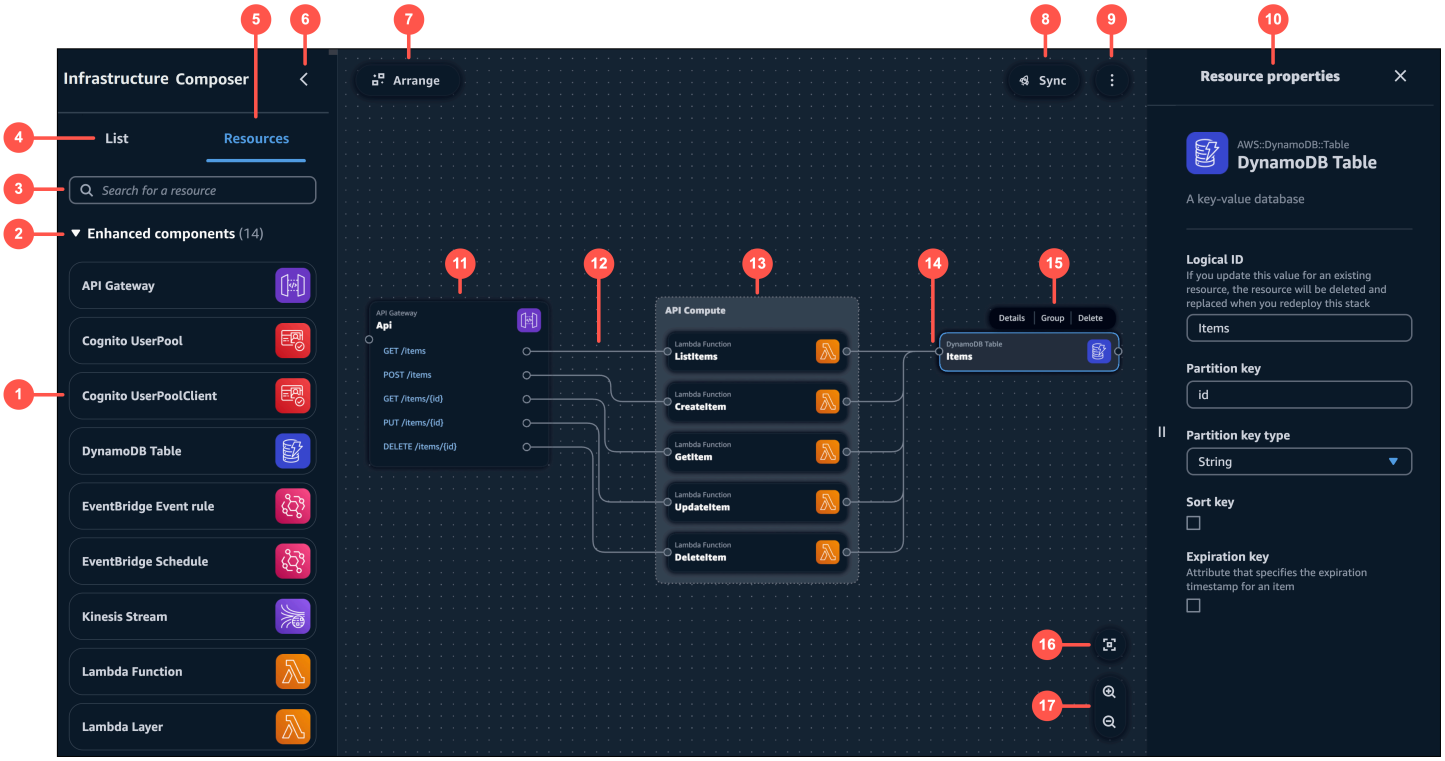
本節說明如何 AWS Infrastructure Composer 從 使用 AWS Toolkit for Visual Studio Code。這包括來自 的 Infrastructure Composer 視覺化概觀 AWS Toolkit for Visual Studio Code。其中也包含說明如何存取此體驗，以及將專案從 VS Code 同步至 AWS 雲端的指示。若要同步，請使用 中的 sam sync命令 AWS SAMCLI。本節也提供從 在 Infrastructure Composer Amazon Q中使用 的指引 AWS Toolkit for Visual Studio Code。

如需從 使用 Infrastructure Composer 的其他指引 AWS Toolkit for Visual Studio Code，請參閱 如何編寫。本節的內容適用於此體驗，以及 Infrastructure Composer 主控台體驗。

主題

的 Infrastructure Composer 視覺化概觀 AWS Toolkit for Visual Studio Code

中的 Infrastructure Composer 視覺效果設計工具 AWS Toolkit for Visual Studio Code 包含視覺效果畫布，其中包括下列影像編號和下列元件。



1. 資源調色盤 – 顯示您可以使用的卡片。
2. 卡片類別 – 卡片會依 Infrastructure Composer 獨有的類別進行組織。
3. 資源搜尋列 – 搜尋可新增至畫布的卡片。
4. 清單 – 顯示應用程式資源的樹狀檢視。
5. 資源 – 顯示資源調色盤。
6. 左窗格切換 – 隱藏或顯示左窗格。
7. 排列 – 在畫布中排列應用程式架構。

8. 同步 – 啟動 AWS Serverless Application Model (AWS SAM) CLI`sam sync`命令來部署您的應用程式。
9. 選單 – 提供一般選項，例如：
   - 匯出畫布
   - 導覽畫布
   - 文件的連結
   - 鍵盤快速鍵
10. 資源屬性面板 – 顯示已在畫布中選取之卡片的相關屬性。此面板是動態的。顯示的屬性會隨著您設定卡片而變更。
11. 卡片 – 在畫布上顯示卡片的檢視。
12. Line – 代表卡片之間的連線。
13. 群組 – 一組卡片。您可以分組視覺化組織的卡片。
14. 連接埠 – 連接至其他卡片。
15. 卡片動作 – 提供您可以對卡片採取的動作。
    - 詳細資訊 – 調出資源屬性面板。
    - 群組 – 將選取的卡片分組在一起。
    - 刪除 – 從您的畫布和範本中刪除卡片。
16. 重新置中 – 在視覺化畫布上重新置中應用程式圖表。
17. 縮放 – 在畫布上放大和縮小。

從 存取 Infrastructure Composer AWS Toolkit for Visual Studio Code

請遵循本主題中的指示，從 存取 Infrastructure Composer AWS Toolkit for Visual Studio Code。

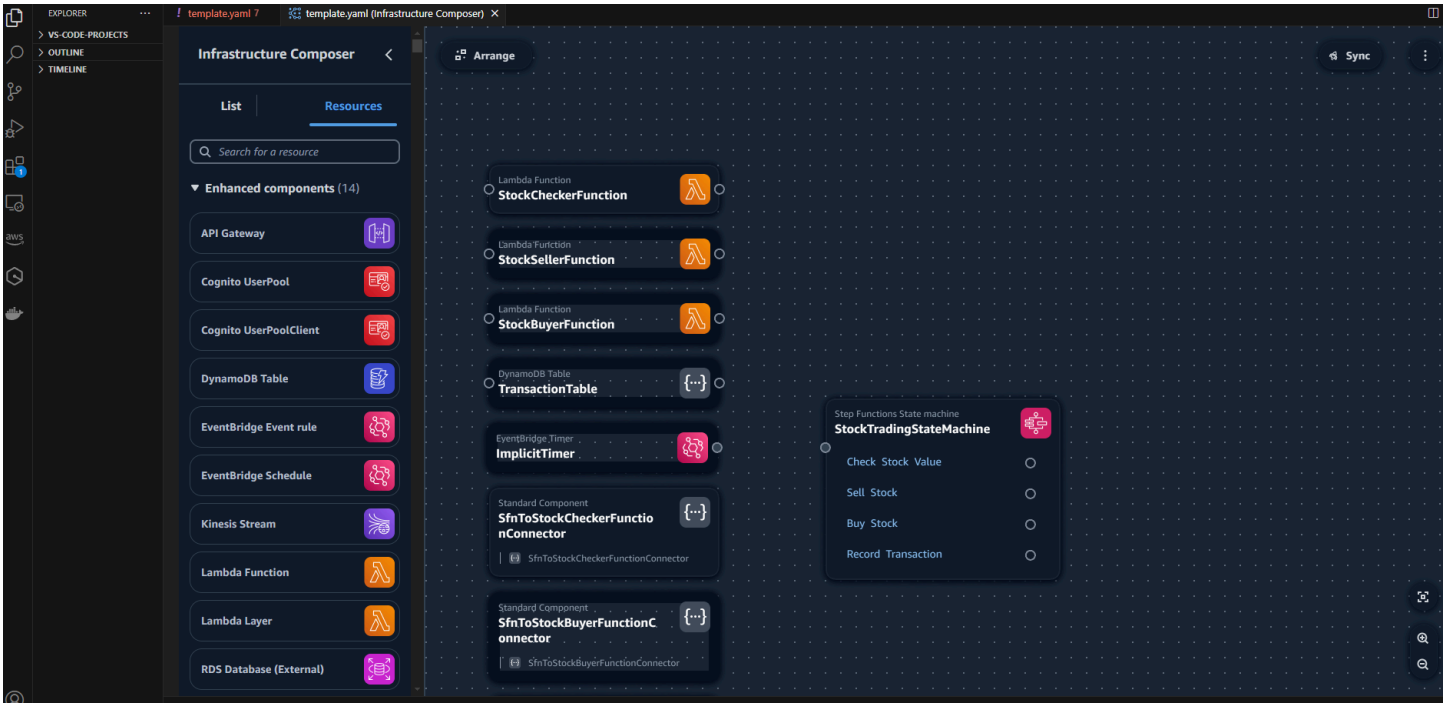
> **Note**
>
> 您必須先下載並安裝 Toolkit for VS Code AWS Toolkit for Visual Studio Code，才能從 存取 Infrastructure Composer。如需說明，請參閱下載 Toolkit for VS Code。

從 Toolkit for VS Code 存取 Infrastructure Composer

您可以透過下列任何方式存取 Infrastructure Composer：

1. 從任何 AWS CloudFormation 或 AWS SAM 範本中選取 Infrastructure Composer 按鈕。
2. 透過內容選單，在 AWS CloudFormation 或 AWS SAM 範本上按一下滑鼠右鍵。
3. 從 VS Code Command Palette。

以下是從 Infrastructure Composer 按鈕存取 Infrastructure Composer 的範例：



如需存取 Infrastructure Composer 的詳細資訊，請參閱AWS Infrastructure Composer 從 Toolkit 存取。

同步要部署到 的基礎設施編譯器 AWS 雲端

使用 AWS Infrastructure Composer 中的同步按鈕 AWS Toolkit for Visual Studio Code ，將您的應用程式部署到 AWS 雲端。

同步按鈕會從`sam sync`命令列界面 () 啟動 AWS SAM 命令CLI。

`sam sync` 命令可以部署新的應用程式，或快速將您本機所做的變更同步至 AWS 雲端。執行中`sam sync`可能包含下列項目：

- 使用 建置您的應用程式`sam build`，透過建立或更新本機`.aws-sam`目錄來準備您的本機應用程式檔案以供部署。
- 對於支援 AWS 服務 APIs的資源， AWS SAM CLI將使用 APIs 部署您的變更。 AWS SAM CLI 這樣做可快速更新雲端中的資源。

- 如有必要， AWS SAM CLI會執行 AWS CloudFormation 部署，透過變更集來更新整個堆疊。

`sam sync` 命令最適合快速更新雲端資源的快速開發環境，讓您的開發和測試工作流程受益。

若要進一步了解 `sam sync`，請參閱《 AWS Serverless Application Model 開發人員指南》中的使用 sam 同步。

設定

若要在 Infrastructure Composer 中使用同步功能，您必須在本機機器上安裝 AWS SAM CLI。如需說明，請參閱《 AWS SAM開發人員指南》中的安裝CLI 。 AWS Serverless Application Model

當您使用 Infrastructure Composer 中的同步功能時， AWS SAM CLI會參考您的組態檔案，以取得將應用程式同步至 所需的資訊 AWS 雲端。如需建立、修改和使用組態檔案的說明，請參閱《 AWS Serverless Application Model 開發人員指南》中的設定專案設定。

同步和部署您的應用程式

將您的應用程式同步至 AWS 雲端

1. 選取 Infrastructure Composer 畫布上的同步按鈕。
2. 您可能會收到確認您正在使用開發堆疊的提示。選取確定以繼續。
3. Infrastructure Composer 可能會提示您設定下列選項：

   - AWS 區域 – 要同步應用程式的區域。
   - AWS CloudFormation 堆疊名稱 – AWS CloudFormation 堆疊的名稱。您可以選取現有的堆疊名稱或建立新的堆疊名稱。
   - Amazon Simple Storage Service (Amazon S3) 儲存貯體 – Amazon S3 儲存貯體的名稱。 AWS SAM CLI 會將您的應用程式檔案和函數程式碼封裝並存放在此處。您可以選取現有的儲存貯體或建立新的儲存貯體。

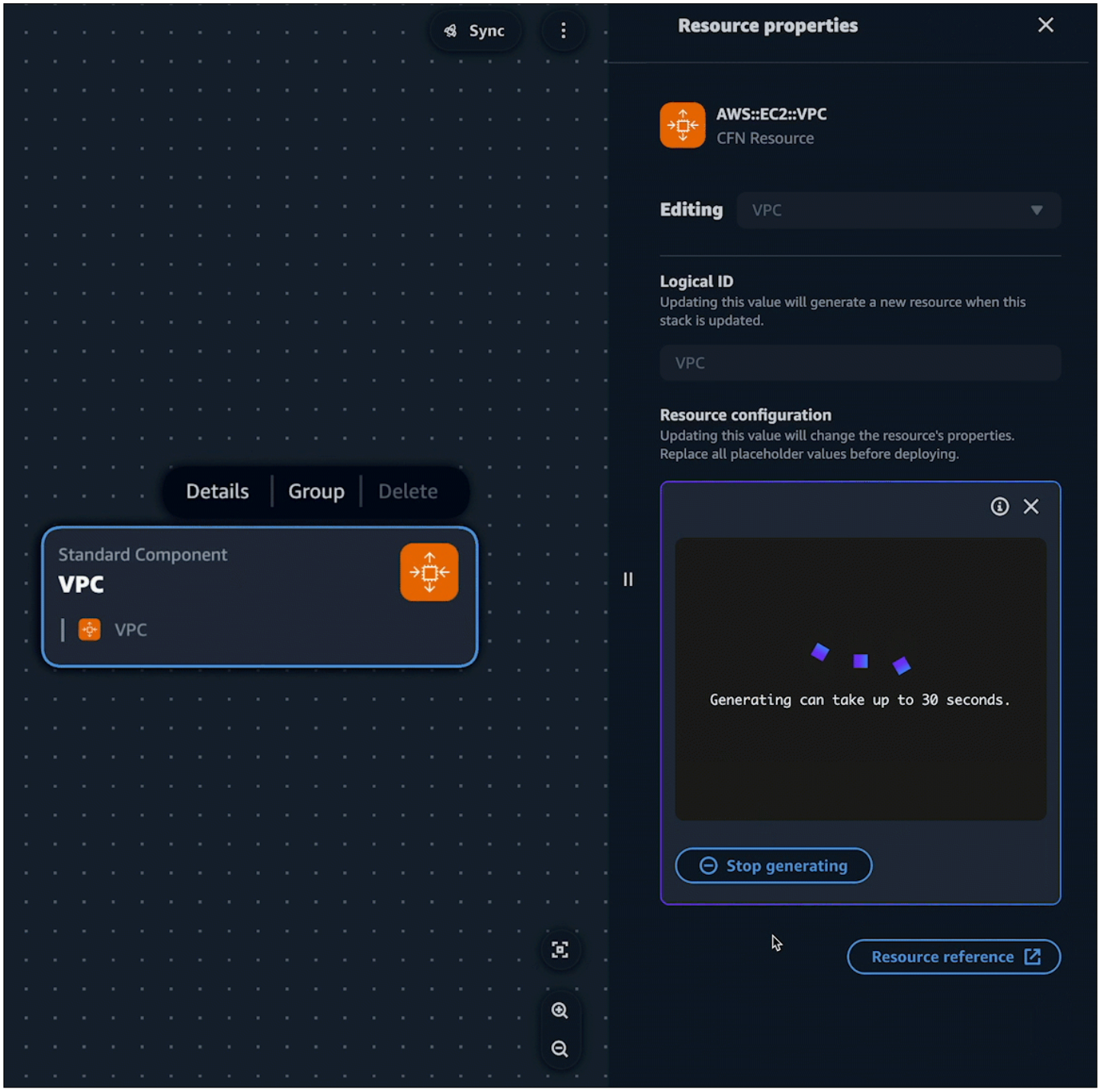
Infrastructure Composer 將啟動 AWS SAM CLI`sam sync`命令，並在 IDE 中開啟終端機視窗以輸出其進度。

AWS Infrastructure Composer 搭配 使用 Amazon Q Developer

AWS Infrastructure Composer 的 AWS Toolkit for Visual Studio Code 提供與 的整合Amazon Q。您可以在設計應用程式時，使用 Amazon Q Infrastructure Composer 為您的 AWS 資源產生基礎設施程式碼。

Amazon Q 是一般用途，採用機器學習技術的程式碼產生器。若要進一步了解，請參閱Amazon Q Developer《 使用者指南》中的什麼是 Amazon Q？。

對於標準資源和標準元件卡，您可以使用 Amazon Q為您的資源產生基礎設施程式碼建議。



標準資源和標準元件卡可以代表 AWS CloudFormation 資源或 AWS CloudFormation 資源集合。如需進一步了解，請參閱 在 Infrastructure Composer 中設定和修改卡片。

設定

若要Amazon Q在 Infrastructure Composer 中使用 ，您必須在 Toolkit Amazon Q中使用 進行身分驗證。如需說明，請參閱《 Amazon Q Developer 使用者指南Amazon Q》中的 VS Code 和 JetBrains 入門。

在 Infrastructure Composer Amazon Q Developer中使用

您可以從任何標準資源或標準元件卡Amazon Q Developer的資源屬性面板使用 。

在 Infrastructure Composer Amazon Q中使用

1. 從標準資源或標準元件卡中，開啟資源屬性面板。
2. 尋找資源組態欄位。此欄位包含卡片的基礎設施代碼。
3. 選取產生建議按鈕。 Amazon Q會產生建議。

> **Note**
>
> 在此階段產生的程式碼不會覆寫範本中現有的基礎設施程式碼。

4. 若要產生更多建議，請選取重新產生。您可以切換範例來比較結果。
5. 若要選取選項，請選擇選取。您可以在將程式碼儲存至應用程式之前，先修改此處的程式碼。若要在不儲存的情況下結束，請選取結束圖示 (X)。
6. 若要將程式碼儲存至應用程式範本，請從資源屬性面板中選取儲存。

進一步了解

若要進一步了解 Amazon Q，請參閱Amazon Q Developer《 使用者指南》中的什麼是 Amazon Q？。

如何在 中編寫 AWS Infrastructure Composer

本節涵蓋從 Infrastructure Composer 主控台、 CloudFormation 主控台模式和 使用 Infrastructure Composer 的基本概念AWS Toolkit for Visual Studio Code。更具體地說，本節中的主題提供如何使用 Infrastructure Composer 編寫應用程式的關鍵詳細資訊，並包含其他功能和捷徑的詳細資訊。主控台和 VS Code 體驗的功能有一些變化，本節中的主題會識別並描述這些變化的發生位置。

撰寫應用程式後，您將準備好將您的 Infrastructure Composer 無伺服器應用程式部署至 AWS 雲端檢閱有關部署應用程式的資訊。

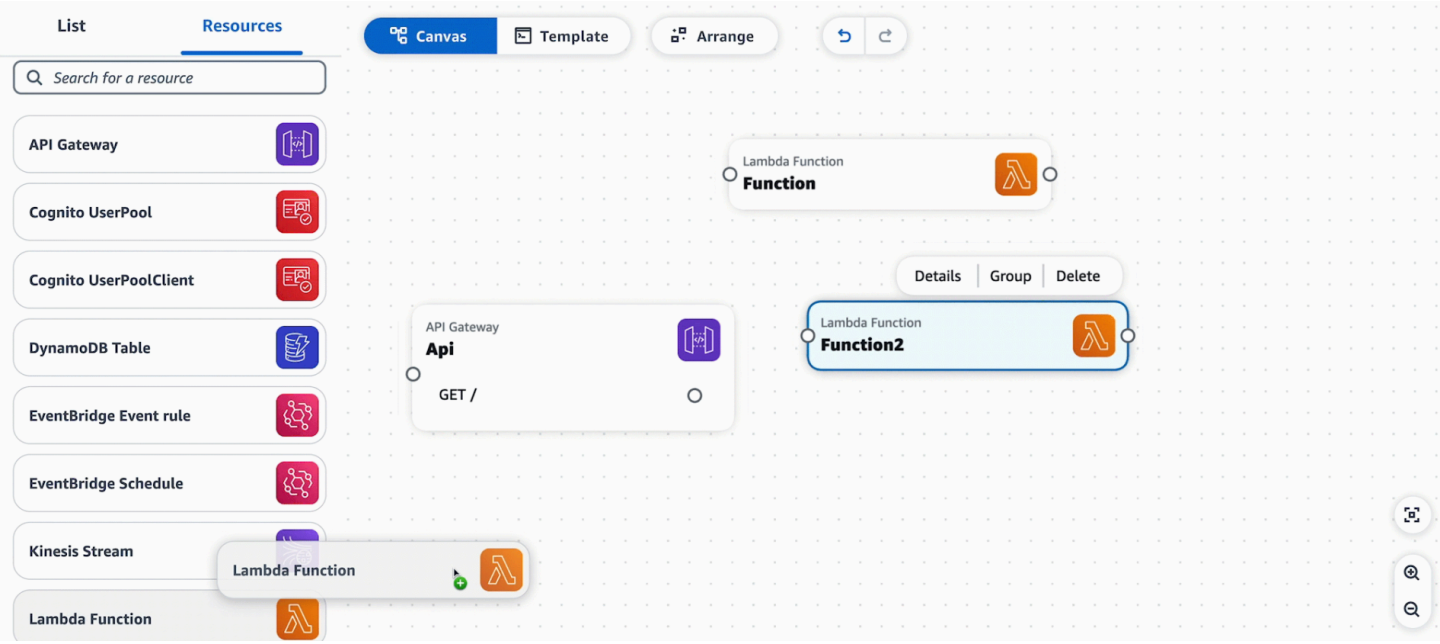
主題

- 將卡片放在 Infrastructure Composer 的視覺化畫布上
- 在 Infrastructure Composer 的視覺化畫布上將卡片分組在一起
- 在 Infrastructure Composer 的視覺化畫布上連接卡片
- 中斷 Infrastructure Composer 中的卡片連線
- 在 Infrastructure Composer 的視覺化畫布上排列卡片
- 在 Infrastructure Composer 中設定和修改卡片
- 在 Infrastructure Composer 中刪除卡片
- 使用 Infrastructure Composer 中的 Change Inspector 檢視程式碼更新
- 參考 Infrastructure Composer 中的外部檔案
- 將基礎設施編譯器與 Amazon Virtual Private Cloud (Amazon VPC) 整合

將卡片放在 Infrastructure Composer 的視覺化畫布上

本節說明如何在 Infrastructure Composer 卡的視覺化畫布中選取和拖曳卡片。開始之前，請識別您的應用程式需要哪些資源，以及它們需要如何互動。如需執行此操作的秘訣，請參閱使用 Infrastructure Composer 建置您的第一個應用程式。

若要將卡片新增至應用程式，請從資源調色盤拖曳卡片，並將其放在視覺化畫布上。



您可以從兩種類型的卡中選擇：增強型元件卡和標準 IaC 資源卡。

將卡片放在視覺化畫布上之後，您就可以分組、連線、安排和設定卡片。如需執行此作業的相關資訊，請參閱下列主題：

- 在 Infrastructure Composer 的視覺化畫布上將卡片分組在一起
- 在 Infrastructure Composer 的視覺化畫布上連接卡片
- 在 Infrastructure Composer 的視覺化畫布上排列卡片
- 在 Infrastructure Composer 中設定和修改卡片

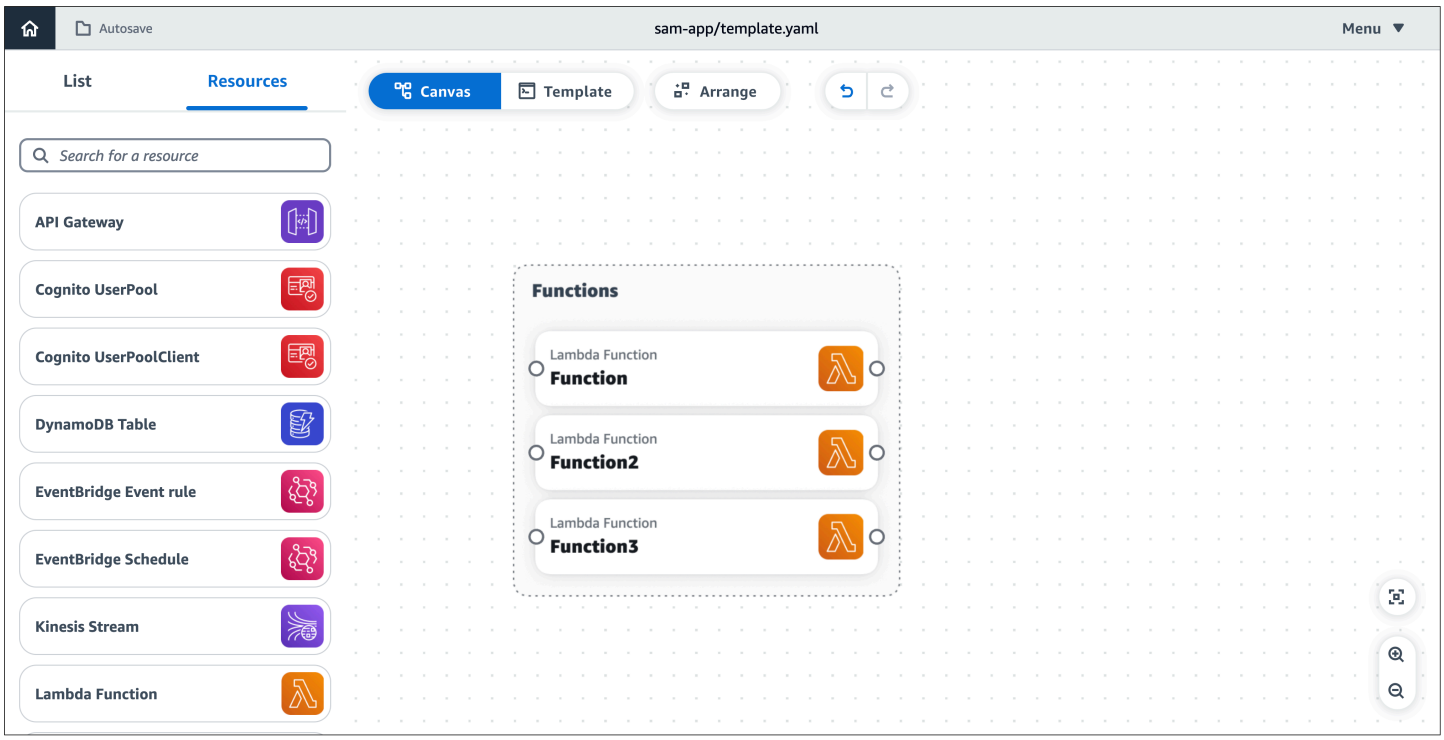
在 Infrastructure Composer 的視覺化畫布上將卡片分組在一起

本主題包含有關分組增強型元件卡和標準元件卡的詳細資訊。分組卡可協助您分類和組織資源，而無需考慮您需要寫入的程式碼或標記。

分組增強型元件卡

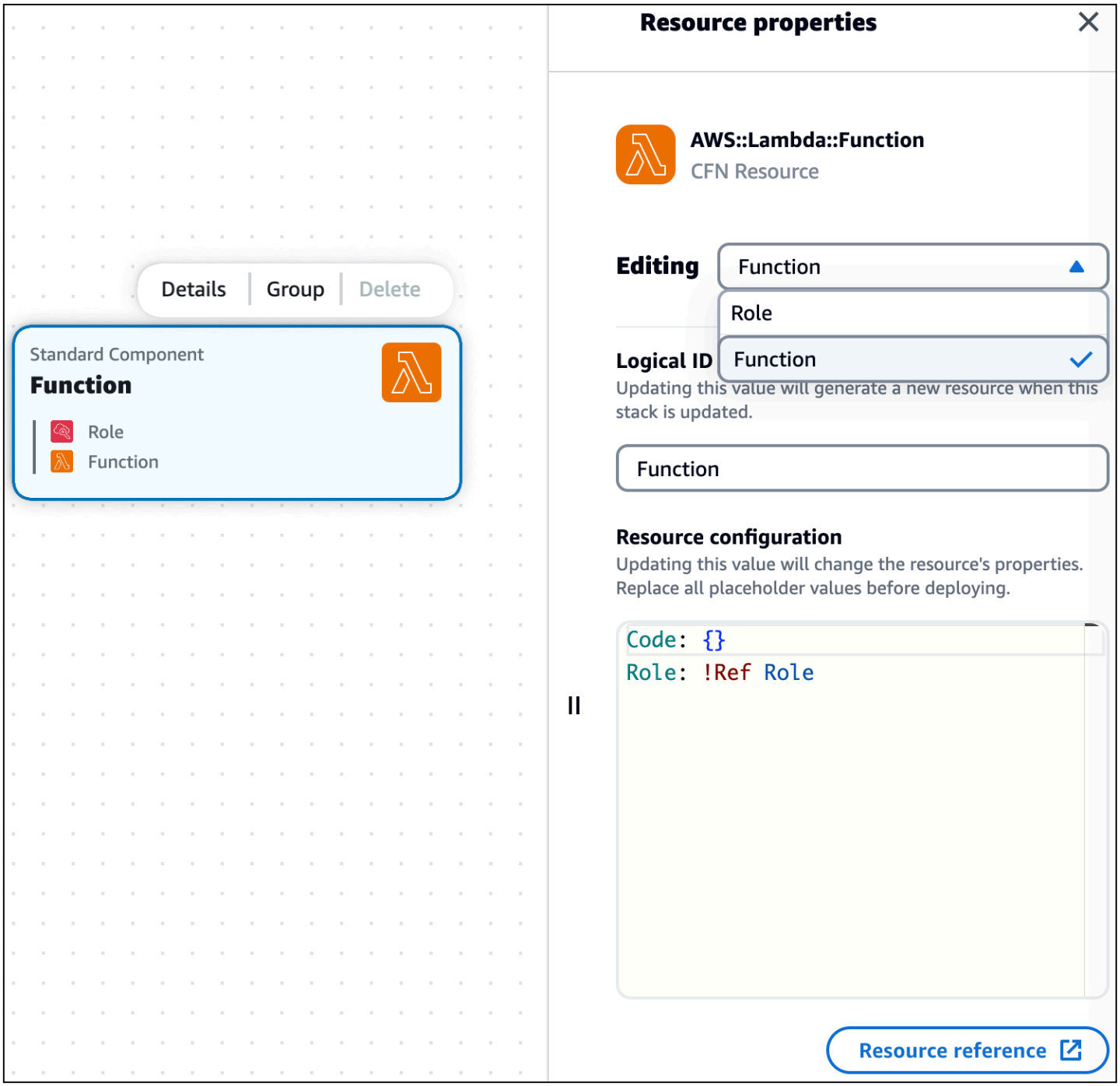
有兩種方式可將增強型元件卡分組在一起：

- 按下 Shift 時，選取要分組的卡片。然後，從資源動作功能表中選擇群組。
- 在群組中選取您想要的卡片。從出現的功能表中，選取群組。這將建立您可以拖放其他卡片的群組。



將標準元件卡分組到另一個

下列範例顯示從資源屬性面板將標準元件卡分組到其他卡片的一種方式：



在資源屬性面板上的資源組態欄位中，Role已在 Lambda 函數中參考 。 這會導致角色卡分組到畫布上的函數卡。

在 Infrastructure Composer 的視覺化畫布上連接卡片

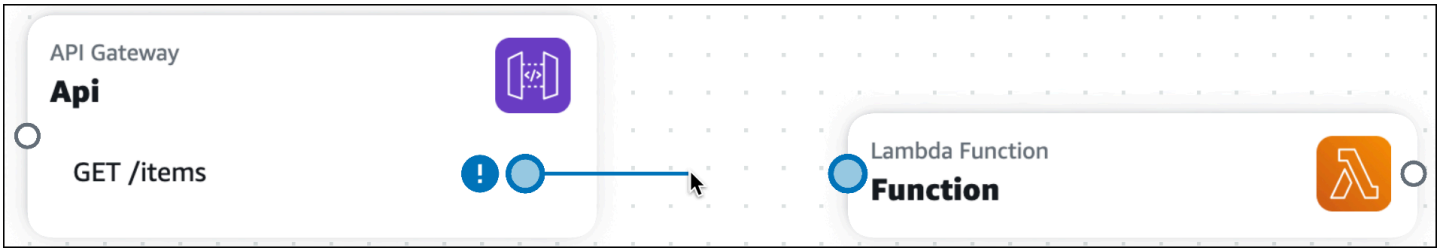
使用此主題來了解如何在 Infrastructure Composer 中連接卡片。本節包含有關連接增強型元件卡和標準元件卡的詳細資訊。它也提供一些範例，說明卡片連線的不同方式。

連接增強型元件卡

在增強型元件卡上，連接埠會以視覺方式識別可進行連線的位置。

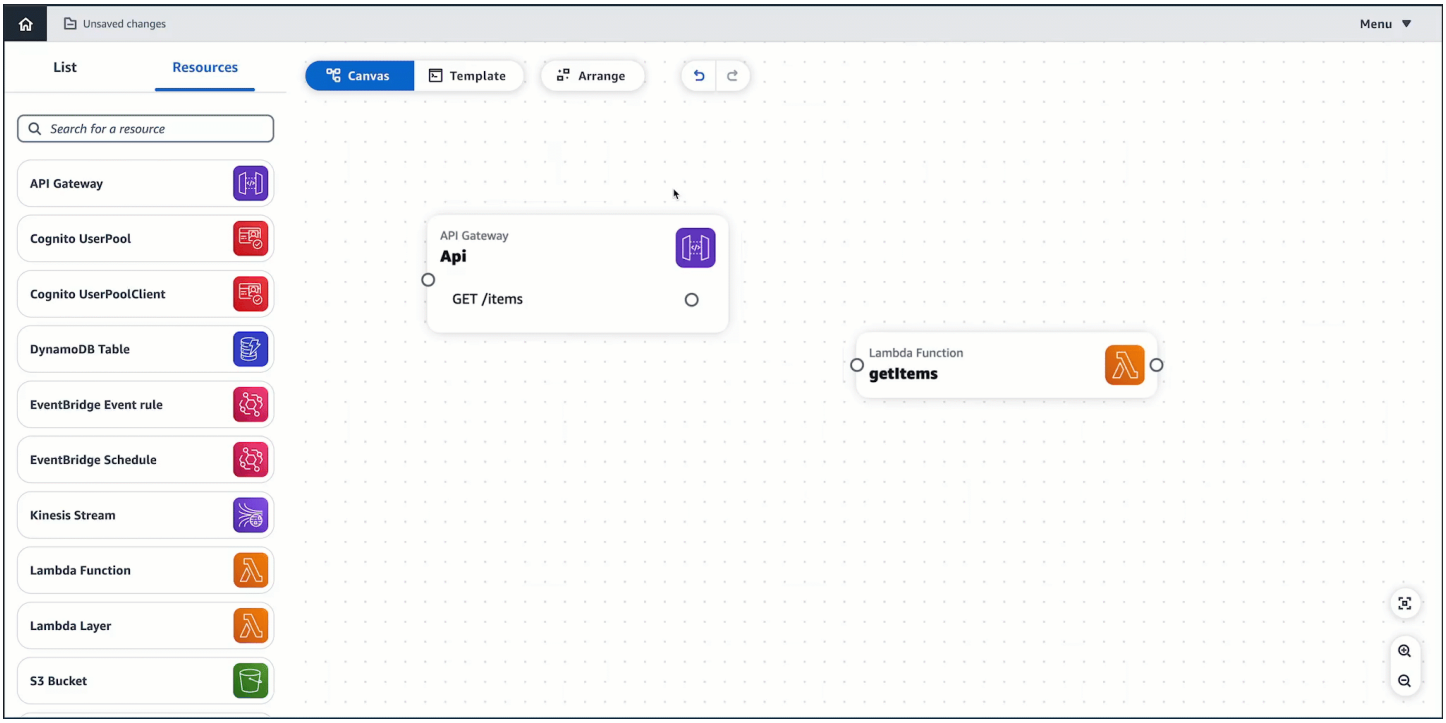
- 卡片右側的連接埠表示卡片有機會叫用另一張卡片。
- 卡片左側的連接埠表示卡片有機會被另一張卡片叫用。

按一下一張卡片的右側連接埠，並將其拖曳至另一張卡片的左側連接埠，以將卡片連接在一起。



當您建立連線時，系統會顯示訊息，讓您知道連線是否已成功建立。選取訊息，查看 Infrastructure Composer 已變更哪些內容來佈建連線。如果連線失敗，您可以選取範本檢視來手動更新基礎設施程式碼以佈建連線。

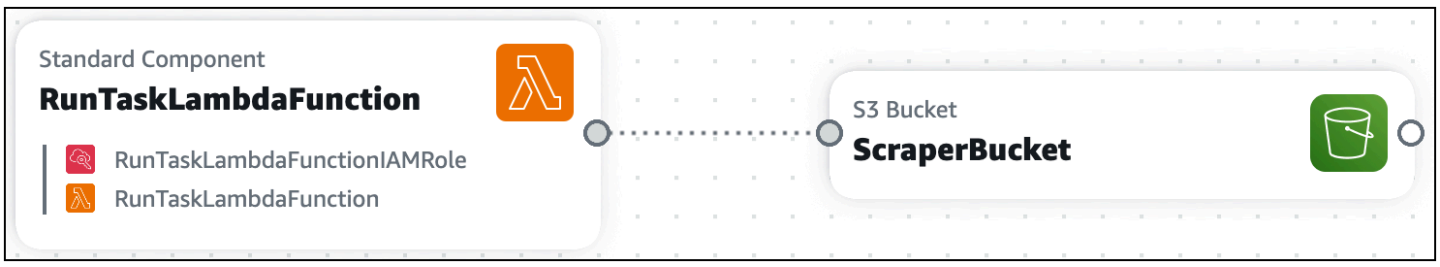
- 成功時，按一下訊息以檢視變更檢查器。在這裡，您可以查看修改了哪些 Infrastructure Composer 來佈建連線。
- 當失敗時，會顯示訊息。您可以選取範本檢視，並手動更新您的基礎設施程式碼來佈建連線。



當您將增強型元件卡連接在一起時，基礎設施編譯器會自動在範本中建立基礎設施程式碼，以佈建資源之間的事件驅動關係。

連接標準元件卡（標準 IaC 資源卡）

標準 IaC 資源卡不包含用於建立與其他資源連線的連接埠。在卡片組態期間，您會在應用程式的範本中指定事件驅動關係， Infrastructure Composer 會自動偵測這些連線，並在卡片之間以虛線呈現。以下是標準元件卡與增強型元件卡之間的連線範例：



下列範例顯示 Lambda 函數如何與 Amazon API Gateway 靜態 API 連線：

```
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
```

在上述範例中，ApiGatewayMethod: 中列出的程式碼片段會Integration:指定連接兩張卡片的事件驅動關係。

Infrastructure Composer 中連接卡片的範例

使用本節中的範例來了解如何在 Infrastructure Composer 中連接卡片。

將項目放入 Amazon Simple Storage Service (Amazon S3) 儲存貯體時叫用 AWS Lambda 函數

在此範例中，Amazon S3 儲存貯體卡已連接至 Lambda 函數卡。將項目放入 Amazon S3 儲存貯體時，會叫用 函數。然後，該函數可用於處理項目，或觸發應用程式中的其他事件。

此互動需要為函數定義事件。以下是 Infrastructure Composer 的規定：

```
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
```

從 Lambda 函數叫用 Amazon S3 儲存貯體

在此範例中，Lambda 函數卡會叫用 Amazon S3 儲存貯體卡。Lambda 函數可用於對 Amazon S3 儲存貯體中的項目執行 CRUD 操作。

此互動需要下列項目，由 Infrastructure Composer 佈建：

- 允許 Lambda 函數與 Amazon S3 儲存貯體互動的 IAM 政策。
- 影響 Lambda 函數行為的環境變數。

```
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
```

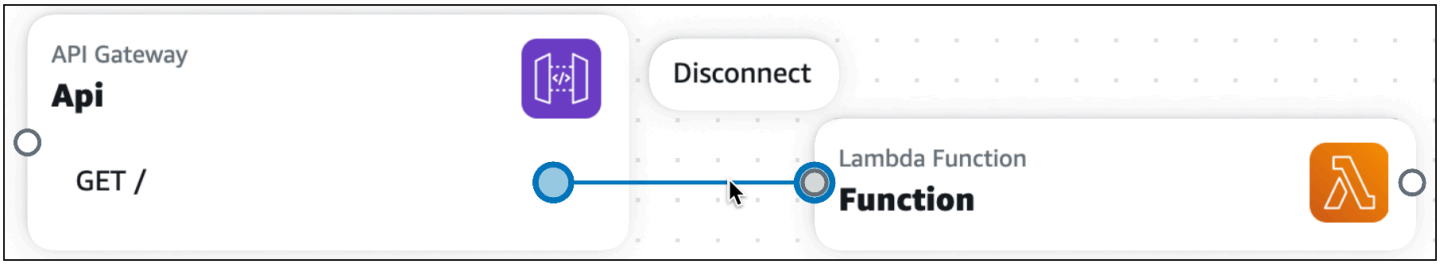
```
      Policies:
        - S3CrudPolicy:
          BucketName: !Ref MyBucket
```

中斷 Infrastructure Composer 中的卡片連線

在 Infrastructure Composer 中，您可以使用增強型元件卡和標準元件卡來連接和中斷連接 AWS 資源。本節說明如何中斷這兩種類型的卡片的連線。

增強型元件卡

若要中斷連線增強型元件卡，請選取該行，然後選擇中斷連線。



Infrastructure Composer 會自動修改您的範本，以從您的應用程式移除事件驅動的關係。

標準元件卡

標準元件卡不包含用於建立與其他資源連線的連接埠。在卡片組態期間，您會在應用程式的範本中指定事件驅動關係， Infrastructure Composer 會自動偵測這些連線，並在卡片之間以虛線呈現。若要中斷連接標準元件卡，請在應用程式的範本中移除事件驅動關係。

下列範例顯示與 Amazon API Gateway 靜態 API 連線的 Lambda 函數：

```
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)
```

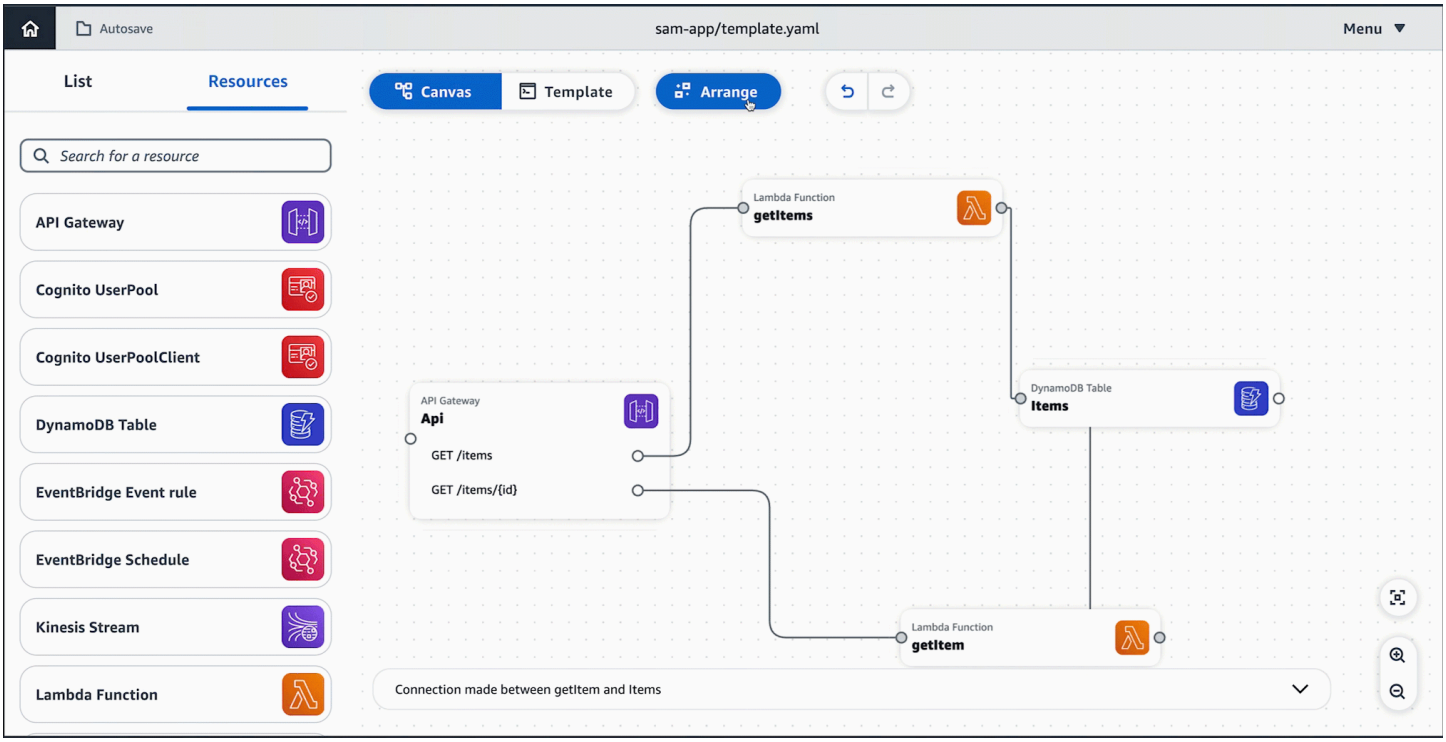
```
      ResourceId: !GetAtt MyApi.RootResourceId
      RestApiId: !Ref MyApi
      AuthorizationType: NONE
      Integration:
        Type: AWS_PROXY
        IntegrationHttpMethod: POST
        Uri: !Sub
          - arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/
${LambdaFunctionArn}/invocations
          - { 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
```

若要移除兩張卡片之間的連線，請移除 ApiGatewayMethod: 中MyLambdaFunction所列 的參考Integration。

在 Infrastructure Composer 的視覺化畫布上排列卡片

選取排列以視覺化方式排列和整理畫布上的卡片。當畫布上有許多卡片和連線時，使用排列按鈕特別有用。



在 Infrastructure Composer 中設定和修改卡片

在 Infrastructure Composer 中，卡片代表您用來設計應用程式架構的資源。當您在 Infrastructure Composer 中設定卡片時，您可以定義應用程式中資源的詳細資訊。這包括卡片邏輯 ID 和分割區金鑰等詳細資訊。此資訊的定義方式因增強型元件卡和標準卡而異。

增強型元件卡是 資源的集合 AWS CloudFormation ，已合併為單一精選的卡，可增強易用性、功能，且專為各種使用案例而設計。標準 IaC 資源卡代表單一 AWS CloudFormation 資源。每個標準 IaC 資源卡一旦拖曳到畫布上，都會標示為標準元件。

本主題提供有關設定增強型元件卡和標準元件卡的詳細資訊。

> **Note**
>
> 本主題適用於在 CloudFormation 主控台模式下使用來自 Infrastructure Composer 主控台、AWS Toolkit for Visual Studio Code 延伸模組和 Infrastructure Composer 的卡片。Lambda 相關卡 (Lambda 函數和 Lambda Layer) 需要程式碼建置和封裝解決方案，這些解決方案在 CloudFormation 主控台模式下無法在 Infrastructure Composer 中使用。如需詳細資訊，請參閱在 CloudFormation 主控台模式中使用 Infrastructure Composer。

主題

Infrastructure Composer 中的增強型元件卡

若要設定增強型元件卡，基礎設施編譯器會在資源屬性面板中提供表單。此表單是專門設計的，以引導您設定每個增強型元件卡。在您填寫表單時，基礎設施編寫器會修改您的基礎設施程式碼。

某些增強型元件卡具有其他功能。本節會檢閱使用增強型元件卡的基本概念，並提供具有其他功能之卡的詳細資訊。

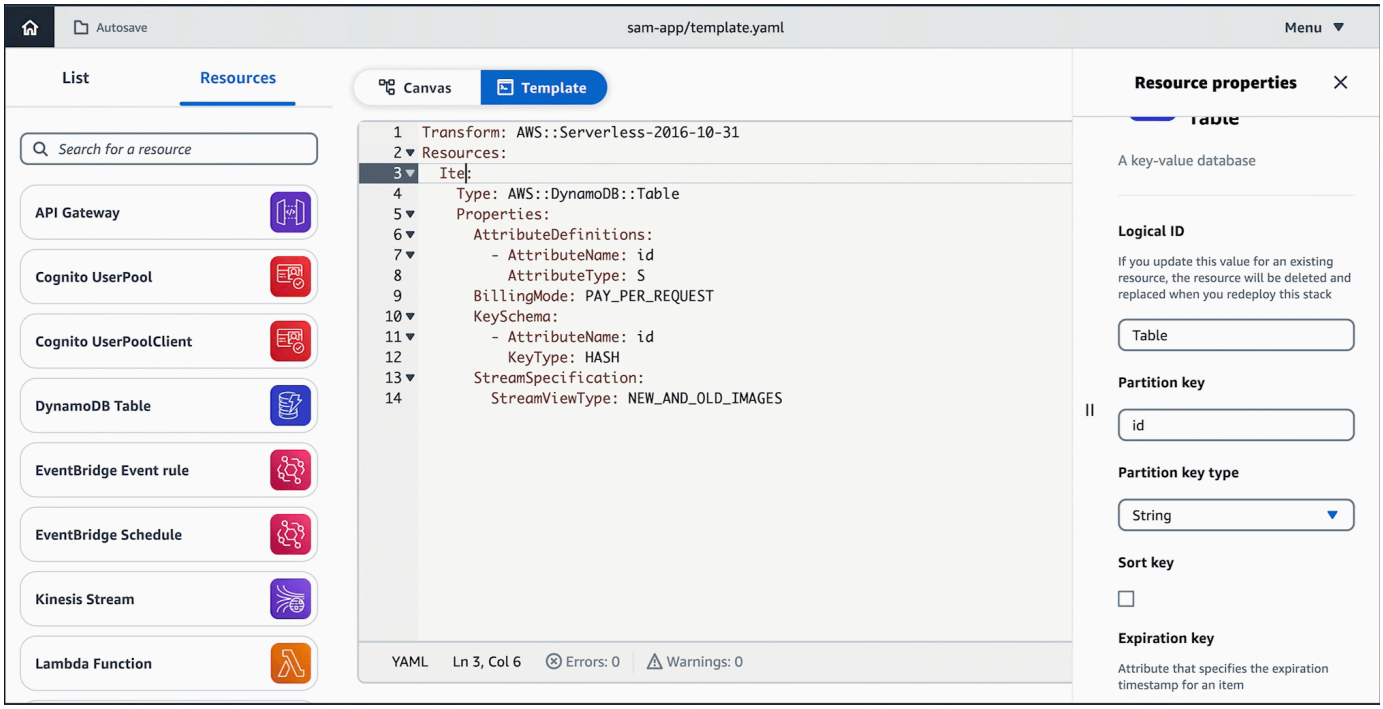
如需增強型元件卡的詳細資訊，請參閱 Infrastructure Composer 中的增強型元件卡 和 Infrastructure Composer 中的增強型元件卡

程序

資源屬性面板可簡化組態，並新增可簡化卡片組態的指南。若要使用此面板，請執行下列步驟：

1. 按兩下卡片以顯示資源屬性面板。
2. 按一下卡片並選取詳細資訊，以顯示資源屬性面板。
3. 對於 Infrastructure Composer AWS Management Console，從 選取 範本以顯示您的應用程式碼。直接從這裡設定 。

   下圖顯示如何完成此操作：



搭配 Amazon Relational Database Service (Amazon RDS) 使用 Infrastructure Composer

AWS Infrastructure Composer 具有與 Amazon Relational Database Service (Amazon RDS) 整合的功能。使用 Infrastructure Composer 中的 RDS 資料庫 （外部） 增強型元件卡，您可以將應用程式連線到另一個 AWS CloudFormation 或 AWS Serverless Application Model (AWS SAM) 範本上定義的 Amazon RDS DB叢集、執行個體和代理。

RDS 資料庫 （外部） 增強型元件卡代表在另一個範本上定義的 Amazon RDS 資源。其中包含：

- 在其他範本上定義的 Amazon RDS DB叢集或執行個體
- Amazon RDS DB代理

RDS 資料庫 （外部） 增強型元件卡可從資源面板取得。

若要使用此卡，請將其拖曳到 Infrastructure Composer 畫布上、進行設定，並將其連接到其他資源。

您可以透過 Lambda 函數將應用程式連線到外部 Amazon RDS DB叢集或執行個體。

要求

若要使用此功能，您必須符合下列要求：

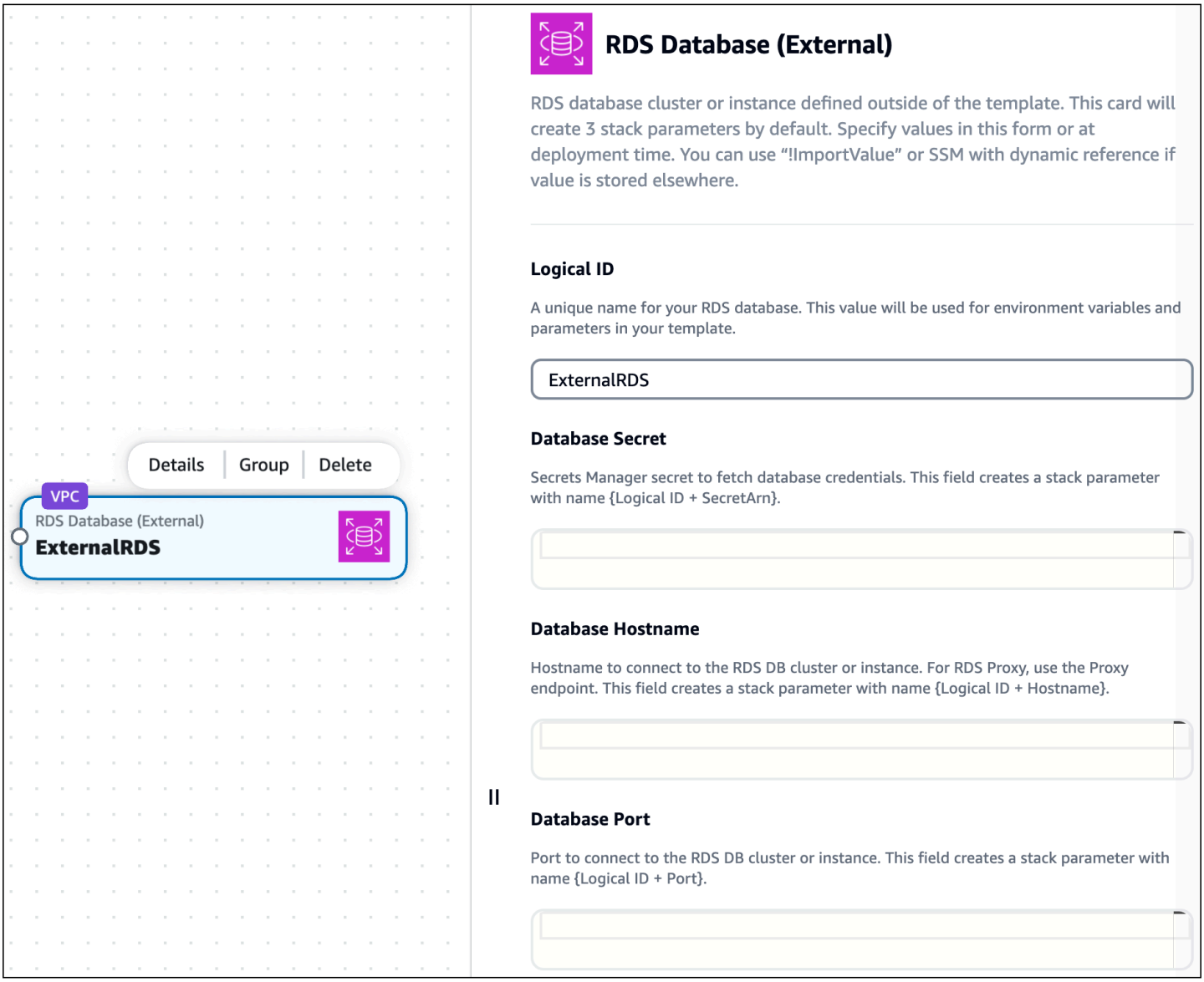
1. 您的外部 Amazon RDS DB叢集、執行個體或代理必須使用 AWS Secrets Manager 來管理使用者密碼。若要進一步了解，請參閱《Amazon RDS 使用者指南》中的使用 Amazon RDS 和 進行密碼管理 AWS Secrets Manager。
2. 您在 Infrastructure Composer 中的應用程式必須是新專案，或最初必須在 Infrastructure Composer 中建立。

程序

步驟 1：設定外部 RDS 資料庫卡

從資源調色盤中，將 RDS 資料庫 （外部） 增強型元件卡拖曳到畫布上。

選取卡片，然後選擇詳細資訊，或按兩下卡片以顯示資源屬性面板。卡片的資源屬性面板將會出現：



您可以在此設定下列項目：

- 邏輯 ID – 外部 Amazon RDS DB叢集、執行個體或代理的唯一名稱。此 ID 不必符合您外部 Amazon RDS DB 資源的邏輯 ID 值。
- 資料庫秘密 – 與您的 Amazon RDS DB叢集、執行個體或代理相關聯的 AWS Secrets Manager 秘密識別符。此欄位接受下列值：
  - 靜態值 – 資料庫秘密的唯一識別符，例如秘密 ARN。以下是範例：`arn:aws:secretsmanager:us-west-2:123456789012:secret:my-path/my-secret-name-1a2b3c`如需詳細資訊，請參閱《AWS Secrets Manager 使用者指南》中的 AWS Secrets Manager 概念。

  - 輸出值 – 部署 Secrets Manager 秘密時 AWS CloudFormation，會建立輸出值。您可以使用Fn::ImportValue內部 函數指定此處的輸出值。例如：!ImportValue MySecret。
  - 來自 SSM 參數存放區的值 – 您可以將秘密存放在 SSM 參數存放區，並使用動態參考指定其值。例如：{{resolve:ssm:MySecret}}。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的 SSM 參數。
- 資料庫主機名稱 – 可用來連線至 Amazon RDS DB叢集、執行個體或代理的主機名稱。此值是在定義 Amazon RDS 資源的外部範本中指定。接受下列值：
  - 靜態值 – 資料庫主機名稱的唯一識別符，例如端點地址。以下是範例：mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-2.rds.amazonaws.com
  - 輸出值 – 部署的 Amazon RDS DB叢集、執行個體或代理的輸出值。您可以使用Fn::ImportValue內部 函數指定輸出值。例如：!ImportValue myStack-myDatabase-abcd1234。
  - 來自 SSM 參數存放區的值 – 您可以在 SSM 參數存放區中存放資料庫主機名稱，並使用動態參考指定其值。例如：{{resolve:ssm:MyDatabase}}。
- 資料庫連接埠 – 可用來連線至 Amazon RDS DB叢集、執行個體或代理的連接埠號碼。此值是在定義 Amazon RDS 資源的外部範本中指定。接受下列值：
  - 靜態值 – 資料庫連接埠。例如：3306。
  - 輸出值 – 部署的 Amazon RDS DB叢集、執行個體或代理的輸出值。例如：!ImportValue myStack-MyRDSInstancePort。
  - 來自 SSM 參數存放區的值 – 您可以在 SSM 參數存放區中存放資料庫主機名稱，並使用動態參考指定其值。例如：{{resolve:ssm:MyRDSInstancePort}}。

> **Note**
>
> 此處只能設定邏輯 ID 值。您可以視需要在部署時間設定其他屬性。

步驟 2：連接 Lambda 函數卡

從資源調色盤中，將 Lambda 函數增強型元件卡拖曳到畫布上。

將 Lambda 函數卡的左側連接埠連接到 RDS 資料庫 （ 外部 ） 卡的右側連接埠。

Infrastructure Composer 將佈建您的範本，以促進此連線。

Infrastructure Composer 如何建立您的連線

當您完成上述程序時， Infrastructure Composer 會執行特定動作，將 Lambda 函數連接至資料庫。

指定外部 Amazon RDS DB叢集、執行個體或代理時

當您將 RDS 資料庫 （外部） 卡拖曳到畫布上時， Infrastructure Composer 會視需要更新範本的 `Metadata`和 `Parameters`區段。以下是範例：

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

中繼資料是 AWS CloudFormation 範本區段，用於存放範本的詳細資訊。Infrastructure Composer 特有的中繼資料存放在`AWS::Composer::ExternalResources`中繼資料金鑰下。在這裡， Infrastructure Composer 會儲存您為 Amazon RDS DB叢集、執行個體或代理指定的值。

範本的參數區段 AWS CloudFormation 用於存放自訂值，這些值可在部署時插入整個範本。根據您提供的值類型，基礎設施編寫器可能會將 Amazon RDS DB叢集、執行個體或代理的值存放在此處，並在範本中指定這些值。

`Metadata` 和 `Parameters`區段中的字串值使用您在 RDS 資料庫 （外部） 卡上指定的邏輯 ID 值。如果您更新邏輯 ID，字串值將會變更。

將 Lambda 函數連接至資料庫時

當您將 Lambda 函數卡連接至 RDS 資料庫 （外部） 卡時， Infrastructure Composer 會佈建環境變數和 AWS Identity and Access Management (IAM) 政策。以下是範例：

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

環境變數是可在執行時間由函數使用的變數。若要進一步了解，請參閱《 AWS Lambda 開發人員指南》中的使用 Lambda 環境變數。

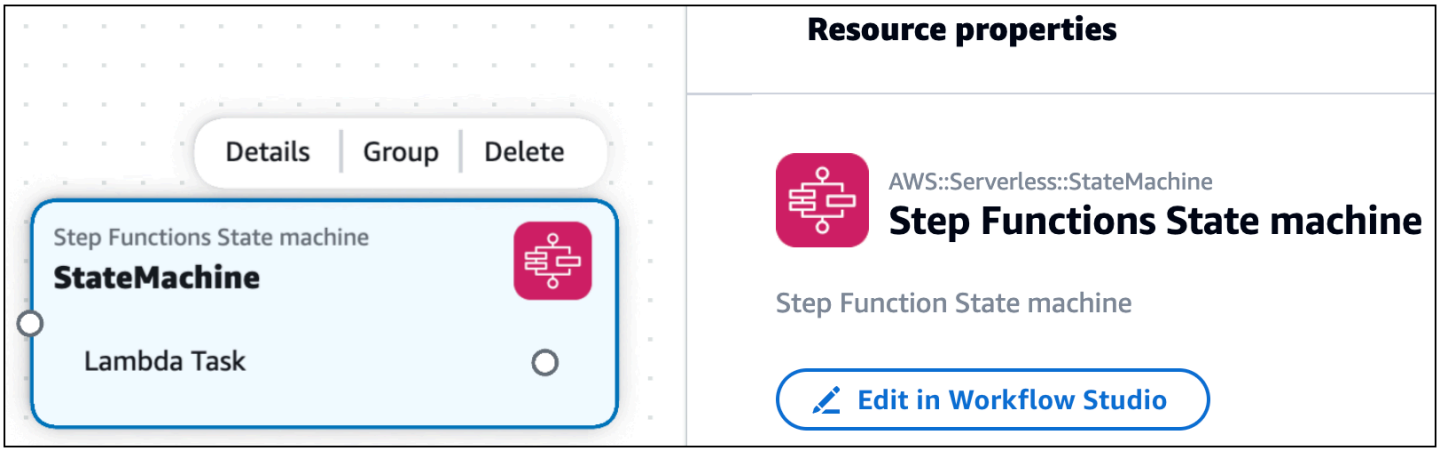
政策佈建函數的許可。在這裡， Infrastructure Composer 會建立政策，以允許從函數讀取存取 Secrets Manager，以取得存取 Amazon RDS DB叢集、執行個體或代理的密碼。

AWS Infrastructure Composer 搭配 使用 AWS Step Functions

AWS Infrastructure Composer 具有與 整合的功能AWS Step FunctionsWorkflow Studio。使用 Infrastructure Composer 執行下列動作：

- Workflow Studio 直接在 Infrastructure Composer 中啟動 Step Functions。
- 建立新的工作流程和管理，或將現有的工作流程匯入 Infrastructure Composer。
- 使用 Infrastructure Composer 畫布將您的工作流程與其他 AWS 資源整合。

下圖是 Step Functions 狀態機器卡



透過 Infrastructure Composer Workflow Studio中的 Step Functions，您可以在單一位置使用兩個強大的視覺化設計工具的優點。當您設計工作流程和應用程式時，基礎設施編寫器會將基礎設施建立為程式碼 (IaC)，以引導您進行部署。

主題

- IAM 政策
- Infrastructure Composer Workflow Studio中的步驟函數入門
- 在 Infrastructure Composer Workflow Studio中使用步驟函數
- 進一步了解

IAM 政策

當您將任務從工作流程連接到資源時，基礎設施編譯器會自動建立所需的 AWS Identity and Access Management (IAM) 政策，以授權資源之間的互動。以下是範例：

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

如有必要，您可以將更多 IAM 政策新增至範本。

Infrastructure Composer Workflow Studio中的步驟函數入門

若要開始使用，您可以建立新的工作流程或匯入現有的工作流程。

建立新的工作流程

1. 從資源調色盤中，將 Step Functions 狀態機器增強型元件卡拖曳到畫布上。

   當您將 Step Functions State 機器卡拖曳到畫布上時， Infrastructure Composer 會建立下列項目：

   - 定義狀態機器 `AWS::Serverless::StateMachine`的資源。根據預設， Infrastructure Composer 會建立標準工作流程。若要建立快速工作流程，請將範本中的`Type`值從 變更為 `STANDARD` `EXPRESS`。
   - 為您的狀態機器定義 Amazon CloudWatch 日誌群組`AWS::Logs::LogGroup`的資源。

2. 開啟卡片的資源屬性面板，然後選取 Workflow Studio 中的編輯，以在 Infrastructure Composer Workflow Studio中開啟。

   Step Functions 會在設計模式中Workflow Studio開啟。若要進一步了解，請參閱《 AWS Step Functions 開發人員指南》中的設計模式。

   > **Note**
   >
   > 您可以修改 Infrastructure Composer，將狀態機器定義儲存在外部檔案中。如需進一步了解，請參閱 使用外部檔案。

3. 建立您的工作流程，然後選擇儲存。若要結束 Workflow Studio，請選擇返回至基礎設施編寫器。

   Infrastructure Composer 會使用 `AWS::Serverless::StateMachine` 資源的 `Defintion` 屬性來定義您的工作流程。

4. 您可以執行下列任何動作來修改工作流程：

   - Workflow Studio 再次開啟並修改您的工作流程。

- 對於 Infrastructure Composer，您可以從主控台開啟應用程式的範本檢視，並修改您的範本。如果使用本機同步，您可以在本機 IDE 中修改工作流程。Infrastructure Composer 會偵測您的變更，並在 Infrastructure Composer 中更新您的工作流程。
- 對於 Toolkit for VS Code 中的 Infrastructure Composer，您可以直接修改範本。Infrastructure Composer 會偵測您的變更，並在 Infrastructure Composer 中更新您的工作流程。

匯入現有的工作流程

您可以從使用 AWS Serverless Application Model (AWS SAM) 範本定義的應用程式匯入工作流程。使用任何以 `AWS::Serverless::StateMachine` 資源類型定義的狀態機器，它將視覺化為 Step Functions 狀態機器增強型元件卡，您可以用來啟動 Workflow Studio。

`AWS::Serverless::StateMachine` 資源可以使用下列任一屬性定義工作流程：

- `Definition` – 工作流程在 AWS SAM 範本中定義為 物件。
- `DefinitionUri` – 使用 Amazon States 語言在外部檔案上定義工作流程。然後使用此屬性指定檔案的本機路徑。

定義屬性

主控台中的基礎設施編寫器

對於使用 `Definition` 屬性定義的工作流程，您可以匯入單一範本或整個專案。

- 範本 – 如需匯入範本的指示，請參閱在 Infrastructure Composer 主控台中匯入現有的專案範本。若要儲存您在 Infrastructure Composer 中所做的變更，您必須匯出範本。
- 專案 – 當您匯入專案時，您必須啟用本機同步。您所做的變更會自動儲存至本機機器。如需匯入專案的指示，請參閱 在 Infrastructure Composer 主控台中匯入現有的專案資料夾。

來自 Toolkit for VS Code 的 Infrastructure Composer

對於使用 `Definition` 屬性定義的工作流程，您可以從 範本開啟 Infrastructure Composer。如需說明，請參閱 從 存取 Infrastructure Composer AWS Toolkit for Visual Studio Code。

DefinitionUri 屬性

主控台中的基礎設施編寫器

對於使用 `DefinitionUri` 屬性定義的工作流程，您必須匯入專案並啟用本機同步。如需匯入專案的指示，請參閱 在 Infrastructure Composer 主控台中匯入現有的專案資料夾。

來自 Toolkit for VS Code 的 Infrastructure Composer

對於使用 `DefinitionUri` 屬性定義的工作流程，您可以從 範本開啟 Infrastructure Composer。如需說明，請參閱 從 存取 Infrastructure Composer AWS Toolkit for Visual Studio Code。

在 Infrastructure Composer Workflow Studio中使用步驟函數

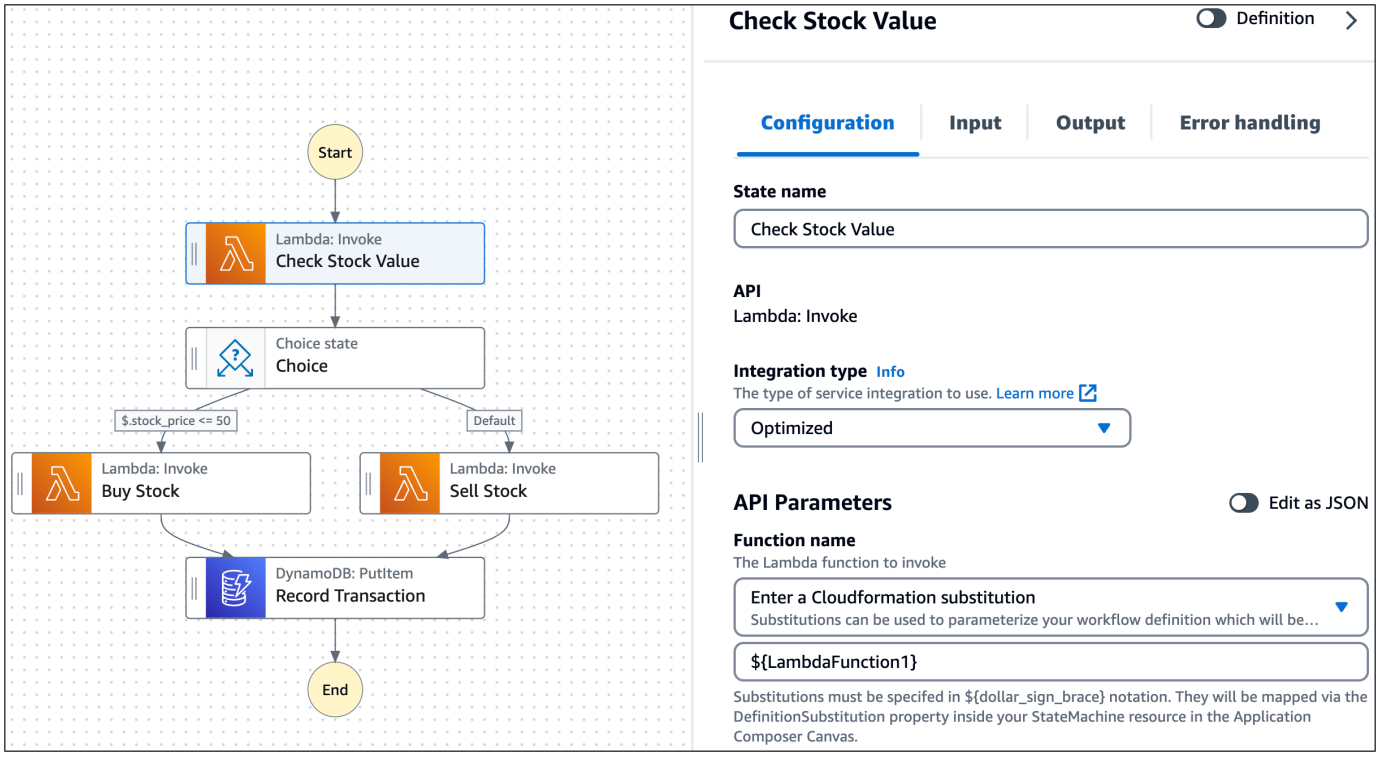
建置工作流程

Infrastructure Composer 使用定義替換，將工作流程任務映射到應用程式中的資源。若要進一步了解定義替換，請參閱《 AWS Serverless Application Model 開發人員指南 `DefinitionSubstitutions`》中的 。

當您在 中建立任務時Workflow Studio，請為每個任務指定定義替代。然後，您可以將任務連接到 Infrastructure Composer 畫布上的資源。

在 中指定定義替換 Workflow Studio

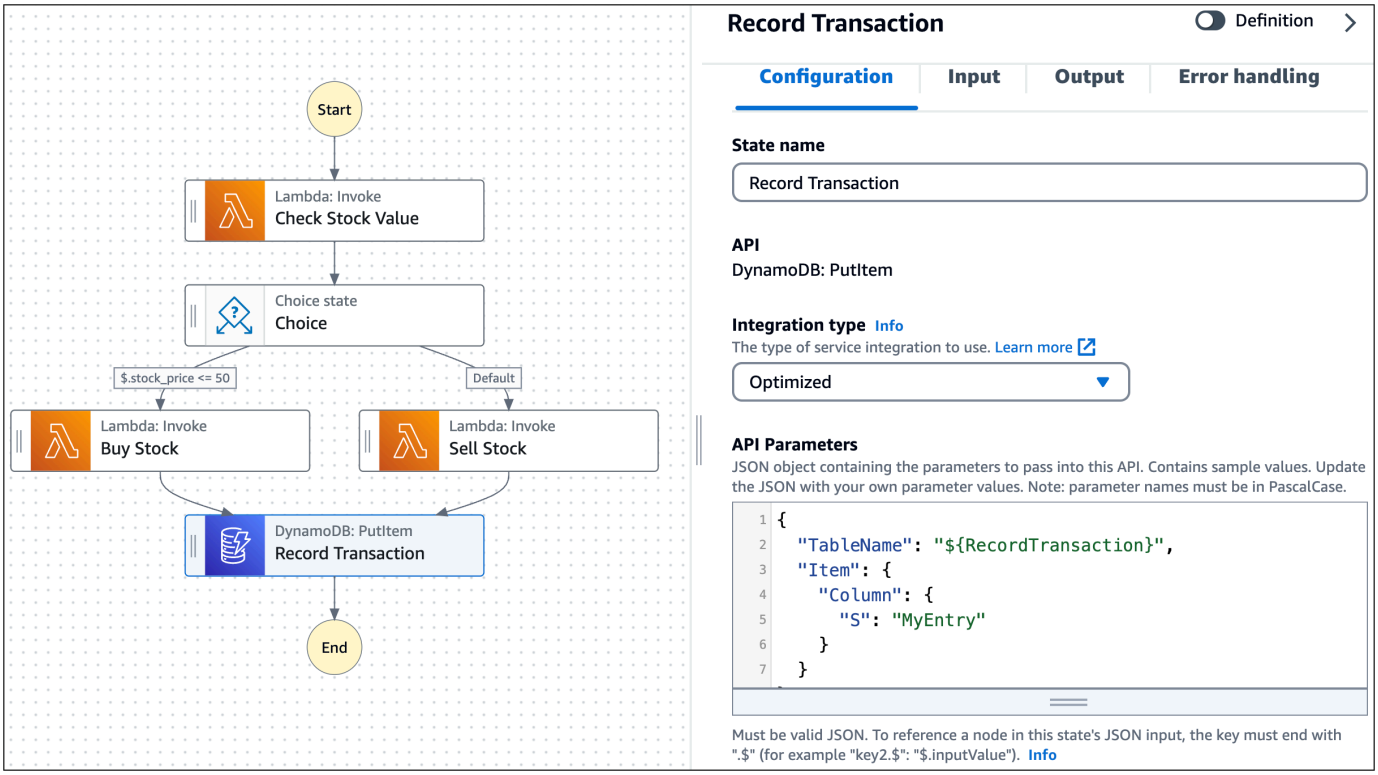
1. 開啟任務的組態索引標籤，並找到 API 參數欄位。



2. 如果 API 參數欄位有下拉式清單選項，請選擇輸入 AWS CloudFormation 替代。然後，提供唯一的名稱。

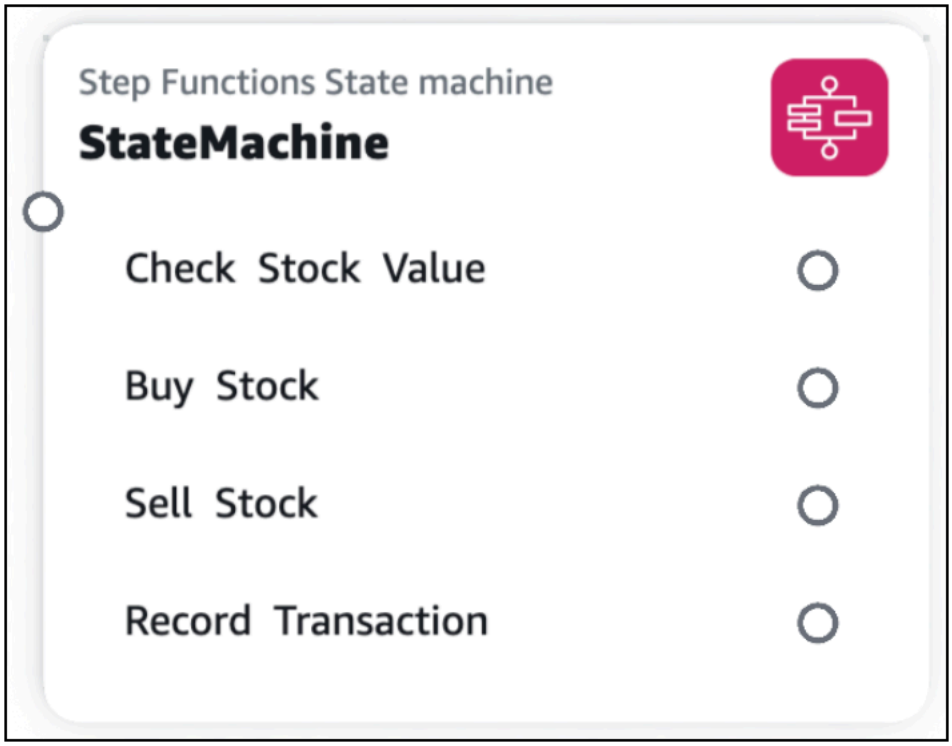
對於連接到相同資源的任務，請為每個任務指定相同的定義替換。若要使用現有的定義替代，請選擇選取 AWS CloudFormation 替代，然後選取要使用的替代。

3. 如果 API 參數欄位包含 JSON 物件，請修改指定資源名稱的項目，以使用定義替換。在下列範例中，我們將 `"MyDynamoDBTable"`變更為 `"${RecordTransaction}"`。



4. 選取儲存並返回基礎設施編寫器。

工作流程中的任務會顯示在 Step Functions 狀態機器卡上。



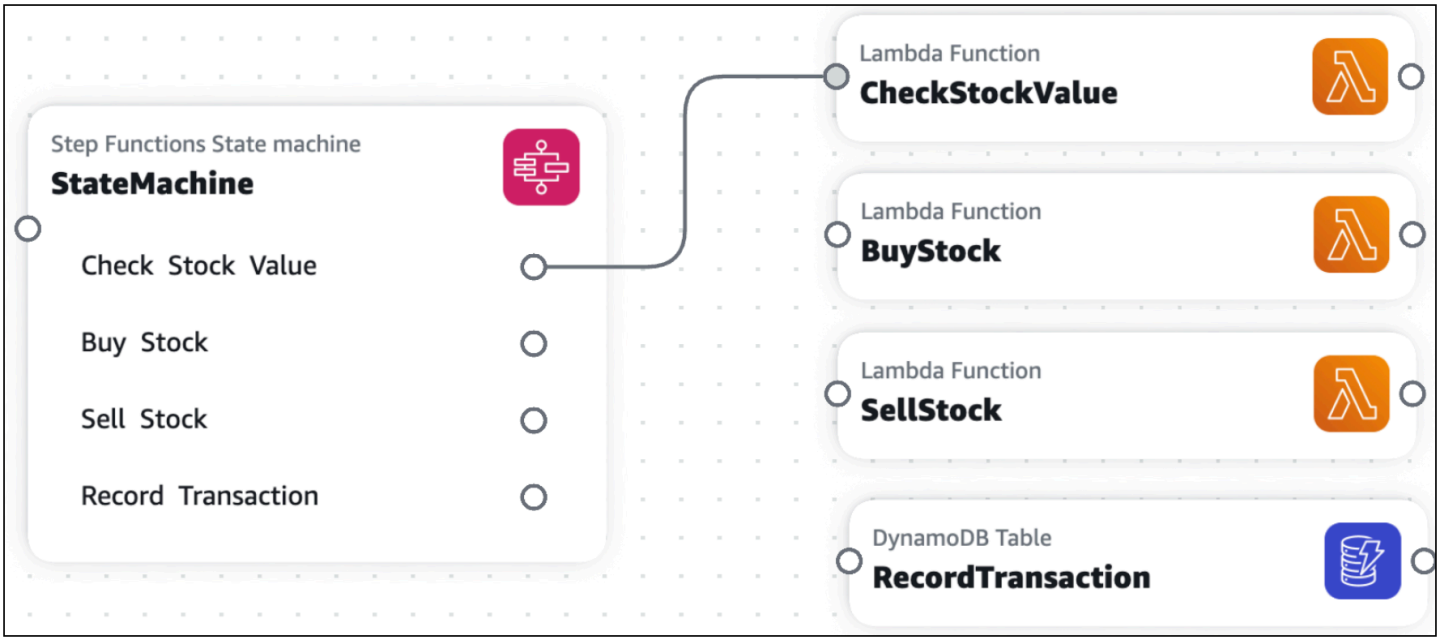
將資源連接至工作流程任務

您可以在 Infrastructure Composer 中，在支援的工作流程任務和支援的 Infrastructure Composer 卡之間建立連線。

- 支援的工作流程任務 – 針對 Step Functions 最佳化 AWS 服務 的 任務。若要進一步了解，請參閱《AWS Step Functions 開發人員指南》中的步驟函數最佳化整合。
- 支援的 Infrastructure Composer 卡 – 支援增強型元件卡。若要進一步了解 Infrastructure Composer 中的卡片，請參閱在 Infrastructure Composer 中設定和修改卡片。

建立連線時，任務和卡片 AWS 服務 的 必須相符。例如，您可以將叫用 Lambda 函數的工作流程任務連接到 Lambda 函數增強型元件卡。

若要建立連線，請按一下任務的連接埠，並將其拖曳至增強型元件卡的左側連接埠。



Infrastructure Composer 會自動更新您的`DefinitionSubstitution`值，以定義您的連線。以下是範例：

```
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:
      ...
```

使用外部檔案

當您從 Step Functions 狀態機器卡建立工作流程時，基礎設施編譯器會使用 `Definition` 屬性將狀態機器定義儲存在範本中。您可以設定 Infrastructure Composer，將狀態機器定義儲存在外部檔案中。

> **Note**
>
> 若要從 搭配 Infrastructure Composer 使用此功能 AWS Management Console，您必須啟用本機同步。如需詳細資訊，請參閱在 Infrastructure Composer 主控台中本機同步和儲存您的專案。

將狀態機器定義儲存至外部檔案

1. 開啟 Step Functions 狀態機器卡的資源屬性面板。
2. 選取使用外部檔案做為狀態機器定義選項。
3. 為您的狀態機器定義檔案提供相對路徑和名稱。
4. 選擇 Save (儲存)。

Infrastructure Composer 將執行下列動作：

1. 將您的狀態機器定義從 `Definition` 欄位移至外部檔案。
2. 使用 Amazon States 語言將狀態機器定義儲存在外部檔案中。
3. 使用 `DefinitionUri` 欄位修改您的範本以參考外部檔案。

進一步了解

若要進一步了解 Infrastructure Composer 中的 Step Functions，請參閱以下內容：

- 在 AWS Step Functions 開發人員指南Workflow Studio中的 Infrastructure Composer 中使用 。
- AWS Step Functions 開發人員指南中的 AWS SAM 範本中的 DefinitionSubstitutions。

Infrastructure Composer 中的標準卡

所有 AWS CloudFormation 資源都可以用作資源調色盤中的標準 IaC 資源卡。 將標準 IaC 資源卡拖曳至視覺化畫布後，就會變成標準元件卡。這只是表示卡片是一或多個標準 IaC 資源。如需更多範例和詳細資訊，請參閱本節中的主題。

您可以透過範本檢視和資源屬性視窗來修改基礎設施程式碼。例如，以下是`Alexa::ASK::Skill`標準 IaC 資源的啟動範本範例：

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

標準 IaC 資源卡啟動範本包含下列項目：

- AWS CloudFormation 資源類型。
- 必要或常用屬性。
- 為每個屬性提供的所需值類型。

> **Note**
>
> 您可以使用 Amazon Q 來產生標準資源卡的基礎設施程式碼建議。如需進一步了解，請參閱 AWS Infrastructure Composer 搭配 使用 Amazon Q Developer。

程序

您可以透過資源屬性面板修改標準元件卡中每個資源的基礎設施代碼。

修改標準元件卡

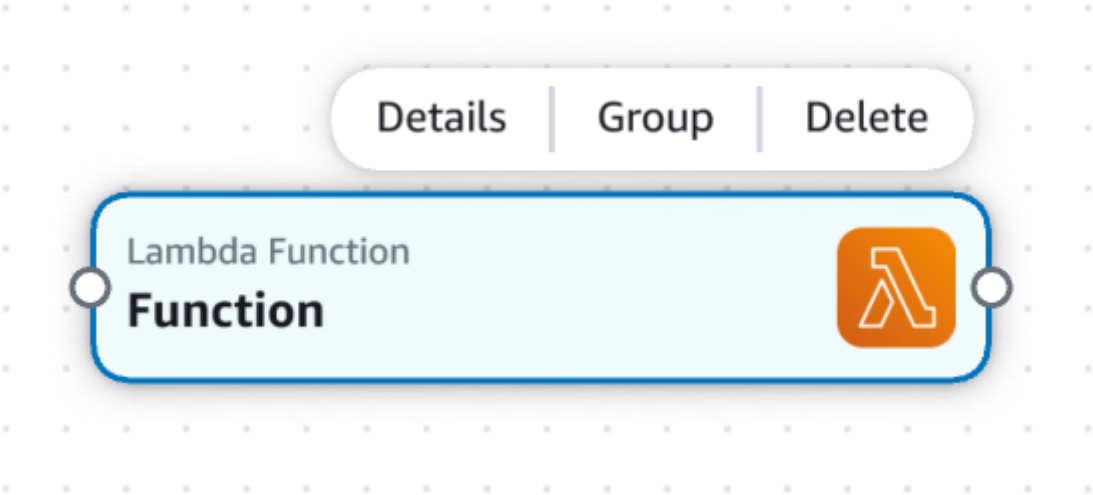
1. 開啟標準 IaC 元件卡的資源屬性面板。
2. 在編輯欄位中，從下拉式清單選取要編輯的標準 IaC 資源。
3. 修改您的基礎設施程式碼並儲存。

在 Infrastructure Composer 中刪除卡片

本節提供在 中刪除卡片的說明 AWS Infrastructure Composer。

增強型元件卡

若要刪除增強型元件卡，請選取您在視覺畫布上放置的卡片。從卡片動作功能表中，選取刪除。



標準元件卡

若要刪除標準元件卡，您必須手動從範本中移除每個 AWS CloudFormation 資源的基礎設施代碼。以下是完成此操作的簡單方法：

1. 請記下要刪除資源的邏輯 ID。
2. 在您的範本上，透過 `Resources`或 `Outputs`區段中的邏輯 ID 來尋找資源。
3. 從您的範本刪除資源。這包括資源邏輯 ID 及其巢狀值，例如 `Type`和 `Properties`。
4. 檢查 Canvas 檢視，確認資源已從畫布中移除。

使用 Infrastructure Composer 中的 Change Inspector 檢視程式碼更新

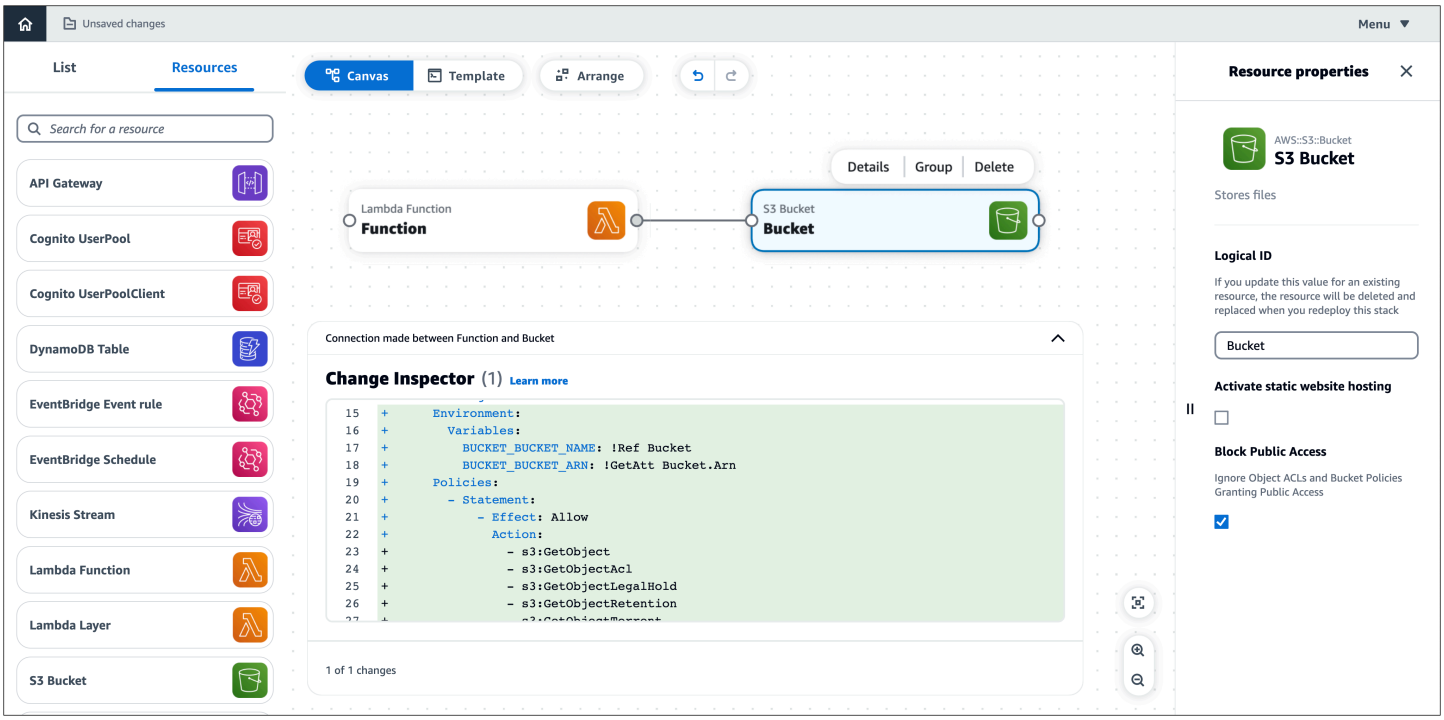
當您在 Infrastructure Composer 主控台中設計時，系統會自動建立您的基礎設施程式碼。使用 Change Inspector 來檢視範本程式碼更新，並了解 Infrastructure Composer 正在為您建立哪些項目。

本主題涵蓋從 AWS Management Console 或 AWS Toolkit for Visual Studio Code 延伸模組使用 Infrastructure Composer。

Change Inspector 是 Infrastructure Composer 中的視覺化工具，可顯示最新的程式碼更新。

- 當您設計應用程式時，訊息會顯示在視覺化畫布底部。這些訊息會針對您正在執行的動作提供註解。
- 支援時，您可以展開訊息來檢視變更檢查器。
- Change Inspector 會顯示來自您最近一次互動的程式碼變更。

下列範例示範變更檢查器的運作方式：



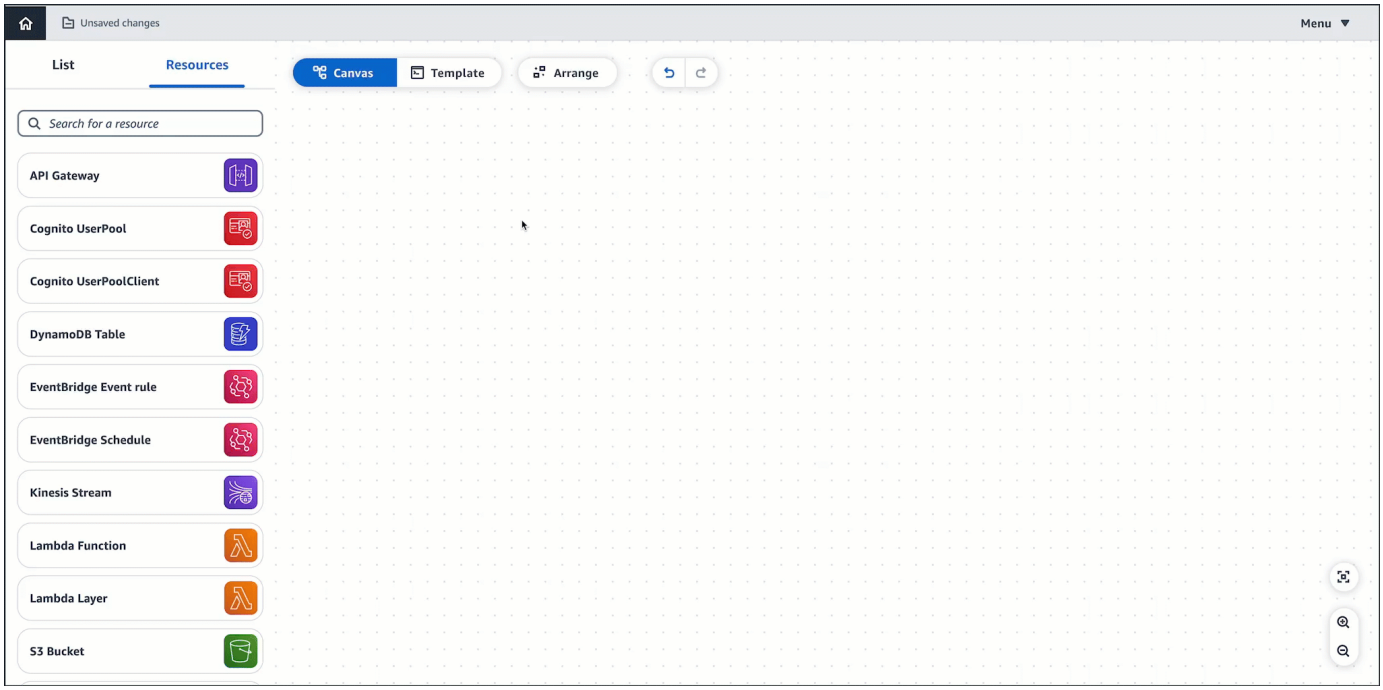
Change Inspector 的優點

Change Inspector 是檢視 Infrastructure Composer 為您建立範本程式碼的好方法。這也是學習如何撰寫基礎設施程式碼的好方法。當您在 Infrastructure Composer 中設計應用程式時，請在 Change Inspector 中檢視程式碼更新，以了解佈建設計所需的程式碼。

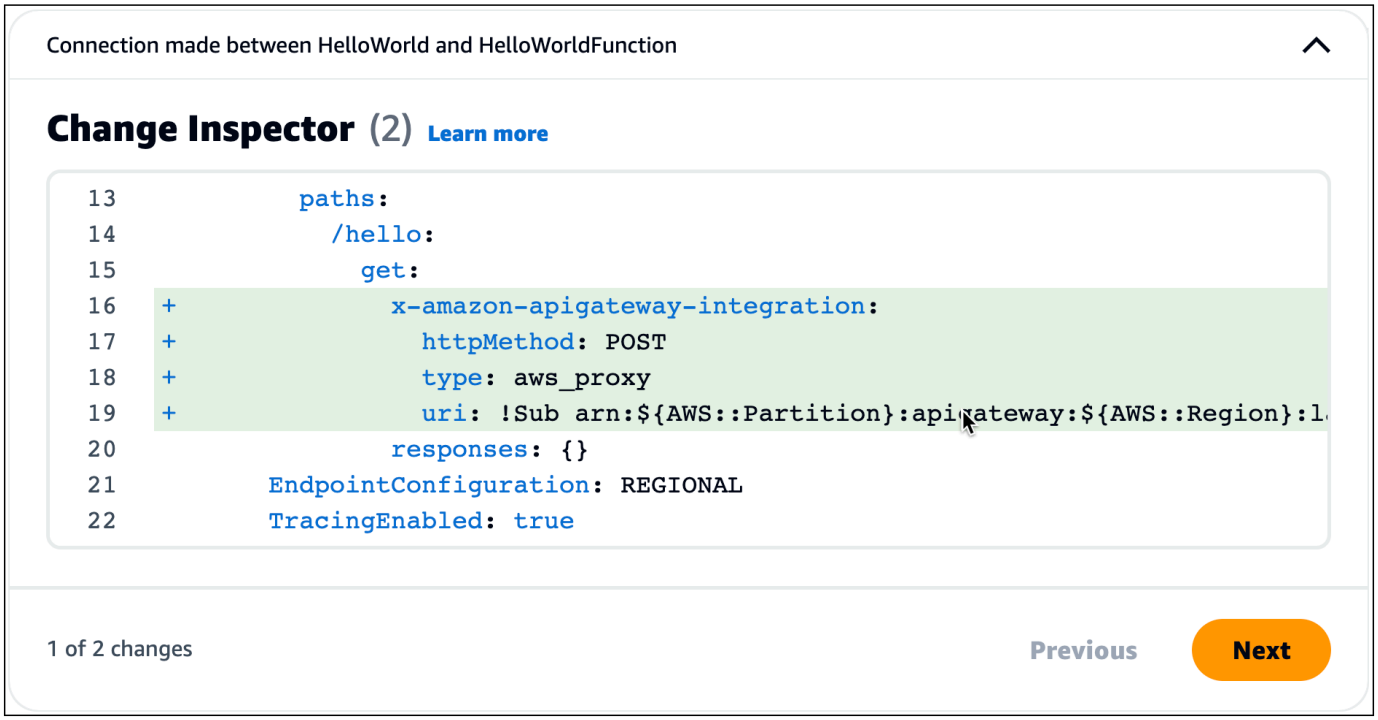
程序

使用變更檢查器

1. 展開訊息以叫出 Change Inspector。



2. 檢視為您自動編寫的程式碼。

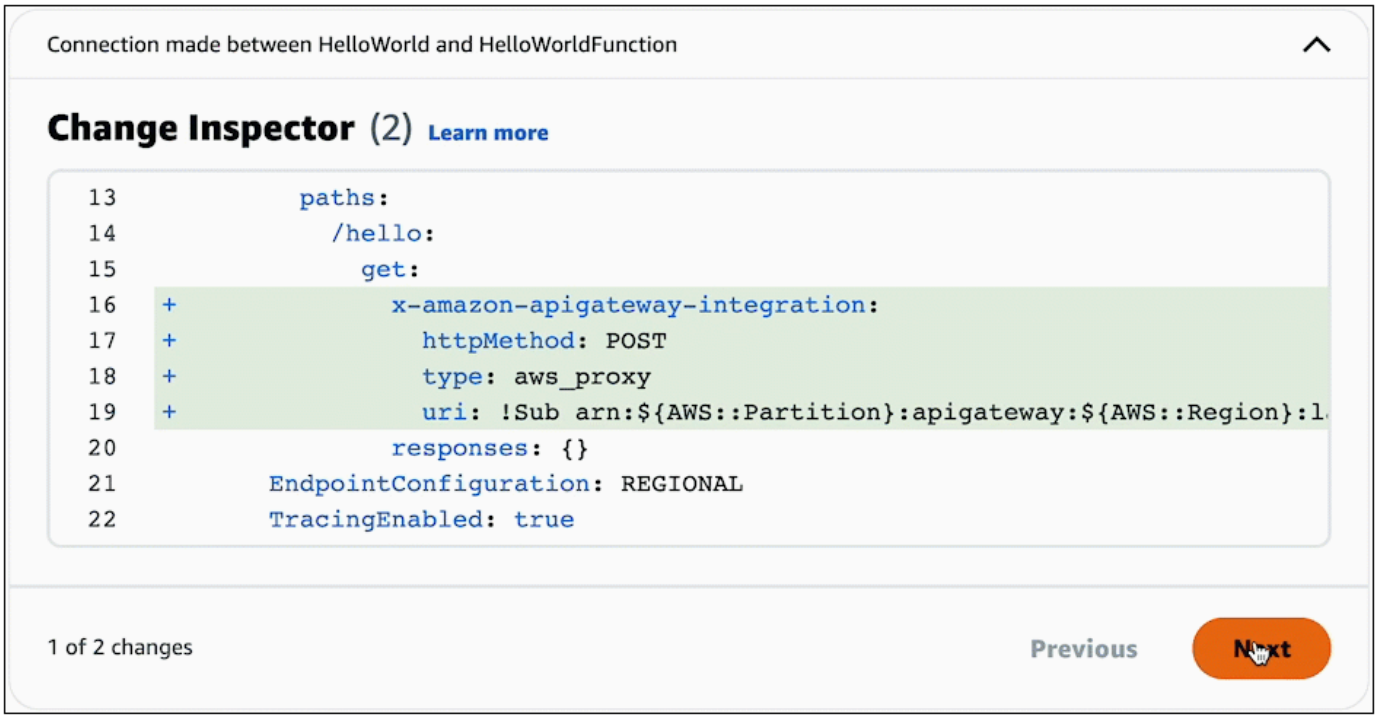


a. 反白為綠色的程式碼表示新增的程式碼。

b. 反白為紅色的程式碼表示新移除的程式碼。

c. 行號表示範本中的位置。

3. 更新範本的多個區段後，Change Inspector 會對其進行整理。選取上一個和下一個按鈕以檢視所有變更。



> **Note**
>
> 對於來自主控台的 Infrastructure Composer，您可以使用範本檢視，在整個範本的內容中檢視程式碼變更。您也可以將 Infrastructure Composer 與本機 IDE 同步，並在本機機器上檢視整個範本。如需進一步了解，請參閱 將 Infrastructure Composer 主控台與本機 IDE 連線。

進一步了解

如需 Infrastructure Composer 所建立程式碼的詳細資訊，請參閱以下內容：

- Infrastructure Composer 中的卡片連線.

參考 Infrastructure Composer 中的外部檔案

您可以使用外部檔案搭配您的 AWS Serverless Application Model (AWS SAM) 範本，重複使用重複的程式碼並整理您的專案。例如，您可能有OpenAPI規格所述的多個 Amazon API Gateway REST API 資源。您可以建立一個外部檔案，並參考每個資源，而不是複寫範本中的OpenAPI規格程式碼。

AWS Infrastructure Composer 支援下列外部檔案使用案例：

- 外部OpenAPI規格檔案定義的 API Gateway REST API 資源。
- AWS Step Functions 外部狀態機器定義檔案定義的狀態機器資源。

若要進一步了解如何為支援的資源設定外部檔案，請參閱以下內容：

- `DefinitionBody` 適用於 `AWS::Serverless::Api`。
- `DefinitionUri` 適用於 `AWS::Serverless::StateMachine`。

> **Note**
>
> 若要從 Infrastructure Composer 主控台參考具有 Infrastructure Composer 的外部檔案，您必須在本機同步模式下使用 Infrastructure Composer。如需詳細資訊，請參閱在 Infrastructure Composer 主控台中本機同步和儲存您的專案。

主題

Infrastructure Composer 外部參考檔案的最佳實務

搭配本機 IDE 使用 Infrastructure Composer

當您在本機同步模式下使用 Infrastructure Composer 搭配本機 IDE 時，您可以使用本機 IDE 來檢視和修改外部檔案。範本上參考的受支援外部檔案的內容，會在 Infrastructure Composer 畫布中自動更新。如需進一步了解，請參閱 將 Infrastructure Composer 主控台與本機 IDE 連線。

將外部檔案保留在專案的父目錄中

您可以在專案的父目錄中建立子目錄，以組織外部檔案。Infrastructure Composer 無法存取存放在專案父目錄外部目錄中的外部檔案。

使用 部署您的應用程式 AWS SAMCLI

將應用程式部署到 時 AWS 雲端，需要先將本機外部檔案上傳到可存取的位置，例如 Amazon Simple Storage Service (Amazon S3)。您可以使用 AWS SAM CLI 自動促進此程序。若要進一步了解，請參閱《 AWS Serverless Application Model 開發人員指南》中的在部署時上傳本機檔案。

在 Infrastructure Composer 中建立外部檔案參考

您可以從支援資源的資源屬性面板建立外部檔案參考。

建立外部檔案參考

1. 從 API Gateway 或 Step Functions 增強型元件卡中，選取詳細資訊以調出資源屬性面板。
2. 尋找並選取使用外部檔案選項。
3. 指定外部檔案的相對路徑。這是從`template.yaml`檔案到外部檔案的路徑。

   例如，若要參考下列專案結構中的`api-spec.yaml`外部檔案，請指定 `./api-spec.yaml`做為您的相對路徑。

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

   > **Note**
   >
   > 如果外部檔案及其指定的路徑不存在，則 Infrastructure Composer 會建立它。

4. 儲存您的變更。

在 Infrastructure Composer 中使用外部檔案參考載入專案

請依照此頁面列出的步驟，使用外部檔案參考載入 Infrastructure Composer 專案。

從 Infrastructure Composer 主控台

1. 完成列於 在 Infrastructure Composer 主控台中匯入現有的專案範本 的步驟。

2. 確認 Infrastructure Composer 會提示您連線至專案的根資料夾

如果您的瀏覽器支援檔案系統存取 API，基礎設施編譯器會提示您連線至專案的根資料夾。Infrastructure Composer 將以本機同步模式開啟您的專案，以支援您的外部檔案。如果不支援參考的外部檔案，您將會收到錯誤訊息。如需錯誤訊息的詳細資訊，請參閱 疑難排解。

從 Toolkit for VS 程式碼

1. 完成列於 從 存取 Infrastructure Composer AWS Toolkit for Visual Studio Code 的步驟。
2. 開啟您要在 Infrastructure Composer 中檢視的範本。

當您從範本存取 Infrastructure Composer 時， Infrastructure Composer 會自動偵測您的外部檔案。如果不支援參考的外部檔案，您將會收到錯誤訊息。如需錯誤訊息的詳細資訊，請參閱 疑難排解。

在 Infrastructure Composer 中建立參考外部檔案的應用程式

此範例使用 AWS SAMCLI來建立參考其狀態機器定義的外部檔案的應用程式。然後，在 Infrastructure Composer 中載入專案，並正確參考外部檔案。

範例

1. 首先，使用 AWS SAM CLI sam init命令初始化名為 的新應用程式demo。在互動式流程中，選取多步驟工作流程快速啟動範本。

```
$ 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

...
```

此應用程式參考狀態機器定義的外部檔案。

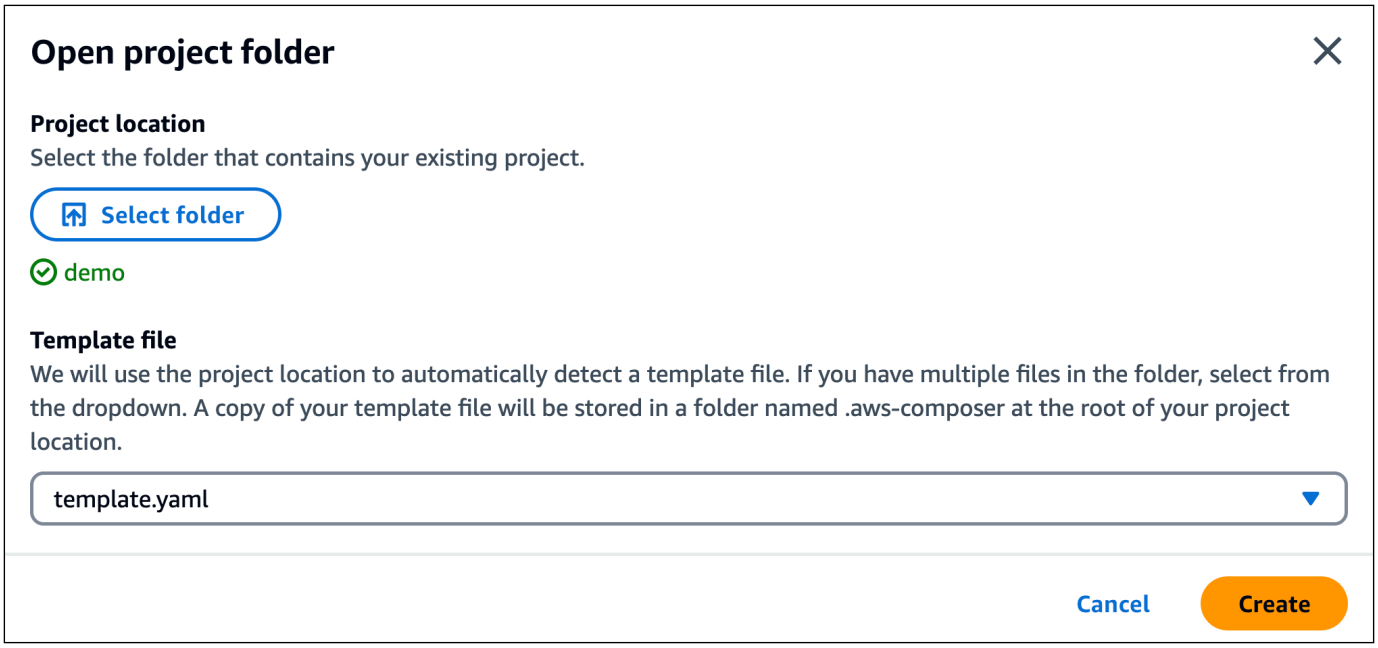
```
...
```

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

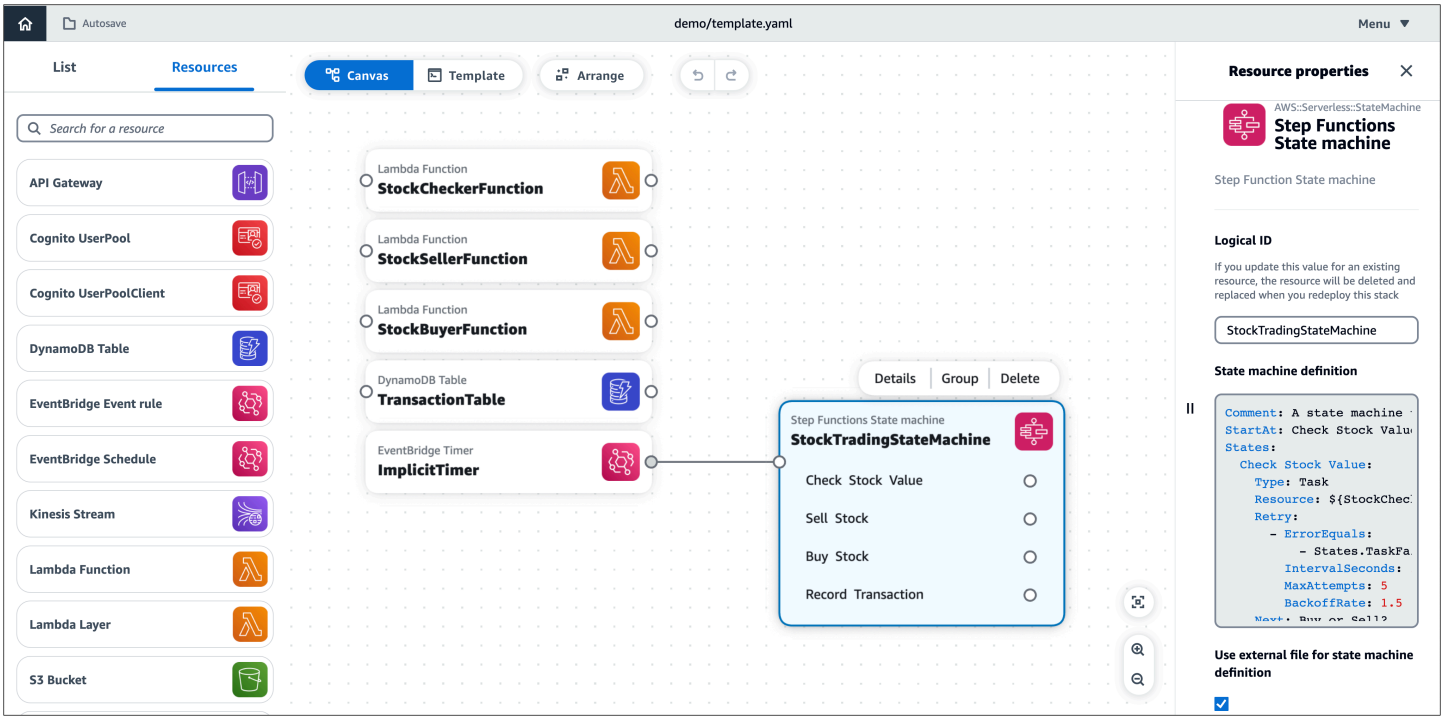
外部檔案位於我們應用程式的`statemachine`子目錄中。

```
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. 接著，從主控台在 Infrastructure Composer 中載入您的應用程式。在基礎設施撰寫者首頁中，選取載入 CloudFormation 範本。
3. 選取我們的`demo`專案資料夾，並允許 提示檢視檔案。選取我們的`template.yaml`檔案，然後選取建立。出現提示時，選取儲存變更。



Infrastructure Composer 會自動偵測外部狀態機器定義檔案，並將其載入。選取我們的 StockTradingStateMachine 資源，然後選擇詳細資訊以顯示資源屬性面板。在這裡，您可以看到 Infrastructure Composer 已自動連線到我們的外部狀態機器定義檔案。



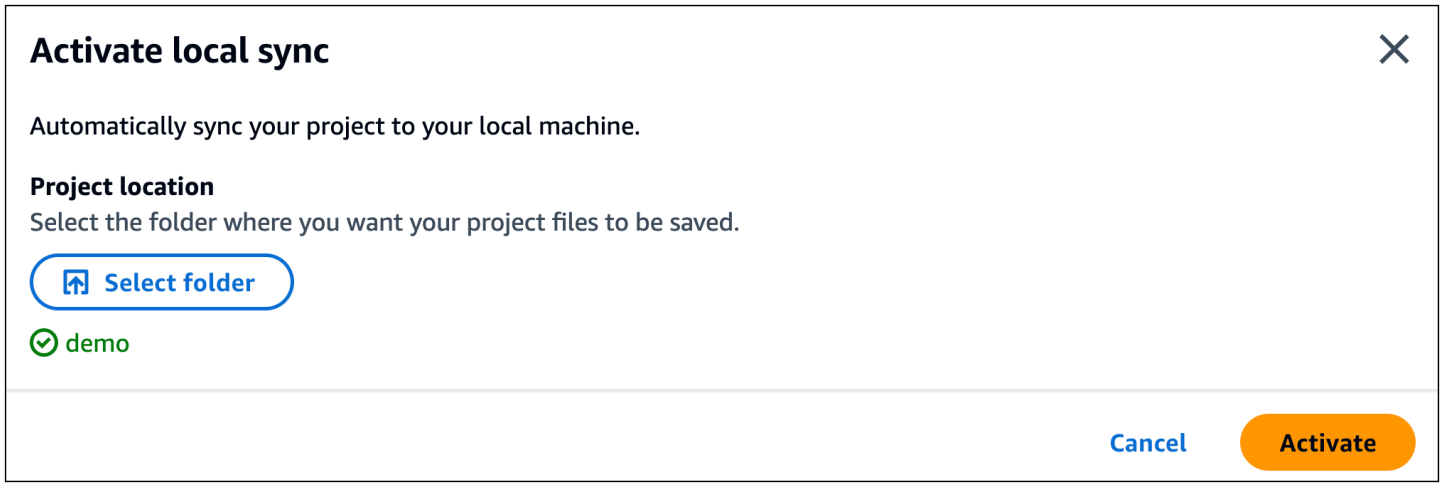
狀態機器定義檔案所做的任何變更都會自動反映在 Infrastructure Composer 中。

使用 Infrastructure Composer 參考OpenAPI規格外部檔案

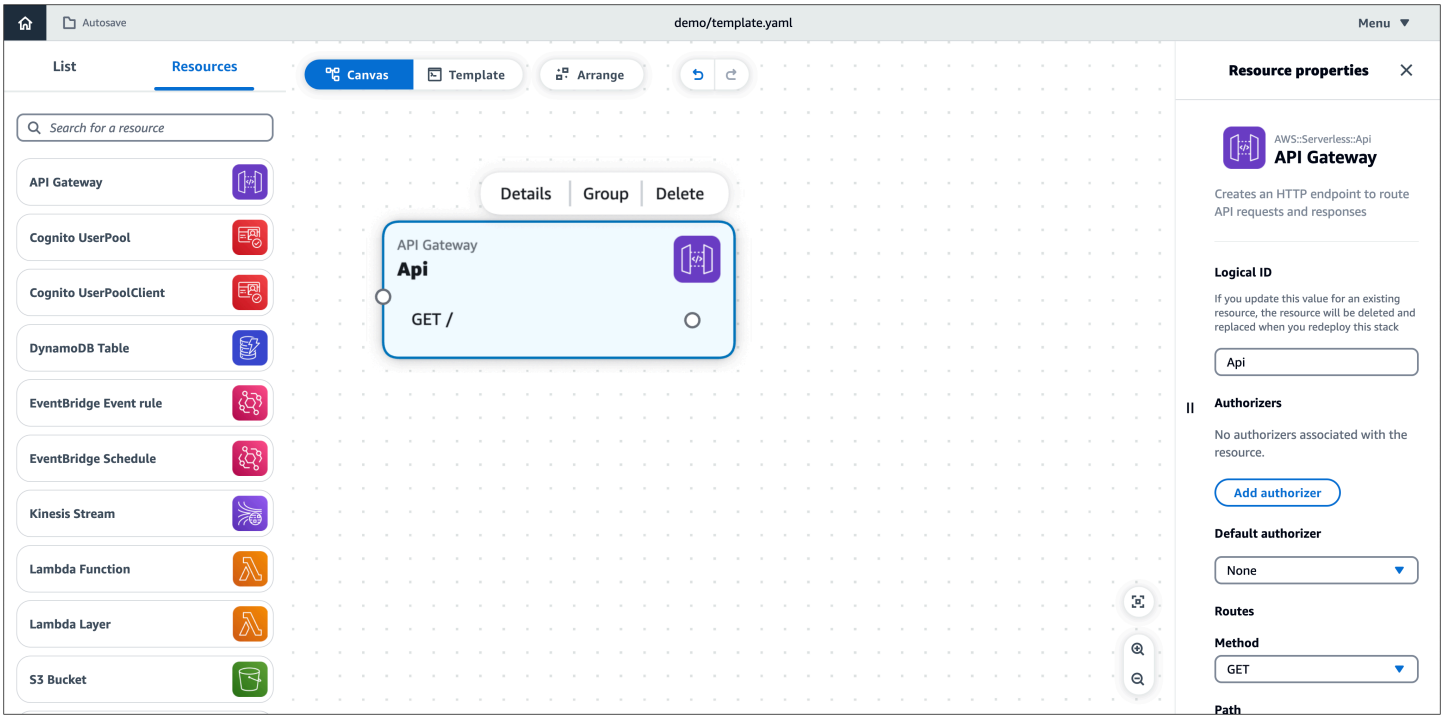
此範例使用主控台的 Infrastructure Composer 來參考定義 API Gateway 的外部OpenAPI規格檔案 REST API。

首先，從 Infrastructure Composer 首頁建立新專案。

接著，從選單中選取啟用本機同步來啟用本機同步。建立新的名為 的資料夾demo，允許提示檢視檔案，然後選取啟用。出現提示時，選取儲存變更。

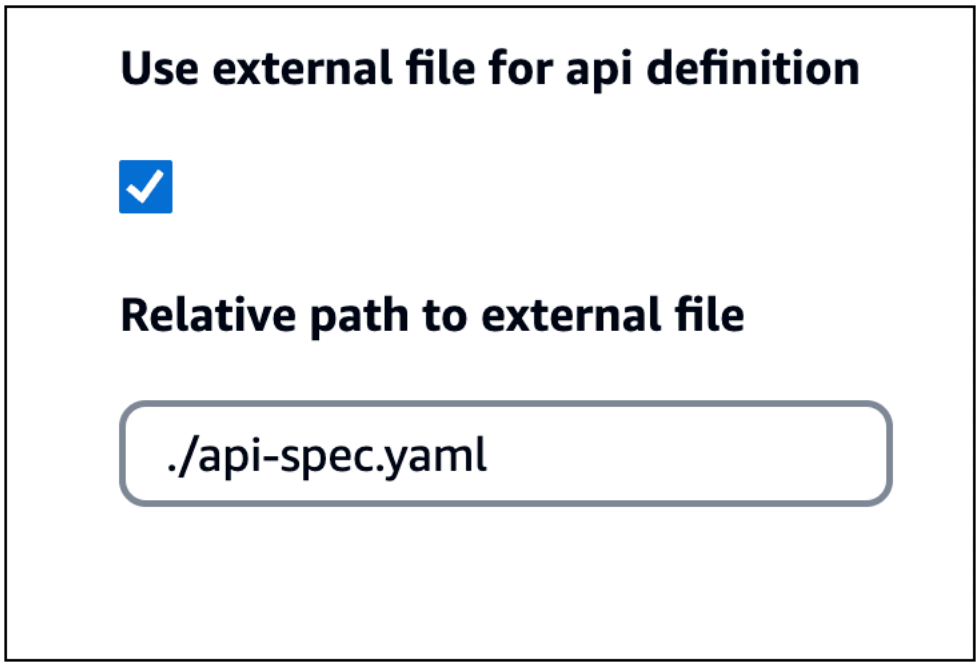


接著，將 Amazon API Gateway 卡拖曳到畫布上。選取詳細資訊，以顯示資源屬性面板。



從資源屬性面板中，設定下列項目並儲存。

- 選取使用外部檔案進行 api 定義選項。
- 輸入 `./api-spec.yaml`做為外部檔案的相對路徑



這會在我們的本機機器上建立下列目錄：

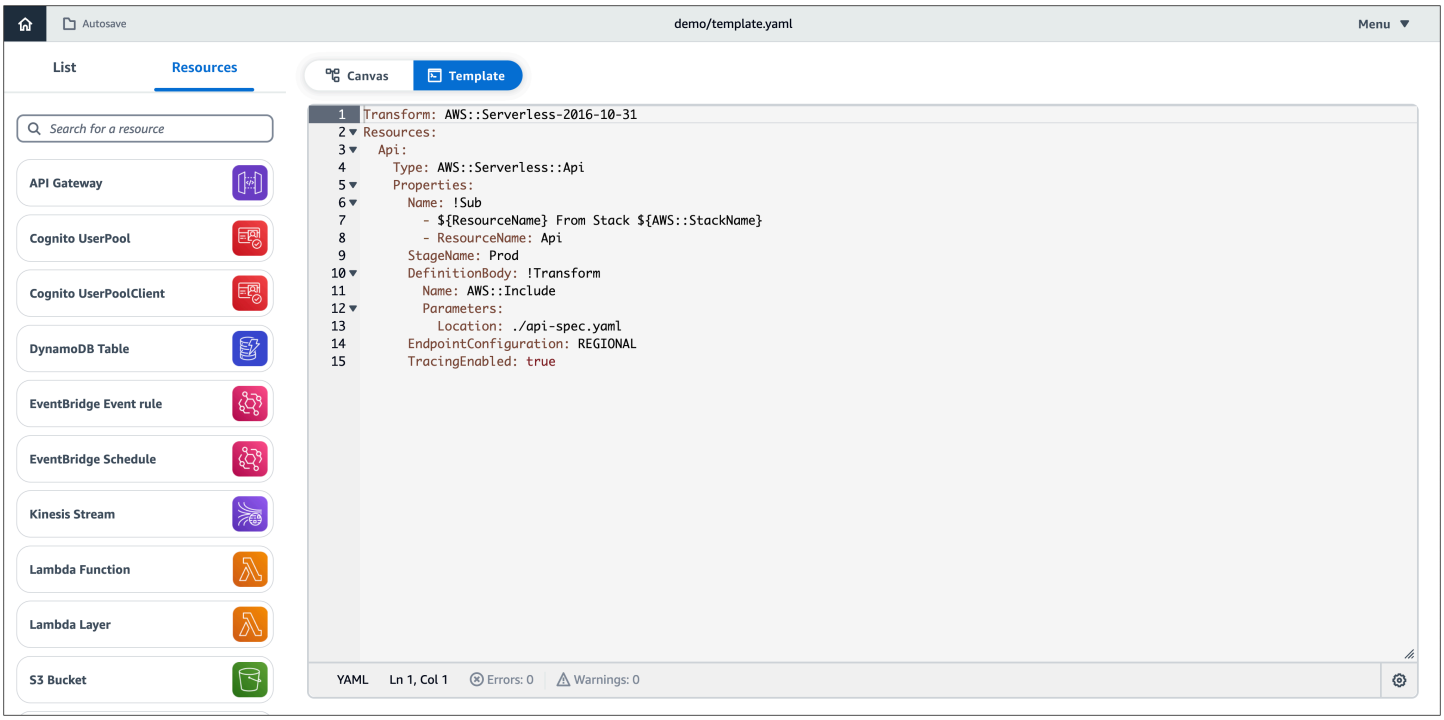
```
demo
### api-spec.yaml
```

現在，您可以在我們的本機機器上設定外部檔案。使用我們的 IDE，開啟`api-spec.yaml`位於您專案資料夾中的 。將其內容取代為下列項目：

```
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
```

在 Infrastructure Composer 範本檢視中，您可以看到 Infrastructure Composer 已自動更新範本以參考外部檔案。



將基礎設施編譯器與 Amazon Virtual Private Cloud (Amazon VPC) 整合

AWS Infrastructure Composer 具有與 Amazon Virtual Private Cloud (Amazon VPC) 服務的整合。使用 Infrastructure Composer，您可以執行下列動作：

- 透過視覺化 VPC 標籤，識別畫布上 VPC 中的資源。
- 從外部範本使用 VPCs 設定 AWS Lambda 函數。

下圖顯示使用 VPC 設定 Lambda 函數的應用程式範例。

若要進一步了解 Amazon VPC，請參閱《Amazon VPC 使用者指南》中的什麼是 Amazon VPC？。

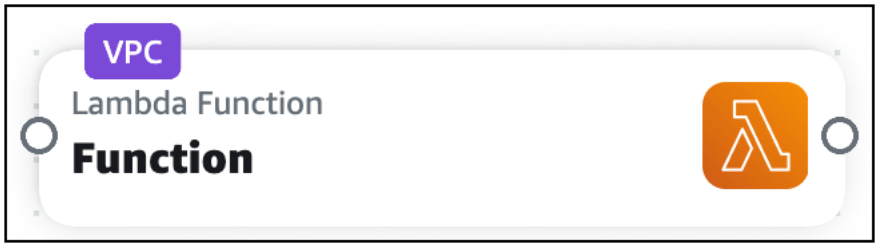
主題

- 識別 VPC 中的 Infrastructure Composer 資源和相關資訊
- 在 Infrastructure Composer 中使用外部 VPCs 設定 Lambda 函數
- 具有 Infrastructure Composer 之外部 VPC 的匯入範本中的參數
- 使用 Infrastructure Composer 將新參數新增至匯入的範本
- 使用 Infrastructure Composer 設定 Lambda 函數和在另一個範本中定義的 VPC

識別 VPC 中的 Infrastructure Composer 資源和相關資訊

若要將 Infrastructure Composer 與 Amazon VPC 整合，您必須先識別 VPC 中的資源，以及完成整合所需的資訊。這也包括與安全群組、子網路識別符、參數類型、SSM 類型、靜態值類型相關的組態資訊。

Infrastructure Composer 使用 VPC 標籤視覺化 VPC 中的資源。此標籤會套用至畫布上的卡片。以下是具有 VPC 標籤的 Lambda 函數範例：



當您執行下列動作時，VPC 標籤會套用至畫布上的卡片：

- 在 Infrastructure Composer 中使用 VPC 設定 Lambda 函數。
- 匯入範本，其中包含使用 VPC 設定的資源。

安全群組和子網路識別符

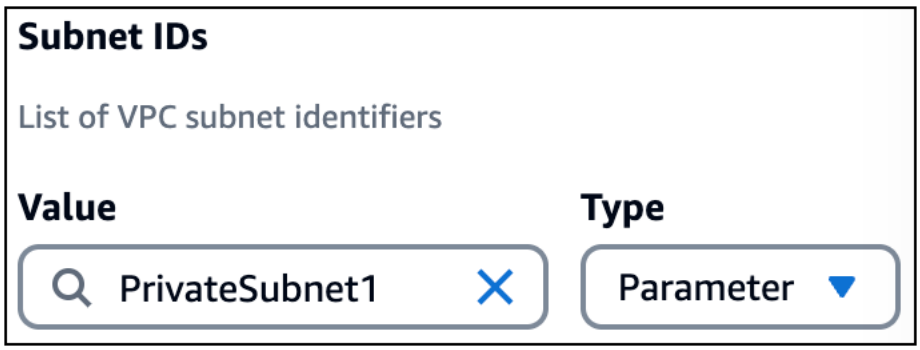
Lambda 函數可以設定多個安全群組和子網路。若要設定 Lambda 函數的安全群組或子網路，請提供值和類型。

- 值 – 安全群組或子網路的識別符。接受的值會根據類型而有所不同。
- 類型 – 允許下列類型的值：
  - 參數名稱
  - AWS Systems Manager (SSM) 參數存放區
  - 靜態值

參數類型

AWS CloudFormation 範本的 `Parameters`區段可用來跨多個範本存放資源資訊。如需參數的詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的參數。

對於 參數類型，您可以提供參數名稱。在下列範例中，我們提供`PrivateSubnet1`參數名稱值：



當您提供參數名稱時， Infrastructure Composer 會在範本的 Parameters區段中定義它。然後， Infrastructure Composer 會參考 Lambda 函數資源中的 參數。以下是範例：

```
...
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 類型

SSM 參數存放區為組態資料管理和秘密管理提供安全的階層式儲存。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的 AWS Systems Manager 參數存放區。

對於 SSM 類型，您可以提供下列值：

- 來自 SSM 參數存放區的 值動態參考。
- 範本中定義的AWS::SSM::Parameter資源邏輯 ID。

動態參考

您可以使用下列格式的動態參考，從 SSM 參數存放區參考值：{{resolve:ssm:reference-key}}。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的 SSM 參數。

Infrastructure Composer 會建立基礎設施程式碼，以使用來自 SSM 參數存放區的值來設定 Lambda 函數。以下是範例：

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

```
      ...
      VpcConfig:
        SecurityGroupIds:
          - '{{resolve:ssm:demo-app/sg-0b61d5c742dc2c773}}'
  ...
```

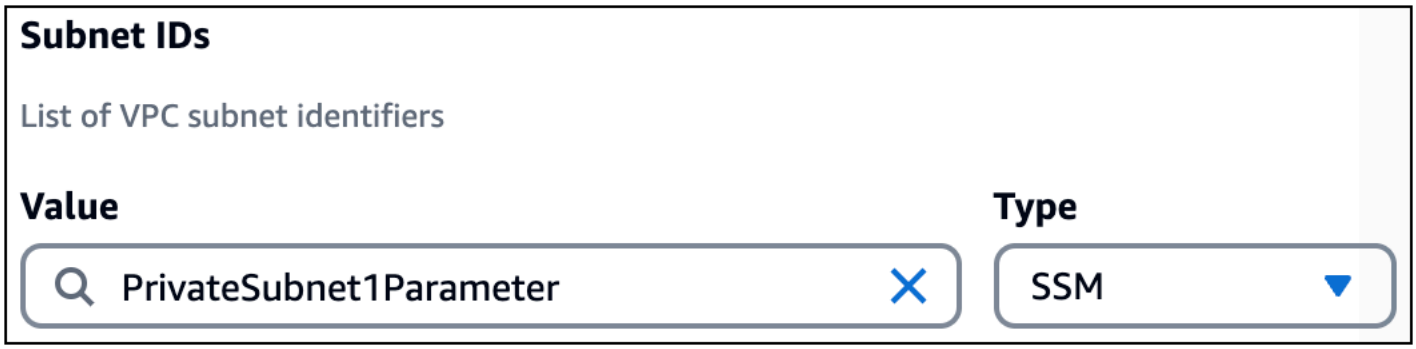
邏輯 ID

您可以透過邏輯 ID 來參考相同範本中的`AWS::SSM::Parameter`資源。

以下是名為 的資源範例`PrivateSubnet1Parameter`，該`AWS::SSM::Parameter`資源會存放 的子網路 ID`PrivateSubnet1`：

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

以下是由 Lambda 函數的邏輯 ID 提供此資源值的範例：



Infrastructure Composer 會建立基礎設施程式碼，以使用 SSM 參數設定 Lambda 函數：

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

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

靜態值類型

部署安全群組或子網路時 AWS CloudFormation，會建立 ID 值。您可以將此 ID 做為靜態值提供。

對於靜態值類型，下列是有效值：

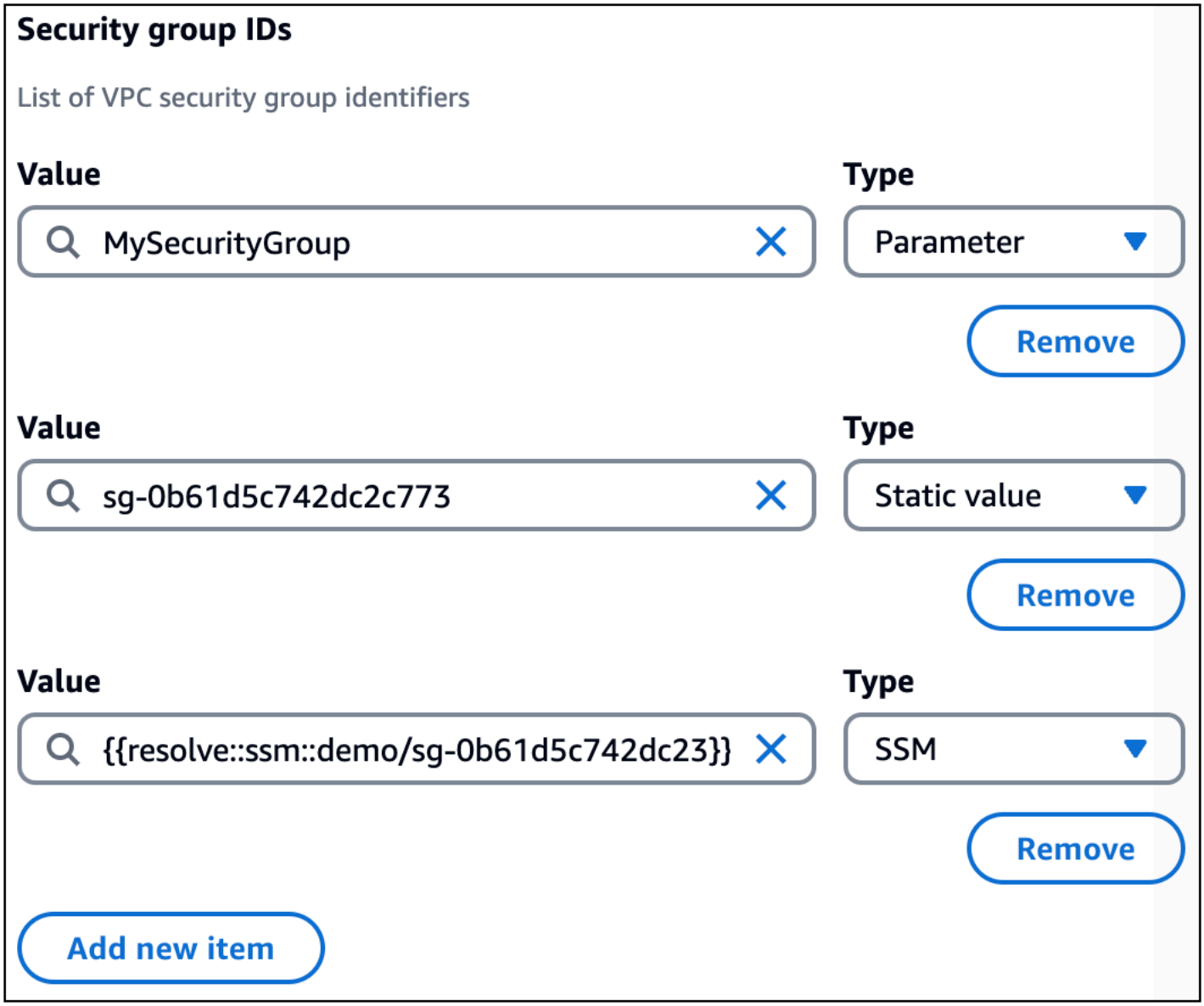
- 對於安全群組，請提供 `GroupId`。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的傳回值。以下是範例：`sg-0b61d5c742dc2c773`
- 對於子網路，請提供 `SubnetId`。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的傳回值。以下是範例：`subnet-01234567890abcdef`

Infrastructure Composer 會建立基礎設施程式碼，以使用靜態值設定 Lambda 函數。以下是範例：

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

使用多種類型

對於安全群組和子網路，您可以同時使用多種類型。以下是透過提供不同類型值，為 Lambda 函數設定三個安全群組的範例：



Infrastructure Composer 參考 `SecurityGroupIds` 屬性下的所有三個值：

```
...
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
```

在 Infrastructure Composer 中使用外部 VPCs 設定 Lambda 函數

若要使用在另一個範本上定義的 VPC 開始設定 Lambda 函數，請使用 Lambda 函數增強型元件卡。此卡代表使用 AWS Serverless Application Model (AWS SAM) `AWS::Serverless::Function` 資源類型的 Lambda 函數。

從外部範本使用 VPC 設定 Lambda 函數

1. 從 Lambda 函數資源屬性面板中，展開 VPC 設定 （進階） 下拉式清單區段。
2. 選取指派給外部 VPC。
3. 提供要為 Lambda 函數設定的安全群組和子網路值。如需詳細資訊，請參閱 安全群組和子網路識別符。
4. 儲存您的變更。

具有 Infrastructure Composer 之外部 VPC 的匯入範本中的參數

當您匯入具有為外部 VPC 的安全群組和子網路定義的參數的現有範本時， Infrastructure Composer 會提供下拉式清單來選取參數。

以下是匯入範本的 `Parameters`區段範例：

```
...
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
...
```

在畫布上為新的 Lambda 函數設定外部 VPC 時，這些參數可從下拉式清單中取得。以下是範例：

匯入清單參數類型的限制

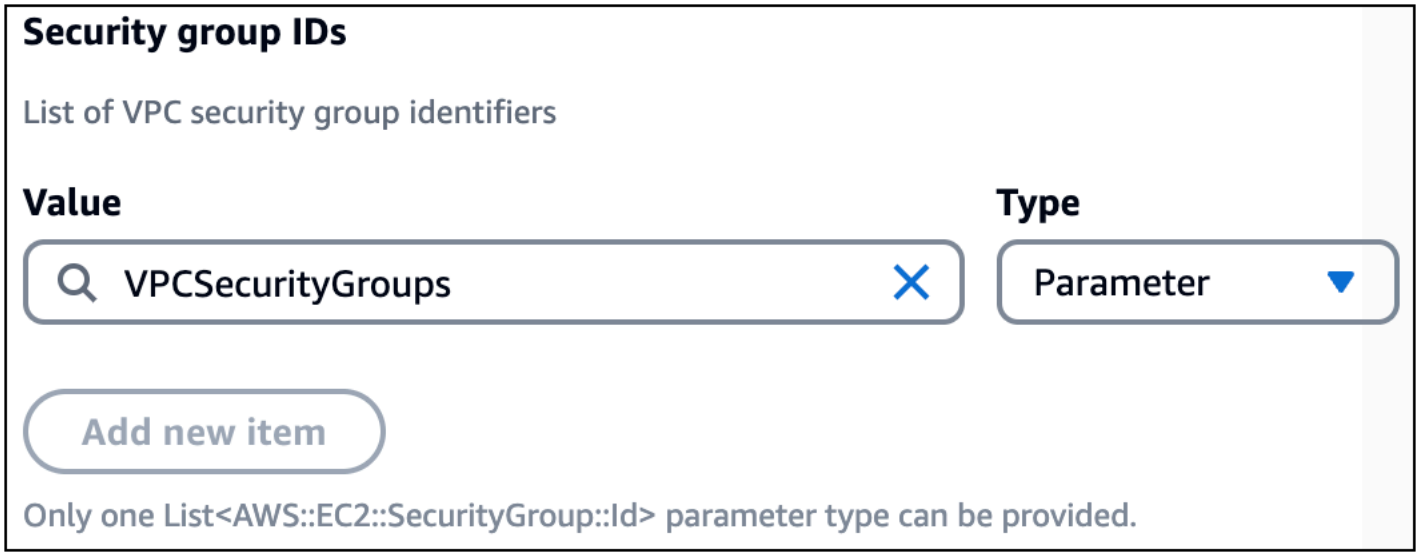
一般而言，您可以為每個 Lambda 函數指定多個安全群組和子網路識別符。如果您現有的範本包含清單參數類型，例如 `List<AWS::EC2::SecurityGroup::Id>`或 `List<AWS::EC2::Subnet::Id>`，您只能指定一個識別符。

如需參數清單類型的詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的支援 AWS的特定參數類型。

以下是將 `VPCSecurityGroups`定義為清單參數類型的範本範例：

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

在 Infrastructure Composer 中，如果您選取`VPCSecurityGroups`值做為 Lambda 函數的安全群組識別符，您會看到下列訊息：



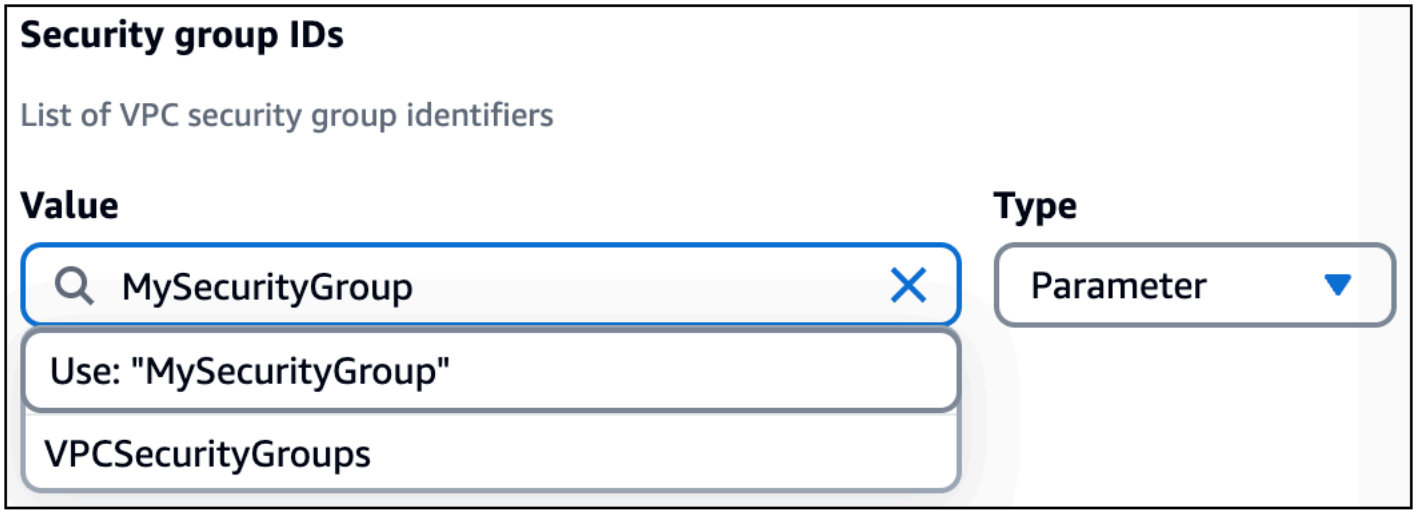
發生此限制是因為AWS::Lambda::Function VpcConfig物件的 SecurityGroupIds和 SubnetIds 屬性都只接受字串值的清單。由於單一清單參數類型包含字串清單，因此它可以是指定時提供的唯一物件。

對於清單參數類型，以下是使用 Lambda 函數設定時如何在範本中定義參數的範例：

```
...
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
```

使用 Infrastructure Composer 將新參數新增至匯入的範本

當您匯入已定義參數的現有範本時，您也可以建立新的參數。提供新的類型和值，而不是從下拉式清單中選取現有的參數。以下是建立名為 之新參數的範例MySecurityGroup：



對於您在 Lambda 函數的資源屬性面板中提供的所有新值， Infrastructure Composer 會在 Lambda 函數的 SecurityGroupIds或 SubnetIds 屬性下的清單中定義它們。以下是範例：

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

如果您想要參考外部範本中清單參數類型的邏輯 ID，建議您使用範本檢視並直接修改範本。清單參數類型的邏輯 ID 應一律提供為單一值，且是唯一的值。

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

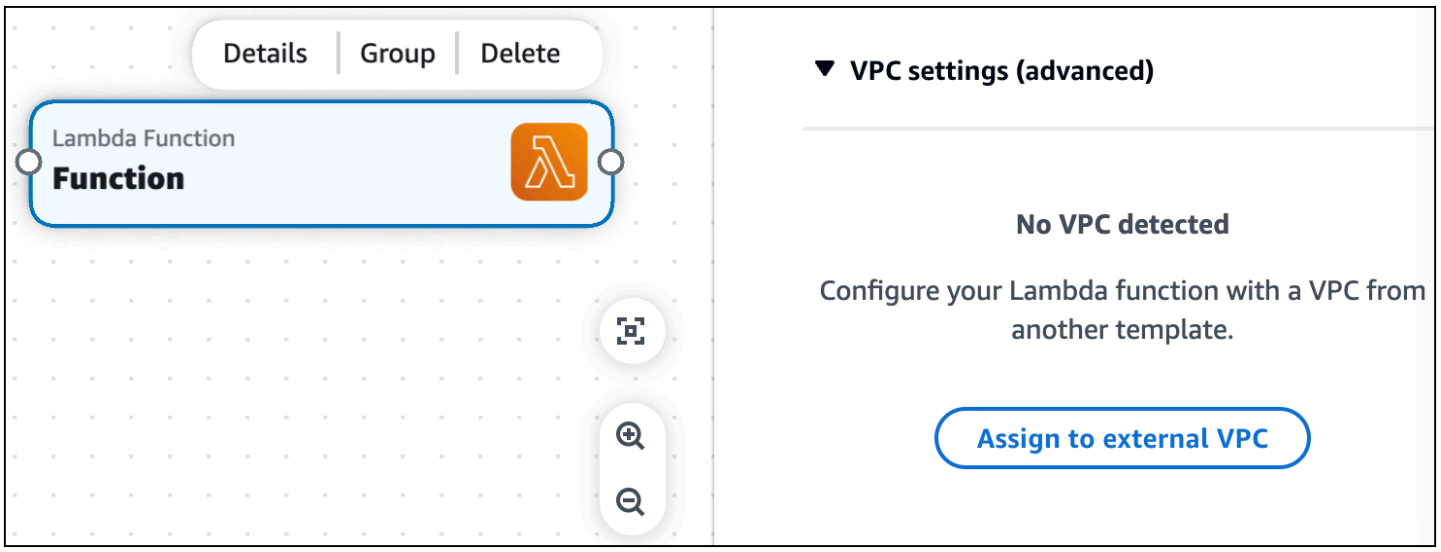
```
  VPCSubnets:
    Description: Subnet IDs generated by Infrastructure Composer
    Type: List<AWS::EC2::Subnet::Id>
Resources:
  ...
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      ...
      VpcConfig:
        SecurityGroupIds: !Ref VPCSecurityGroups # Valid syntax
      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

使用 Infrastructure Composer 設定 Lambda 函數和在另一個範本中定義的 VPC

在此範例中，我們會在 Infrastructure Composer 中設定 Lambda 函數，並在另一個範本上定義 VPC。

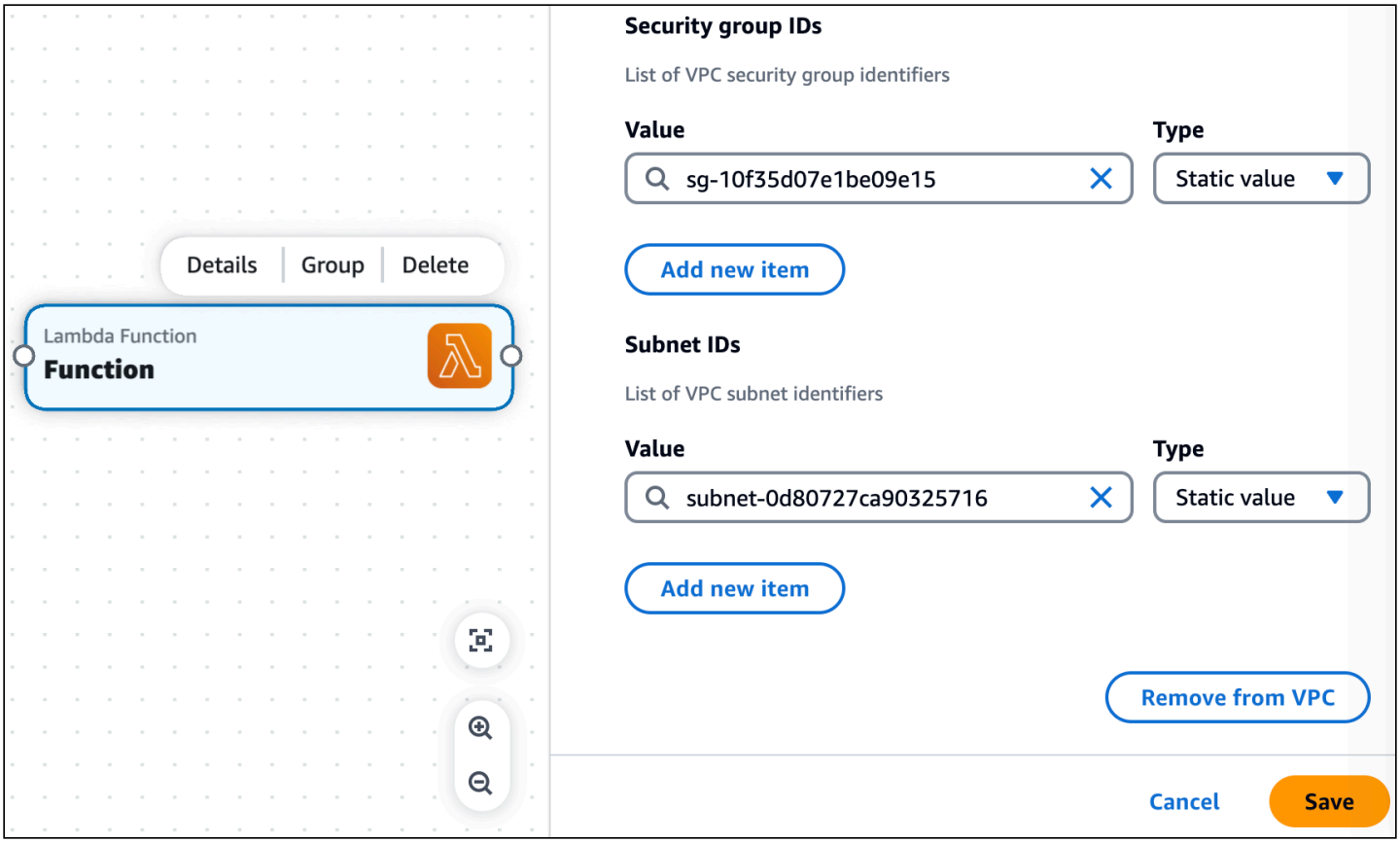
首先，我們將 Lambda Function 增強型元件卡拖曳到畫布上。

接著，我們開啟卡片的資源屬性面板，並展開 VPC 設定 （進階） 下拉式清單區段。

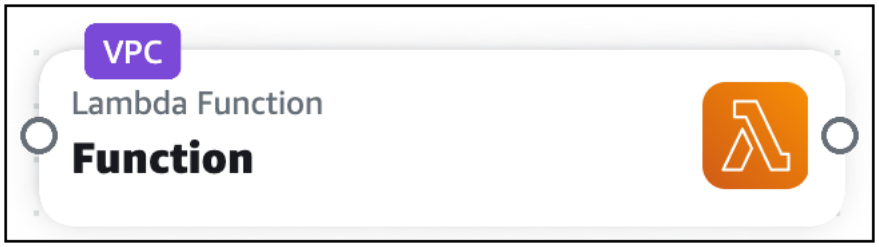


接下來，我們選取指派給外部 VPC 以開始從外部範本設定 VPC。

在此範例中，我們參考安全群組 ID 和子網路 ID。這些值會在部署定義 VPC 的範本時建立。我們選擇靜態值類型，並輸入 IDs的值。完成後，我們會選取儲存。



現在我們的 Lambda 函數已設定我們的 VPC，VPC 標籤會顯示在我們的卡片上。



Infrastructure Composer 已建立基礎設施程式碼，以使用外部 VPC 的安全群組和子網路來設定 Lambda 函數。

```
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}
```

將您的 Infrastructure Composer 無伺服器應用程式部署至 AWS 雲端

使用 AWS Infrastructure Composer 設計可立即部署的無伺服器應用程式。若要部署，請使用任何 AWS CloudFormation 相容的 服務。建議使用 AWS Serverless Application Model (AWS SAM)。

AWS SAM 是開放原始碼架構，提供開發人員工具來建置和執行無伺服器應用程式 AWS。使用 AWS SAM的速記語法，開發人員會宣告 AWS CloudFormation 資源和專用無伺服器資源，這些資源會在部署期間轉換為基礎設施。

重要 AWS SAM 概念

在使用之前 AWS SAM，請務必先熟悉其一些基本概念。

- AWS SAM 運作方式：此主題位於 AWS Serverless Application Model 開發人員指南中，提供用於建立無服務應用程式的主要元件的重要資訊： AWS SAMCLI、 AWS SAM 專案和 AWS SAM 範本。
- 如何使用 AWS Serverless Application Model (AWS SAM)：此主題位於 AWS Serverless Application Model 開發人員指南中，提供您需要完成的步驟的高階概觀，以 AWS SAM 將應用程式部署到 AWS 雲端。

當您在 Infrastructure Composer 中設計應用程式時，您可以使用 sam sync命令讓 AWS SAMCLI自動偵測本機變更，並將這些變更部署至 AWS CloudFormation。若要進一步了解，請參閱《 AWS Serverless Application Model 開發人員指南》中的使用 sam 同步。

後續步驟

請參閱 以設定 以使用 AWS SAMCLI和 Infrastructure Composer 部署準備部署您的應用程式。

設定 以使用 AWS SAMCLI和 Infrastructure Composer 部署

若要使用 部署應用程式 AWS SAM，您首先需要安裝和存取 AWSCLI和 AWS SAMCLI。本節中的主題提供執行此操作的詳細資訊。

安裝 AWSCLI

建議您先安裝和設定 ， AWSCLI再安裝 AWS SAMCLI。如需說明，請參閱AWS Command Line Interface 《 使用者指南》中的安裝或更新至最新版本的 AWS CLI。

> **Note**
>
> 安裝 之後 AWSCLI，您必須設定 AWS 登入資料。若要進一步了解，請參閱AWS Command Line Interface 《 使用者指南》中的快速設定。

安裝 AWS SAMCLI

若要安裝 AWS SAMCLI，請參閱《 AWS SAM開發人員指南》中的安裝CLI 。 AWS Serverless Application Model

存取 AWS SAMCLI

如果您從 使用 Infrastructure Composer AWS Management Console，您有下列選項可以使用 AWS SAMCLI。

啟用本機同步模式

使用本機同步模式時，您的專案資料夾，包括 AWS SAM 範本，會自動儲存至您的本機機器。Infrastructure Composer 會以 AWS SAM 可識別的方式建構您的專案目錄。您可以從專案的AWS SAMCLI根目錄執行 。

如需本機同步模式的詳細資訊，請參閱在 Infrastructure Composer 主控台中本機同步和儲存您的專案。

匯出您的範本

您可以將範本匯出至本機機器。然後， AWS SAMCLI從包含範本的父資料夾執行 。您也可以搭配任何 AWS SAMCLI命令使用 `--template-file`選項，並提供範本的路徑。

從 使用 Infrastructure Composer AWS Toolkit for Visual Studio Code

您可以從 Toolkit for VS Code 使用 Infrastructure Composer，將 Infrastructure Composer 帶入本機機器。然後，使用 Infrastructure Composer 和 AWS SAMCLI VS Code 中的 。

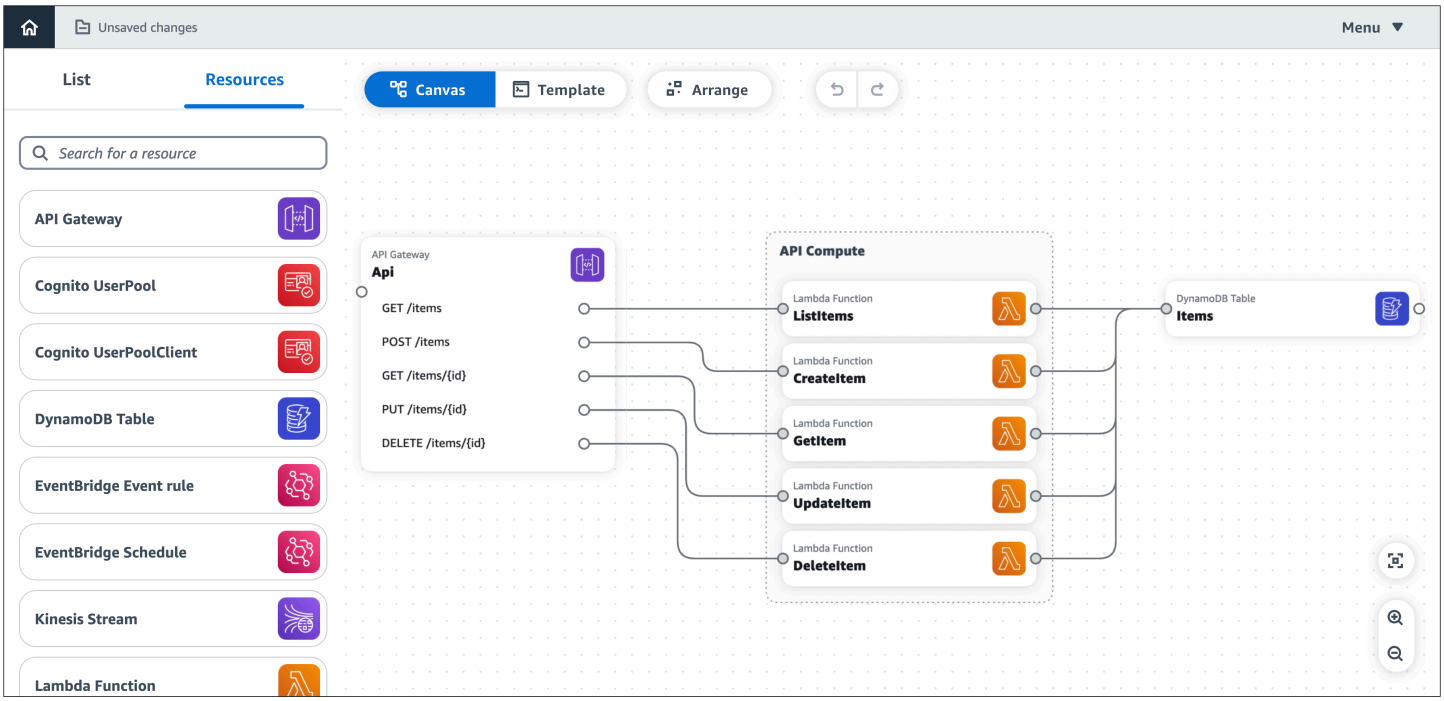
後續步驟

若要部署您的應用程式，請參閱 搭配 使用 Infrastructure Composer AWS SAM 來建置和部署。

搭配 使用 Infrastructure Composer AWS SAM 來建置和部署

現在您已完成 設定 以使用 AWS SAMCLI和 Infrastructure Composer 部署，您可以使用 AWS SAM 和 Infrastructure Composer 部署應用程式。本節提供詳細說明如何執行此操作的範例。您也可以參閱《AWS Serverless Application Model 開發人員指南》中的使用 部署您的應用程式和資源 AWS SAM，以取得使用 部署應用程式的指示 AWS SAM。

此範例說明如何建置和部署 Infrastructure Composer 示範應用程式。示範應用程式具有下列資源：



> **Note**
>
> - 若要進一步了解示範應用程式，請參閱載入和修改 Infrastructure Composer 示範專案。
> - 在此範例中，我們使用已啟用本機同步的 Infrastructure Composer。

1. 使用 sam build命令來建置應用程式。

```
$ 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
```

會在專案資料夾中 AWS SAMCLI建立`./aws-sam`目錄。此目錄包含應用程式 Lambda 函數的建置成品。以下是專案目錄的輸出：

```
.
### 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. 現在，應用程式已準備好進行部署。我們將使用 sam deploy --guided。這可讓您的應用程式準備好透過一系列提示進行部署。

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

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

```
    Setting default arguments for 'sam deploy'
    =========================================
    Stack Name [aws-app-composer-basic-api]:
    AWS Region [us-west-2]:
    #Shows you resources changes to be deployed and require a 'Y' to initiate
 deploy
    Confirm changes before deploy [y/N]:
    #SAM needs permission to be able to create roles to connect to the resources in
 your template
    Allow SAM CLI IAM role creation [Y/n]:
    #Preserves the state of previously provisioned resources when an operation
 fails
    Disable rollback [y/N]:
    ListItems may not have authorization defined, Is this okay? [y/N]: y
    CreateItem may not have authorization defined, Is this okay? [y/N]: y
    GetItem may not have authorization defined, Is this okay? [y/N]: y
    UpdateItem may not have authorization defined, Is this okay? [y/N]: y
    DeleteItem may not have authorization defined, Is this okay? [y/N]: y
    Save arguments to configuration file [Y/n]:
    SAM configuration file [samconfig.toml]:
    SAM configuration environment [default]:
```

AWS SAMCLI 會顯示將部署項目的摘要：

```
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-samcliamzn-s3-demo-
bucket-1b3x26zbcdkqr
    Capabilities                 : ["CAPABILITY_IAM"]
    Parameter overrides          : {}
    Signing Profiles             : {}
```

會先建立 AWS CloudFormation 變更集來 AWS SAMCLI部署應用程式：

```
Initiating deployment
=====================
```

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

Waiting for changeset to be created..
CloudFormation stack changeset
-----------------------------------------------------------------------------------------
Operation                               LogicalResourceId
 ResourceType                            Replacement
-----------------------------------------------------------------------------------------
+ Add                                   ApiDeploymentcc153d135b
 AWS::ApiGateway::Deployment            N/A
+ Add                                   ApiProdStage
 AWS::ApiGateway::Stage                 N/A
+ Add                                   Api
 AWS::ApiGateway::RestApi               N/A
+ Add                                   CreateItemApiPOSTitemsPermissionP
 AWS::Lambda::Permission                N/A
                                        rod
+ Add                                   CreateItemRole
 AWS::IAM::Role                         N/A
+ Add                                   CreateItem
 AWS::Lambda::Function                  N/A
+ Add                                   DeleteItemApiDELETEitemsidPermiss
 AWS::Lambda::Permission                N/A
                                        ionProd
+ Add                                   DeleteItemRole
 AWS::IAM::Role                         N/A
+ Add                                   DeleteItem
 AWS::Lambda::Function                  N/A
+ Add                                   GetItemApiGETitemsidPermissionPro
 AWS::Lambda::Permission                N/A
                                        d
+ Add                                   GetItemRole
 AWS::IAM::Role                         N/A
+ Add                                   GetItem
 AWS::Lambda::Function                  N/A
+ Add                                   Items
 AWS::DynamoDB::Table                   N/A
+ Add                                   ListItemsApiGETitemsPermissionPro
 AWS::Lambda::Permission                N/A
                                        d
+ Add                                   ListItemsRole
 AWS::IAM::Role                         N/A
```

```
+ Add                                ListItems
 AWS::Lambda::Function                N/A
+ Add                                UpdateItemApiPUTitemsidPermission
 AWS::Lambda::Permission              N/A
                                     Prod
+ Add                                UpdateItemRole
 AWS::IAM::Role                       N/A
+ Add                                UpdateItem
 AWS::Lambda::Function                N/A
-----------------------------------------------------------------------------------------

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

然後，AWS SAMCLI 部署應用程式：

```
CloudFormation events from stack operations (refresh every 0.5 seconds)
-----------------------------------------------------------------------------------------
ResourceStatus                       ResourceType
 LogicalResourceId                    ResourceStatusReason
-----------------------------------------------------------------------------------------
CREATE_IN_PROGRESS                   AWS::DynamoDB::Table                Items
                        -
CREATE_IN_PROGRESS                   AWS::DynamoDB::Table                Items
                        Resource creation Initiated
CREATE_COMPLETE                      AWS::DynamoDB::Table                Items
                        -
CREATE_IN_PROGRESS                   AWS::IAM::Role
 DeleteItemRole                       -
CREATE_IN_PROGRESS                   AWS::IAM::Role
 ListItemsRole                        -
CREATE_IN_PROGRESS                   AWS::IAM::Role
 UpdateItemRole                       -
CREATE_IN_PROGRESS                   AWS::IAM::Role                      GetItemRole
                        -
CREATE_IN_PROGRESS                   AWS::IAM::Role
 CreateItemRole                       -
CREATE_IN_PROGRESS                   AWS::IAM::Role
 DeleteItemRole                       Resource creation Initiated
CREATE_IN_PROGRESS                   AWS::IAM::Role
 ListItemsRole                        Resource creation Initiated
```

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

```
CREATE_COMPLETE                    AWS::Lambda::Function              GetItem
                        -
CREATE_IN_PROGRESS                 AWS::ApiGateway::RestApi           Api
                        -
CREATE_IN_PROGRESS                 AWS::ApiGateway::RestApi           Api
                       Resource creation Initiated
CREATE_COMPLETE                    AWS::ApiGateway::RestApi           Api
                        -
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 GetItemApiGETitemsidPermissionPro  -
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 ListItemsApiGETitemsPermissionPro  -
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 DeleteItemApiDELETEitemsidPermiss  -
                                                                       ionProd
CREATE_IN_PROGRESS                 AWS::ApiGateway::Deployment
 ApiDeploymentcc153d135b            -
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 UpdateItemApiPUTitemsidPermission  -
                                                                       Prod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 CreateItemApiPOSTitemsPermissionP  -
                                                                       rod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 GetItemApiGETitemsidPermissionPro  Resource creation Initiated
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 UpdateItemApiPUTitemsidPermission  Resource creation Initiated
                                                                       Prod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 CreateItemApiPOSTitemsPermissionP  Resource creation Initiated
                                                                       rod
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 ListItemsApiGETitemsPermissionPro  Resource creation Initiated
                                                                       d
CREATE_IN_PROGRESS                 AWS::Lambda::Permission
 DeleteItemApiDELETEitemsidPermiss  Resource creation Initiated
                                                                       ionProd
CREATE_IN_PROGRESS                 AWS::ApiGateway::Deployment
 ApiDeploymentcc153d135b            Resource creation Initiated
CREATE_COMPLETE                    AWS::ApiGateway::Deployment
 ApiDeploymentcc153d135b            -
```

```
CREATE_IN_PROGRESS                  AWS::ApiGateway::Stage
 ApiProdStage                        -
CREATE_IN_PROGRESS                  AWS::ApiGateway::Stage
 ApiProdStage                        Resource creation Initiated
CREATE_COMPLETE                     AWS::ApiGateway::Stage
 ApiProdStage                        -
CREATE_COMPLETE                     AWS::Lambda::Permission
 CreateItemApiPOSTitemsPermissionP   -
                                                                    rod
CREATE_COMPLETE                     AWS::Lambda::Permission
 UpdateItemApiPUTitemsidPermission   -
                                                                    Prod
CREATE_COMPLETE                     AWS::Lambda::Permission
 ListItemsApiGETitemsPermissionPro   -
                                                                    d
CREATE_COMPLETE                     AWS::Lambda::Permission
 DeleteItemApiDELETEitemsidPermiss   -
                                                                    ionProd
CREATE_COMPLETE                     AWS::Lambda::Permission
 GetItemApiGETitemsidPermissionPro   -
                                                                    d
CREATE_COMPLETE                     AWS::CloudFormation::Stack      aws-app-
composer-basic-api          -
-------------------------------------------------------------------------------------------
```

最後會顯示訊息，通知您部署成功：

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

使用 Infrastructure Composer 搭配 AWS SAM 刪除堆疊

此範例說明如何使用 sam delete命令刪除 AWS CloudFormation 堆疊。

在 sam delete中輸入 命令 AWS SAMCLI，並確認您是否要刪除堆疊和範本：

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

```
 - Deleting S3 object with key 875e4bcf4b10a6a1144ad83158d84b6d
 - Deleting S3 object with key 20b869d98d61746dedd9aa33aa08a6fb
 - Deleting S3 object with key c513cedc4db6bc184ce30e94602741d6
 - Deleting S3 object with key c7a15d7d8d1c24b77a1eddf8caebc665
 - Deleting S3 object with key e8b8984f881c3732bfb34257cdd58f1e
 - Deleting S3 object with key 3185c59b550594ee7fca7f8c36686119.template
 - Deleting S3 object with key 30439348c0be6e1b85043b7a935b34ab.template
 - Deleting Cloudformation stack aws-app-composer-basic-api

Deleted successfully
```

AWS Infrastructure Composer 故障診斷

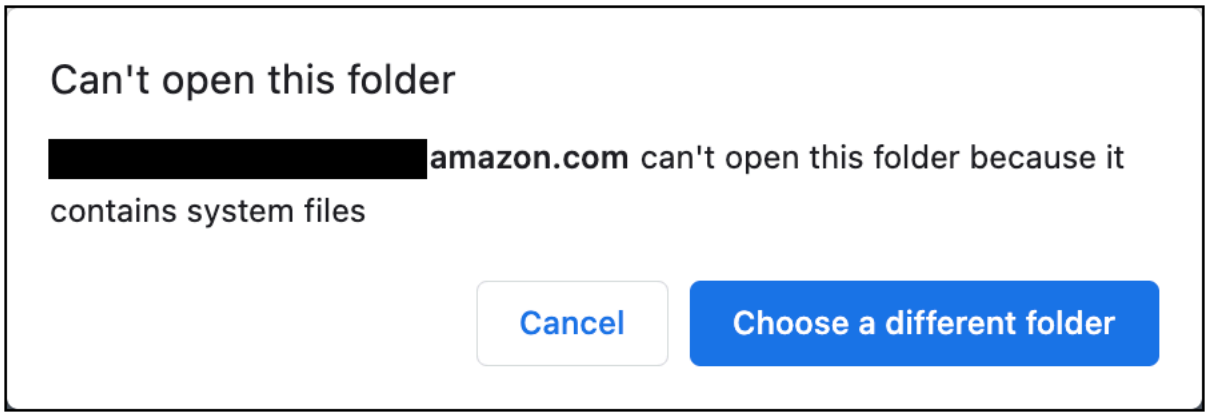
本節中的主題提供使用 時疑難排解錯誤訊息的指引 AWS Infrastructure Composer。

主題

錯誤訊息

「無法開啟此資料夾」

錯誤範例：



可能原因：基礎設施撰寫者無法使用本機同步模式存取敏感目錄。

若要進一步了解此錯誤，請參閱 Data Infrastructure Composer 存取。

嘗試連線至不同的本機目錄，或使用停用本機同步的 Infrastructure Composer。

「不相容的範本」

錯誤範例：在 Infrastructure Composer 中載入新專案時，您會看到以下內容：

可能原因：您的專案包含 Infrastructure Composer 不支援的外部參考檔案。

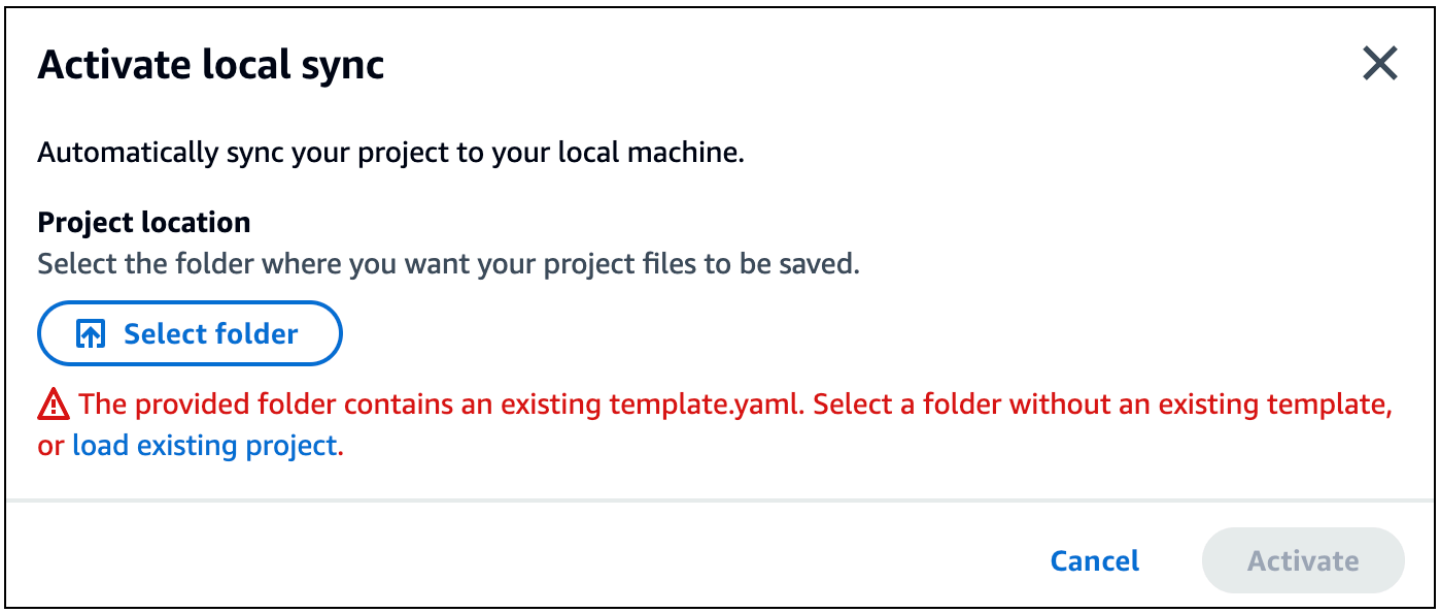
若要了解 Infrastructure Composer 中支援的外部檔案，請參閱 參考外部檔案。

可能原因：您的專案連結至不同本機目錄中的外部檔案。

將外部參考檔案移至您選取要與 Infrastructure Composer 本機同步模式搭配使用的目錄子目錄。

「提供的資料夾包含現有的 template.yaml」

嘗試啟用本機同步時，您會看到下列錯誤：



可能原因：您選取的資料夾已包含 template.yaml 檔案。

選取不包含應用程式範本的另一個目錄，或建立新的目錄。

「您的瀏覽器沒有將專案儲存在該資料夾中的許可...」

可能原因：基礎設施撰寫者無法使用本機同步模式存取敏感目錄。

若要進一步了解此錯誤，請參閱 Data Infrastructure Composer 存取。

嘗試連線至不同的本機目錄，或使用停用本機同步的 Infrastructure Composer。

中的安全性 AWS Infrastructure Composer

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 和 之間的共同責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。第三方稽核人員會定期測試和驗證我們安全的有效性，做為AWS 合規計畫的一部分。若要了解適用於 的合規計劃 AWS Infrastructure Composer，請參閱AWS 合規計劃範圍內的服務。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 Infrastructure Composer 時套用共同責任模型。下列主題說明如何設定 Infrastructure Composer 以符合您的安全和合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 Infrastructure Composer 資源。

主題

- 中的資料保護 AWS Infrastructure Composer
- AWS Identity and Access Management 適用於 AWS Infrastructure Composer
- 的合規驗證 AWS Infrastructure Composer
- 中的彈性 AWS Infrastructure Composer

中的資料保護 AWS Infrastructure Composer

AWS 共同責任模型適用於 AWS Infrastructure Composer 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 Infrastructure Composer 或其他 AWS 服務 使用主控台、API AWS CLI或 AWS SDKs時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

> **Note**
>
> 您輸入至 Infrastructure Composer 的所有資料僅用於在 Infrastructure Composer 內提供功能，以及產生本機儲存到機器的專案檔案和目錄。Infrastructure Composer 不會儲存、存放或傳輸任何這些資料。

資料加密

Infrastructure Composer 不會加密客戶內容，因為資料不會儲存、儲存或傳輸。

靜態加密

Infrastructure Composer 不會加密客戶內容，因為資料不會儲存、儲存或傳輸。

傳輸中加密

Infrastructure Composer 不會加密客戶內容，因為資料不會儲存、儲存或傳輸。

金鑰管理

Infrastructure Composer 不支援金鑰管理，因為客戶內容不會儲存、儲存或傳輸。

網際網路流量隱私權

Infrastructure Composer 不會產生具有內部部署用戶端和應用程式的流量。

AWS Identity and Access Management 適用於 AWS Infrastructure Composer

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證 （ 登入 ） 和授權 （ 具有許可 ），以使用 Infrastructure Composer 資源。IAM 是 AWS 服務 您可以免費使用的 。

主題

目標對象

Infrastructure Composer 至少需要對 的唯讀存取權 AWS Management Console。具有此授權的任何使用者都可以使用 Infrastructure Composer 的所有功能。不支援精細存取 Infrastructure Composer 的特定功能。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （ 登入 AWS)。

您可以使用透過身分來源提供的登入資料，以聯合身分 AWS 身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

根據您身分的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的 憑證以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需

使用建議的方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的適用於 API 請求的AWS Signature 第 4 版。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來提高帳戶的安全性。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的多重要素驗證和《IAM 使用者指南》中的 IAM 中的AWS 多重要素驗證。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取 帳戶中的所有 AWS 服務 和資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳實務是， 要求人類使用者，包括需要管理員存取權的使用者，使用臨時登入資料 AWS 服務 來使用與身分提供者的聯合來存取 。

聯合身分是來自您的企業使用者目錄、Web 身分提供者、 AWS Directory Service、身分中心目錄，或是使用透過身分來源提供的憑證 AWS 服務 存取的任何使用者。當聯合身分存取時 AWS 帳戶，它們會擔任角色，而角色會提供臨時憑證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，或者您可以連接並同步到您自己的身分來源中的一組使用者和群組，以用於所有 AWS 帳戶 和 應用程式。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是 中具有單一人員或應用程式特定許可 AWS 帳戶 的身分。建議您盡可能依賴臨時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色（主控台）。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱《IAM 使用者指南》中的為第三方身分提供者 (聯合) 建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以直接將政策連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在 中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，並結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱「IAM 使用者指南」中的建立角色以委派許可權給 AWS 服務。

- 服務連結角色 – 服務連結角色是連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時憑證，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體，並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件， AWS 當與身分或資源相關聯時， 會定義其許可。當委託人（使用者、根使用者或角色工作階段）發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 的形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console、 AWS CLI或 API AWS 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，請參閱《IAM 使用者指南》中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包括帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 in. 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 的多個的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括支援 RCPs AWS 服務 的 清單，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 RCPs)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作

階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS Infrastructure Composer 如何使用 IAM

AWS Infrastructure Composer 至少需要對 的唯讀存取權 AWS Management Console。具有此授權的任何使用者都可以使用 Infrastructure Composer 的所有功能。不支援精細存取 Infrastructure Composer 的特定功能。

當您將專案範本和檔案部署到 時 AWS CloudFormation，您將需要具備必要的許可。若要進一步了解，請參閱AWS CloudFormation 《 使用者指南》中的使用 控制存取權 AWS Identity and Access Management。

下表顯示可搭配哪些 IAM 功能使用 AWS Infrastructure Composer。

IAM 功能	Infrastructure Composer 支援
身分型政策	否
資源型政策	否
政策動作	否
政策資源	否
政策條件索引鍵	否
ACL	否
ABAC(政策中的標籤)	否
暫時性憑證	是
主體許可	否

IAM 功能	Infrastructure Composer 支援
服務角色	否
服務連結角色	否

若要深入了解 Infrastructure Composer 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱《IAM 使用者指南》中的AWS 與 IAM 搭配使用的 服務。

Infrastructure Composer 的身分型政策

支援以身分為基礎的政策：否

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

Infrastructure Composer 中的資源型政策

支援資源型政策：否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包括帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當委託人和資源位於不同的位置時 AWS 帳戶，信任帳戶中的 IAM 管理員也必須授予委託人實體 （使用者或角色） 存取資源的許可。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

Infrastructure Composer 的政策動作

支援政策動作：否

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 Infrastructure Composer 動作清單，請參閱服務授權參考中的 AWS Infrastructure Composer 定義的動作。

Infrastructure Composer 的政策資源

支援政策資源：否

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。陳述式必須包含 `Resource` 或 `NotResource` 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

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

若要查看 Infrastructure Composer 資源類型及其 ARNs 的清單，請參閱服務授權參考中的 AWS Infrastructure Composer 定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 AWS Infrastructure Composer 定義的動作。

Infrastructure Composer 的政策條件索引鍵

支援服務特定的政策條件索引鍵：否

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值， 會使用邏輯OR操作 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件索引鍵和服務特定條件索引鍵。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 Infrastructure Composer 條件索引鍵的清單，請參閱服務授權參考中的 AWS Infrastructure Composer 的條件索引鍵。若要了解您可以使用條件索引鍵的動作和資源，請參閱 AWS Infrastructure Composer 定義的動作。

Infrastructure Composer ACLs

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 與 Infrastructure Composer

支援 ABAC （政策中的標籤）：否

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在 中 AWS，這些屬性稱為標籤。您可以將標籤連接至 IAM 實體 （使用者或角色） 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 aws:ResourceTag/*key-name*、aws:RequestTag/*key-name* 或 aws:TagKeys 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的使用屬性型存取控制 (ABAC)。

搭配 Infrastructure Composer 使用臨時憑證

支援臨時憑證：是

當您使用臨時登入資料登入時，有些 AWS 服務 無法使用。如需詳細資訊，包括哪些 AWS 服務 使用臨時登入資料，請參閱《AWS 服務 IAM 使用者指南》中的使用 IAM 的 。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則會使用臨時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的從使用者切換至 IAM 角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

您可以使用臨時登入資料，透過 存取 Infrastructure Composer AWS Management Console。如需範例，請參閱《IAM 使用者指南》中的啟用對 AWS 主控台的自訂身分代理程式存取。

Infrastructure Composer 的跨服務主體許可

支援轉寄存取工作階段 (FAS)：否

當您使用 IAM 使用者或角色在 中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，並結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。

Infrastructure Composer 的服務角色

支援服務角色：否

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱「IAM 使用者指南」中的建立角色以委派許可權給 AWS 服務。

> **Warning**
>
> 變更服務角色的許可可能會中斷 Infrastructure Composer 功能。只有在 Infrastructure Composer 提供指引時，才能編輯服務角色。

Infrastructure Composer 的服務連結角色

支援服務連結角色：否

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱可搭配 IAM 運作的AWS 服務。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

的合規驗證 AWS Infrastructure Composer

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱在 中下載報告 AWS Artifact。

使用 時的合規責任 AWS 服務 取決於資料的敏感度、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源以協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- HIPAA 合格服務參考 – 列出 HIPAA 合格服務。並非所有 AWS 服務 都符合 HIPAA 資格。
- AWS 合規資源 – 此工作手冊和指南的集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同責任模型。本指南摘要說明保護 的最佳實務， AWS 服務 並將指南映射到跨多個架構的安全控制 （ 包括國家標準和技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO))。
- AWS Config 開發人員指南中的使用規則評估資源 – AWS Config 服務會評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub – 這 AWS 服務 可讓您全面檢視其中的安全狀態 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱「Security Hub 控制參考」。
- Amazon GuardDuty – 這可透過監控您的環境是否有可疑和惡意活動，來 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。

- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和產業標準的方式。

中的彈性 AWS Infrastructure Composer

AWS 全域基礎設施是以 AWS 區域 和 可用區域為基礎建置。 AWS 區域 提供多個實體隔離和隔離的可用區域，這些區域與低延遲、高輸送量和高度冗餘聯網連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和 可用區域的詳細資訊，請參閱AWS 全球基礎設施。

您輸入至 Infrastructure Composer 的所有資料僅用於在 Infrastructure Composer 內提供功能，以及產生本機儲存到機器的專案檔案和目錄。Infrastructure Composer 不會儲存或存放任何這些資料。

Infrastructure Composer 的文件歷史記錄

下表說明 Infrastructure Composer 的重要文件版本。如需有關此文件更新的通知，您可以訂閱 RSS 摘要。

- 文件最近更新時間：2023 年 11 月 30 日

變更	描述	日期
開發人員指南中的重組和更新內容	重組和重組指南，以改善可探索性和可用性。更新並改善了標題。介紹主題和概念時提供其他詳細資訊。	2024 年 8 月 1 日
新增在 CloudFormation 主控台模式中使用 Infrastructure Composer 的文件，並重新組織 Infrastructure Composer 開發人員指南。	AWS Infrastructure Composer 現在可以在 AWS CloudFormation 主控台模式中使用。若要進一步了解，請參閱在 CloudFormation 主控台模式中使用 Infrastructure Composer。此外，使用者指南中的大部分內容都經過重新組織，以建立簡化的體驗。	2024 年 3 月 28 日
新增了與 CodeWhisperer 整合的 Infrastructure Composer 文件	AWS Infrastructure Composer 從 Toolkit for VS Code 提供與 Amazon CodeWhisperer 的整合。若要進一步了解，請參閱搭配使用 AWS Infrastructure Composer 與 Amazon CodeWhisperer。	2023 年 11 月 30 日
已新增從 使用 Infrastructure Composer 部署應用程式的文件 AWS Toolkit for Visual Studio Code	使用 Infrastructure Composer 畫布的同步按鈕，將您的應用程式部署到 AWS 雲端。若要	2023 年 11 月 30 日

	進一步了解，請參閱使用 sam 同步部署您的應用程式。	
從 新增 Infrastructure Composer 的文件 AWS Toolkit for Visual Studio Code	您現在可以將來自 VS Code 的 Infrastructure Composer 與 搭配使用 AWS Toolkit for Visual Studio Code。若要進一步了解，請參閱AWS Infrastructure Composer 從使用 AWS Toolkit for Visual Studio Code。	2023 年 11 月 30 日
新增了 Step Functions Workflow Studio 整合	從 Infrastructure Composer 畫布啟動 Step Functions Workflow Studio。若要進一步了解，請參閱AWS Infrastructure Composer 搭配 使用 AWS Step Functions。	2023 年 11 月 27 日
新增了 Lambda 主控台和 Infrastructure Composer 整合	從 Lambda 主控台啟動 Infrastructure Composer 畫布。若要進一步了解，請參閱搭配使用 AWS Infrastructure Composer 與 AWS Lambda 主控台。	2023 年 11 月 14 日
新增 Amazon VPC 為具有 Infrastructure Composer 的精選服務	Infrastructure Composer 推出 VPC 標籤，以視覺化使用 VPC 設定的資源。您也可以使用外部範本上定義的 VPCs 來設定 Lambda 函數。若要進一步了解，請參閱搭配 Amazon VPC 使用 Infrastructure Composer。	2023 年 10 月 17 日

新增 Amazon RDS 為具有 Infrastructure Composer 的精選服務	將您的 Infrastructure Composer 應用程式連接至外部範本上定義的 Amazon RDS 資料庫叢集或執行個體。若要進一步了解，請參閱搭配 Amazon RDS 使用 Infrastructure Composer。	2023 年 10 月 17 日
新增了 Infrastructure Composer 支援，以使用所有 AWS CloudFormation 資源進行設計	從 AWS CloudFormation 資源調色盤中選取任何資源以設計您的應用程式。若要進一步了解，請參閱使用任何 AWS CloudFormation 資源。	2023 年 9 月 26 日
新增 Infrastructure Composer 中卡片的文件	Infrastructure Composer 支援多種類型的卡片，可用來設計和建置您的應用程式。若要進一步了解，請參閱 Infrastructure Composer 中的使用卡片設計。	2023 年 9 月 20 日
新增復原和重做功能的文件	使用 Infrastructure Composer 畫布上的復原和重做按鈕。若要進一步了解，請參閱復原和重做。	2023 年 8 月 1 日
新增本機同步模式的文件	使用本機同步模式自動同步專案，並將專案儲存至本機機器。若要進一步了解，請參閱本機同步模式。	2023 年 8 月 1 日
新增了匯出畫布功能的文件	使用匯出畫布功能，將應用程式的畫布匯出為本機機器的影像。若要進一步了解，請參閱匯出畫布。	2023 年 8 月 1 日

Infrastructure Composer 支援外部檔案參考	參考 Infrastructure Composer 中支援的 資源的外部檔案。若要進一步了解，請參閱使用參考外部檔案的範本。	2023 年 5 月 17 日
連線資源的新文件	將資源連接在一起，以定義應用程式中資源之間的事件驅動關係。若要進一步了解，請參閱使用 Infrastructure Composer 視覺化畫布將資源連接在一起。	2023 年 3 月 7 日
新的變更檢查器功能	使用 Change Inspector 來檢視範本程式碼更新，並了解 Infrastructure Composer 為您建立哪些項目。若要進一步了解，請參閱使用 Change Inspector 檢視程式碼更新。	2023 年 3 月 7 日
Infrastructure Composer 現在已全面推出	AWS Infrastructure Composer 現已正式推出。若要進一步了解，請參閱AWS Infrastructure Composer 現已全面推出 - 以視覺化方式快速建置無伺服器應用程式。	2023 年 3 月 7 日
擴展使用連線模式的優點	在連線模式下使用 Infrastructure Composer 搭配本機 IDE 來加速開發。若要進一步了解，請參閱將 Infrastructure Composer 與本機 IDE 搭配使用。	2023 年 3 月 7 日

更新使用其他 AWS 服務來部署應用程式的主題	使用 Infrastructure Composer 設計可立即部署的無伺服器應用程式。使用 AWS SAM 部署無伺服器應用程式。若要進一步了解，請參閱搭配 AWS CloudFormation 和 使用 Infrastructure Composer AWS SAM。	2023 年 3 月 3 日
新增無伺服器概念區段	在使用 Infrastructure Composer 之前，了解基本的無伺服器概念。若要進一步了解，請參閱無伺服器概念。	2023 年 3 月 2 日
公開發行	Infrastructure Composer 的初始公開版本。	2022 年 12 月 1 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。