



使用者指南

# AWS IoT TwinMaker



# AWS IoT TwinMaker: 使用者指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

什麼是 AWS IoT TwinMaker ? .....	1
運作方式 .....	1
關鍵概念和組成部分 .....	2
工作區 .....	3
實體組件模型 .....	3
視覺效果 .....	4
開始使用 AWS IoT TwinMaker .....	6
建立及管理下列項目的服務角色 AWS IoT TwinMaker .....	7
指派信任 .....	7
Amazon S3 許可 .....	7
將許可指派給特定的 Amazon S3 儲存貯體 .....	8
內建連接器的權限 .....	10
外部資料來源之連接器的權限 .....	13
修改您的工作區 IAM 角色以使用 Athena 資料連接器 .....	14
建立工作區 .....	15
建立您的第一個實體 .....	17
設置一個 AWS 帳戶 .....	21
註冊一個 AWS 帳戶 .....	21
建立具有管理權限的使用者 .....	22
使用和建立元件類型 .....	24
內建元件類型 .....	24
AWS IoT TwinMaker 元件类型的核心功能 .....	25
建立屬性定義 .....	26
建立函數 .....	27
範例元件類型 .....	27
警報 (摘要) .....	27
時間流遙測 .....	29
報警 (從抽象報警繼承) .....	30
設備實例 .....	30
批量操作 .....	34
關鍵概念和術語 .....	34
AWS IoT TwinMaker metadataTransferJob 功能 .....	35
執行大量匯入和匯出作業 .....	36
metadataTransferJob 前提 .....	36

IAM 許可 .....	37
執行大量作業 .....	40
錯誤處理 .....	43
匯入詮釋資料樣 .....	44
AWS IoT TwinMaker metadataTransferJob 例子 .....	47
AWS IoT TwinMaker 詮釋資料傳送工作網 .....	49
資料連接器 .....	66
資料連接器 .....	66
結構描述初始設定式 .....	67
DataReaderByEntity .....	68
DataReaderByComponentType .....	69
DataReader .....	70
AttributePropertyValueReaderByEntity .....	71
DataWriter .....	72
範例 .....	73
Athena 表格資料連接 .....	82
AWS IoT TwinMaker Athena 資料連接器的 .....	82
使用 Athena 資料連接器 .....	83
使用 Athena 表格式資料連接器 JSON 參考 .....	87
使用 Athena 資料連接器 .....	88
在 Grafana 中視覺化 Athena 表格式資料 .....	88
AWS IoT TwinMaker 時間序列資料連接器 .....	90
AWS IoT TwinMaker 時間序列資料連接器必要 .....	91
時間序列資料連接器背景 .....	91
開發時間序列資料連接器 .....	92
改善您的資料連接器 .....	101
測試您的連接器 .....	102
安全 .....	102
建立 AWS IoT TwinMaker 資源 .....	102
下一步是什麼 .....	104
AWS IoT TwinMaker 餅乾工廠數據連接器 .....	104
建立 AWS IoT TwinMaker 場景 .....	109
建立場景之前 .....	109
將資源匯入之前，先將資源最佳化 AWS IoT TwinMaker .....	109
效能的最佳做法 AWS IoT TwinMaker .....	110
進一步了解 .....	110

上傳資源 AWS IoT TwinMaker .....	110
使用主控台將檔案上傳至資源庫 .....	110
建立您的場景 .....	111
在場景中使用 3D 導覽 AWS IoT TwinMaker .....	112
新增固定攝影機 .....	113
增強的編輯 .....	114
場景物件的目標放置 .....	114
子模型選擇 .....	115
編輯場景階層中的圖元 .....	115
將註釋新增至實體 .....	116
將覆蓋添加到標籤 .....	121
編輯您的場景 .....	129
新增模型 .....	129
新增小工具 .....	130
新增標籤 .....	132
最佳化您的 3D 模型 .....	133
在場景中使用 3D 瓷磚 .....	133
動態場景 .....	135
靜態與動態場景 .....	135
全景零組件類型和圖元 .....	136
動態場景概念 .....	137
AWS IoT TwinMaker 應用程式包集成 .....	138
切換 AWS IoT TwinMaker 定價模式 .....	139
知識圖 .....	141
AWS IoT TwinMaker 知識圖核心概念 .....	141
使用知識圖 .....	142
產生場景圖 .....	144
AWS IoT TwinMaker 場景圖先決條件 .....	145
在場景中綁定 3D 節點 .....	146
建立網路應用程式 .....	147
知識圖表 Grafana 面板 .....	149
AWS IoT TwinMaker 查詢編輯器必備 .....	149
知識圖表 Grafana 權限 .....	150
知識圖其他資源 .....	154
資產同步處理 AWS IoT SiteWise .....	167
使用資產同步 AWS IoT SiteWise .....	167

使用自訂工作區 .....	167
使用 IoT SiteWiseDefaultWorkspace .....	171
自訂和預設工作區之間的差異 .....	171
資源同步來源 AWS IoT SiteWise .....	172
自訂和預設工作區 .....	172
僅預設工作區 .....	173
資源未同步 .....	174
在中使用已同步的實體和元件類型 AWS IoT TwinMaker .....	174
分析同步狀態和錯誤 .....	175
同步工作狀態 .....	175
刪除同步工作 .....	177
資產同步限制 .....	178
設置 Grafana 儀表板 .....	179
CORS 組態 .....	179
設定您的 Grafana 環境 .....	181
Amazon Managed Grafana .....	181
自我管理 Grafana .....	182
建立儀表板角色 .....	182
建立 IAM 政策 .....	182
從邊緣上傳視頻 .....	186
新增更多權限 .....	186
建立 Grafana 儀表板 IAM 角色 .....	188
建立AWS IoT TwinMaker視訊播放程式政策 .....	188
縮減對資源的存取範圍 .....	190
縮減 GET 權限的範圍 .....	190
範圍關閉AWS IoT SiteWise BatchPutAssetPropertyValue 權限 .....	191
將警報 Connect 到儀表板 Grafana .....	194
AWS IoT SiteWise 警示組態先決條 .....	194
定義 AWS IoT SiteWise 警示元件 IAM 角色 .....	194
透過 AWS IoT TwinMaker API 進行查詢和更新 .....	195
為您的 Grafana 儀表板設定鬧鐘 .....	197
使用 Grafana 儀表板進行警報可視化 .....	199
馬特港集成 .....	201
整合概述 .....	202
物流端整合先決條件 .....	203
事業端口 SDK 憑證 .....	204

將馬特港認證存儲在 AWS Secrets Manager .....	205
場景中的馬特港掃描 AWS IoT TwinMaker .....	208
Grafana 儀表板中的 AWS IoT TwinMaker 馬特波特 .....	213
與應用程式套件 AWS IoT 集成 .....	213
將視訊串流至 AWS IoT TwinMaker .....	214
使用 Kinesis 視訊串流的邊緣連接器來串流影片 AWS IoT TwinMaker .....	214
必要條件 .....	214
建立AWS IoT TwinMaker場景的視訊元件 .....	214
將 Kinesis 視訊串流中的視訊和中繼資料新增至 Grafana 儀表板 .....	215
使用快速AWS IoT TwinMaker連結資源庫 .....	217
記錄和監控 .....	218
使用亞馬遜 CloudWatch 指標監控 .....	218
指標 .....	219
使用 AWS CloudTrail 記錄 API 呼叫 .....	221
AWS IoT TwinMaker中的資訊 CloudTrail .....	221
安全 .....	223
資料保護 .....	223
靜態加密 .....	224
傳輸中加密 .....	224
身分和存取權管理 .....	224
物件 .....	225
使用身分驗證 .....	226
使用政策管理存取權 .....	228
如何與 IAM AWS IoT TwinMaker 搭配使用 .....	230
身分型政策範例 .....	236
故障診斷 .....	239
使用服務連結角色 .....	240
AWS 受管理政策 .....	242
VPC 端點 (AWS PrivateLink) .....	246
AWS IoT TwinMaker VPC 端點的考量 .....	247
為 AWS IoT TwinMaker建立介面 VPC 端點 .....	248
透 AWS IoT TwinMaker 過介面 VPC 端點存取 .....	249
建立 VPC 端端點原則 AWS IoT TwinMaker .....	250
合規驗證 .....	251
恢復能力 .....	252
基礎設施安全性 .....	252

---

端點和配額 .....	254
AWS IoT TwinMaker 端點和配額 .....	254
其他端點資訊 .....	254
文件歷史紀錄 .....	255
.....	cclvi



# 什麼是 AWS IoT TwinMaker ？

AWS IoT TwinMaker 是一項 AWS IoT 服務，您可以用來構建實體和數位系統的運作數位雙胞胎。AWS IoT TwinMaker 使用來自各種真實世界感測器、攝影機和企業應用程式的量測和分析來建立數位視覺化，協助您追蹤實體工廠、建築物或工業工廠。您可以使用這些真實世界的資料來監視作業、診斷和修正錯誤，以及最佳化作業。

數字孿生是系統及其所有物理和數字組件的實時數字表示。它與數據動態更新，以模仿系統的真實結構，狀態和行為。您可以使用它來推動業務成果。

終端使用者使用使用者介面應用程式，與您的數位孿生資料互動。

## 運作方式

要滿足創建數字孿生的最低要求，您必須執行以下操作。

- 在實體位置建立裝置、設備、空間和程序的模型。
- 將這些模型 Connect 到存儲重要上下文信息的數據源，例如傳感器數據攝像機饋送。
- 創建可幫助用戶了解數據和見解的可視化內容，以便更有效地制定業務決策。
- 讓終端使用者可以使用數位雙胞胎，以推動業務成果。

AWS IoT TwinMaker 透過提供下列功能來解決這些挑戰。

- **實體組件系統知識圖：**在知識圖中 AWS IoT TwinMaker 提供用於建模設備，設備，空間和過程的工具。

此知識圖表包含有關系統的中繼資料，並可連線至不同位置的資料。AWS IoT TwinMaker 為儲存在 Kinesis Video Streams 中 AWS IoT SiteWise 的資料提供內建連接器。您也可以為儲存在其他位置的資料建立自訂連接器。

知識圖和連接器共同提供單一介面，用於查詢不同位置的資料。

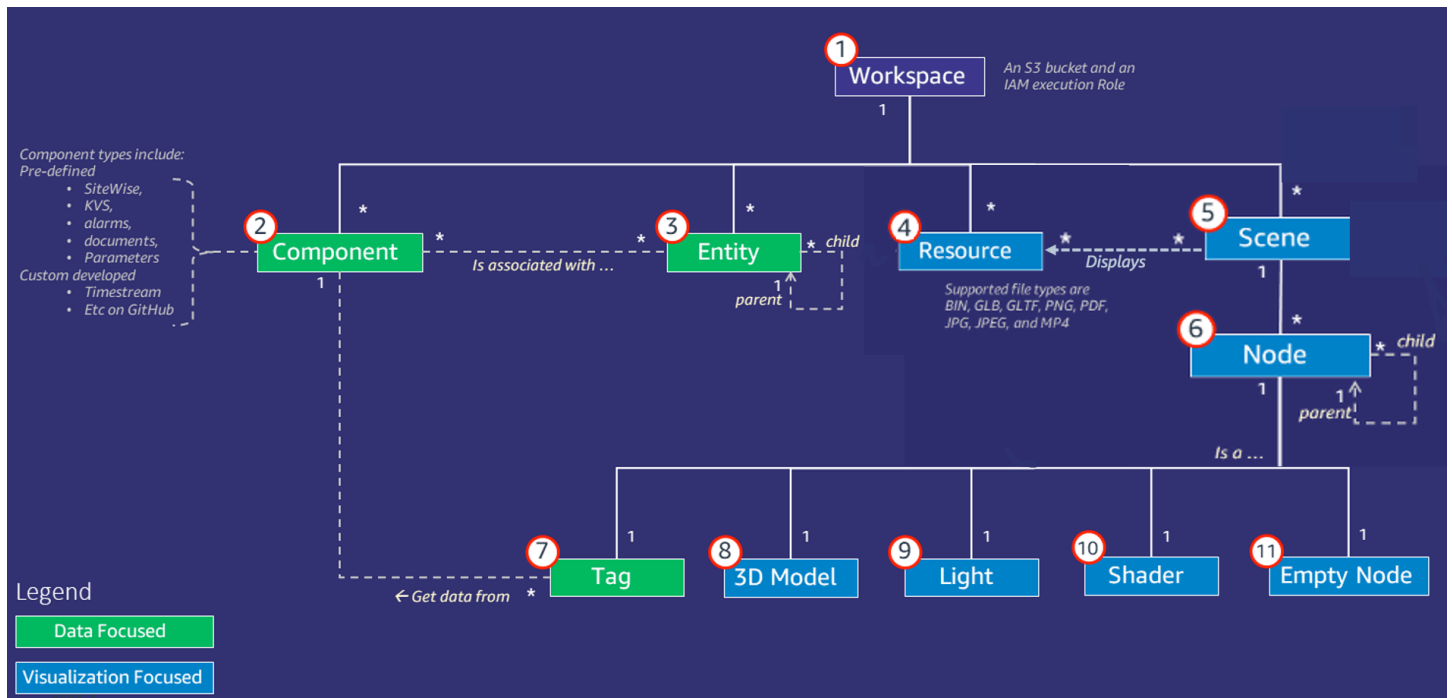
- **場景作曲家：**AWS IoT TwinMaker 控制台提供場景合成工具，用於在 3D 中創建場景。您可以上傳先前建立的 3D/CAD 模型，並針對網頁顯示進行最佳化，並轉換為 .gltf 或 .glb 格式。然後使用場景管理員在單一場景中放置多個模型，建立其作業的視覺表現法。

您還可以在場景中疊加數據。例如，您可以在連接感測器溫度資料的場景位置中建立標籤。這會將資料與位置相關聯。

- 應用程式：為 Grafana 和 Amazon 受管 Grafana AWS IoT TwinMaker 提供外掛程式，您可以使用此外掛程式為最終使用者建立儀表板應用程式。
- 第三方工具：Mendix 與之合作，AWS IoT TwinMaker 為工業 IoT 提供完整的解決方案。請參閱 [Mendix 的研討會精益日常管理應用程式](#)，並 [AWS IoT TwinMaker](#) 開始使用 Mendix 低程式碼應用程式開發平台 (LCAP) 與 Kinesis Video Streams AWS IoT TwinMaker 等 AWS 服務。AWS IoT SiteWise

## 關鍵概念和組成部分

下圖說明的關鍵概念如何 AWS IoT TwinMaker 配合在一起。



### Note

圖表中的星號 (\*) 表示 one-to-many 關係。有關這些關係的配額，請參閱 [AWS IoT TwinMaker 端點和配額](#)。

以下各節說明了圖中所示的概念。

## 工作區

工作區是數位孿生應用程式的頂層容器。您可以在此工作區內為數位對手建立實體、元件、場景資產和其他資源的邏輯集合。它還可以作為安全界限，用於管理對數字孿生應用程序及其包含的資源的訪問。每個工作區都會連結至存放工作區資料的 Amazon S3 儲存貯體。您可以使用 IAM 角色來限制對工作區的存取。

工作區可以包含多個元件、實體、場景和資源。元件類型、實體、場景或資源只存在於一個工作區內。

## 實體組件模型

AWS IoT TwinMaker 提供您使用 entity-component-based 知識圖表建立系統模型的工具。您可以使用實體元件架構來建立實體系統的表示。此實體元件模型由實體、元件和關係組成。如需實體元件系統的詳細資訊，請參閱[實體元件](#)系統。

### 實體

實體是數位雙胞胎中元素的數位表示，可擷取該元素的功能。此元素可以是實體設備、概念或程序。實體具有與其相關聯的元件。這些元件提供相關聯實體的資料和內容。

使用 AWS IoT TwinMaker，您可以將實體組織到自訂階層中，以提高管理效率。實體和元件系統的預設檢視是階層式的。

### 元件

組件為場景中的圖元提供上下文和數據。您可以將零組件加入至圖元。一個組件的生命週期綁定到一個實體的生命週期。

元件可以加入靜態資料，例如文件清單或地理位置的座標。它們還可以具有連接到其他系統的功能，包括包含時間序列數據的系統，例如 AWS IoT SiteWise 和其他時間序列雲歷史學家。

組件由描述數據源和之間的連接的 JSON 文檔定義 AWS IoT TwinMaker。元件可以描述內建的外部資料來源或資料來源 AWS IoT TwinMaker。元件會使用 JSON 文件中指定的 Lambda 函數來存取外部資料來源。一個工作區可以包含許多元件。元件透過關聯實體向標籤提供資料。

AWS IoT TwinMaker 提供數個內建元件，您可以從主控台新增這些元件。您也可以建立自己的自訂元件，以連接至時間流遙測和地理空間座標等資料來源。這些範例包括 TimeStream 遙測、地理空間元件，以及連接至第三方資料來源 (例如 Snowflake) 的連接器。

AWS IoT TwinMaker 針對常見使用案例提供下列類型的內建元件：

- 文件，例如使用者手冊或位於指定 URL 的影像。
- 時間序列，例如來自的傳感器數據 AWS IoT SiteWise。
- 警示，例如來自外部資料來源的時間序列警示。
- 視頻，來自連接到 Kinesis Video Streams 的 IP 攝像機。
- 用於連接至其他資料來源的自訂元件。例如，您可以建立自訂連接器，將 AWS IoT TwinMaker 實體連接至儲存在外部的時間序列資料。

## 資料來源

數據源是數字對手的源數據的位置。AWS IoT TwinMaker 支援兩種類型的資料來源：

- 階層連接器，可讓您持續將外部模型同步至 AWS IoT TwinMaker。
- 時間序列連接器，可讓您連線至時間序列資料庫，例如 AWS IoT SiteWise

## 屬性

屬性是包含在組件中的值，包含靜態和時間序列。當您將元件新增至實體時，元件中的屬性會描述實體目前狀態的詳細資訊。

AWS IoT TwinMaker 支援三種屬性：

- 單一值、non-time-series 屬性 — 這些屬性通常是靜態索引鍵值配對，並直接儲存在 AWS IoT TwinMaker 關聯實體的中繼資料中。
- 時間序列屬性 — AWS IoT TwinMaker 儲存這些屬性之時間序列存放區的參照。這預設為最新值。
- 關係屬性 — 這些屬性會儲存另一個實體或元件的參照。例如，seen\_by 是一個關係元件，可能會將攝影機圖元與該攝影機直接視覺化的另一個圖元相關聯。

您可以使用統一的資料查詢介面，跨異質資料來源查詢屬性值。

## 視覺效果

您可 AWS IoT TwinMaker 以用來增加數位雙胞胎的三維表示法，然後在 Grafana 中檢視它。若要建立場景，請使用現有的 CAD 或其他 3D 檔案類型。然後，您可以使用數據覆蓋為數字雙胞胎添加相關數據。

## 場景

場景是三維表示，可為連接到的資料提供視覺上下文 AWS IoT TwinMaker。場景可以通過在整個環境中使用單個 gltf ( GL 傳輸格式 ) 或 glb 3D 模型創建，也可以通過使用多個模型的組成來創建場景。場景也包含標籤，以表示場景中的興趣點。

場景是視覺效果的頂層容器。場景由一個或多個節點組成。

一個工作區可以包含多個場景。例如，工作區可以針對設施的每個樓層包含一個場景。

## 資源

場景顯示資源，這些資源在 AWS IoT TwinMaker 控制台中顯示為節點。一個場景可以包含許多資源。

資源是用 glTF 於創建場景的圖像和基於三維模型。資源可以代表單一設備或完整網站。

您可以將資源置入場景，方法是將 .gltf 或 .glb 檔案上傳至工作區資源庫，然後將它們新增至場景。

## 增強使用者介面

AWS IoT TwinMaker 您可以使用資料覆疊來增強場景，將重要的上下文和資訊 (例如感應器資料) 新增至場景中的位置。

**節點：**節點是標籤、光源和三維模型的例證。它們也可以是空的，以將結構添加到場景層次結構中。例如，您可以將多個節點群組在單一空白節點下。

**標籤：**標籤是一種節點類型，表示來自組件 ( 通過實體 ) 的數據。一個標籤只能與一個元件相關聯。標籤是加入到場景指定  $x, y, z$  座標位置的註釋。標籤通過使用實體屬性連接這個場景部分到知識圖。您可以使用標籤來設定場景中項目的行為或視覺外觀，例如鬧鐘。

**光源：**您可以在場景中加入光源以使某些物件成為焦點，或在物件上投射陰影以指出其實際位置。

**三維模型：**三維模型是匯入為資源的 .gltf 或 .glb 檔案的視覺化表示法。

### Note

AWS IoT TwinMaker 不適用於任何可能導致嚴重人身傷害或死亡或造成環境或財產損害的危險環境或關鍵系統的操作，或與之相關聯。

通過使用收集的數據 AWS IoT TwinMaker 應根據您的使用案例進行準確評估。AWS IoT TwinMaker 不應用作人工監控實體系統的替代品，以評估這些系統是否安全運作。

# 開始使用 AWS IoT TwinMaker

本節中的主題說明如何執行下列作業。

- 建立並設定新工作區。
- 創建一個實體並添加一個組件到它。

事前準備：

若要建立您的第一個工作區和場景，您需要以下 AWS 資源。

- [AWS 帳戶](#)。
- 的 IAM 服務角色 AWS IoT TwinMaker。當您在[AWS IoT TwinMaker 主控台](#)中建立新 AWS IoT TwinMaker 工作區時，依預設會自動產生此角色。

如果您不選擇允許 AWS IoT TwinMaker 自動建立新的 IAM 服務角色，則必須指定已建立的角色。

如需建立和管理此服務角色的指示，請參閱[???](#)。

如需 IAM 服務角色的詳細資訊，請參閱[建立角色以將許可委派給 AWS 服務](#)。

## Important

此服務角色必須具有附加政策，以授予服務讀取和寫入 Amazon S3 儲存貯體的權限。AWS IoT TwinMaker 使用此角色代表您存取其他服務。您還需要在此角色之間指派信任關係，以 AWS IoT TwinMaker 便服務可以擔任該角色。如果您的雙胞胎與其他 AWS 服務互動，請為這些服務添加必要的權限。

主題

- [建立及管理下列項目的服務角色 AWS IoT TwinMaker](#)
- [建立工作區](#)
- [建立您的第一個實體](#)
- [設置一個 AWS 帳戶](#)

## 建立及管理下列項目的服務角色 AWS IoT TwinMaker

AWS IoT TwinMaker 要求您使用服務角色來允許其代表您存取其他服務中的資源。此角色必須與信任關係 AWS IoT TwinMaker。建立工作區時，必須將此角色指派給工作區。本主題包含範例原則，說明如何設定常見案例的權限。

### 指派信任

下列原則會在您的角色和之間建立信任關係 AWS IoT TwinMaker。將此信任關係指派給您用於工作區的角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iottwinmaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

### Amazon S3 許可

以下政策允許您的角色從 Amazon S3 儲存貯體讀取和刪除和寫入。工作區將資源存放在 Amazon S3 中，因此所有工作區都需要 Amazon S3 許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",

```

```

    "s3:ListBucket",
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "s3:DeleteObject"
  ],
  "Resource": [
    "arn:aws:s3:::*/DO_NOT_DELETE_WORKSPACE_*"
  ]
}
]
}

```

### Note

建立工作區時，請在 Amazon S3 儲存貯體中 AWS IoT TwinMaker 建立一個檔案，指出工作區正在使用該工作區。此原則 AWS IoT TwinMaker 授與刪除工作區時刪除該檔案的權限。AWS IoT TwinMaker 放置與工作區相關的其他物件。您有責任在刪除工作區時刪除這些物件。

## 將許可指派給特定的 Amazon S3 儲存貯體

在 AWS IoT TwinMaker 主控台中建立工作區時，可以選擇為您 AWS IoT TwinMaker 建立 Amazon S3 儲存貯體。您可以使用以下 AWS CLI 命令找到有關此存儲桶的信息。

```
aws iottwinmaker get-workspace --workspace-id workspace name
```

下面的例子顯示了這個命令的輸出的格式。

```
{
```



```

"arn": "arn:aws:iottwinmaker:region:account Id:workspace/workspace name",
"creationDateTime": "2021-11-30T11:30:00.000000-08:00",
"description": "",
"role": "arn:aws:iam::account Id:role/service role name",
"s3Location": "arn:aws:s3::bucket name",
"updateDateTime": "2021-11-30T11:30:00.000000-08:00",
"workspaceId": "workspace name"
}

```

若要更新政策以便為特定 Amazon S3 儲存貯體指派許可，請使用儲存貯體##的值。

以下政策允許您的角色讀取和刪除特定 Amazon S3 儲存貯體，以及寫入特定的 Amazon S3 儲存貯體。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3::bucket name",
        "arn:aws:s3::bucket name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::iottwinmakerbucket/DO_NOT_DELETE_WORKSPACE_*"
      ]
    }
  ]
}

```

## 內建連接器的權限

如果您的工作區使用內建連接器與其他 AWS 服務互動，您必須在此原則中包含這些服務的權限。如果您使用的是 連接器元件類型，您必須包含的權限。AWS IoT SiteWise 如需元件類型的詳細資訊，請參閱[???](#)。

### Note

如果您使用自訂元件類型與其他 AWS 服務互動，則必須授與該角色執行 Lambda 函數的權限，該函數會在您的元件類型中實作該函數。如需詳細資訊，請參閱 [???](#)。

下列範例顯示如何包含 AWS IoT SiteWise 在您的政策中。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name",
        "arn:aws:s3:::bucket name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAsset"
      ],
      "Resource": "asset ARN"
    }
  ]
}
```

```

    "Effect": "Allow",
    "Action": [
      "iotsitewise:DescribeAssetModel"
    ],
    "Resource": "asset model ARN"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:DeleteObject"
    ],
    "Resource": [
      "arn:aws:s3::*/DO_NOT_DELETE_WORKSPACE_*"
    ]
  }
]
}

```

如果您使用的是 `.amazon.iotsitewise.Connect` 元件類型 AWS IoT SiteWise，而且需要從中讀取屬性資料，您必須在原則中包含下列權限。

```

...
{
  "Action": [
    "iotsitewise:GetPropertyValueHistory",
  ],
  "Resource": [
    "AWS IoT SiteWise asset resource ARN"
  ],
  "Effect": "Allow"
},
...

```

如果您使用的是 `.amazon.iotsitewise.Connect` 元件類型 AWS IoT SiteWise，而且需要將屬性資料寫入，您必須在原則中包含下列權限。

```

...
{

```

```

    "Action": [
      "iotsitewise:BatchPutPropertyValues",
    ],
    "Resource": [
      "AWS IoT SiteWise asset resource ARN"
    ],
    "Effect": "Allow"
  },
  ...

```

如果您使用的是網路影片元件類型，您必須包含和 Kinesis Video Streams 的權限。AWS IoT SiteWise 下列範例政策說明如何在您的原則中包含 AWS IoT SiteWise 和 Kinesis Video Streams 權限。

```

...
{
  "Action": [
    "iotsitewise:DescribeAsset",
    "iotsitewise:GetAssetPropertyValue"
  ],
  "Resource": [
    "AWS IoT SiteWise asset resource ARN for the Edge Connector for Kinesis Video Streams"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "iotsitewise:DescribeAssetModel"
  ],
  "Resource": [
    "AWS IoT SiteWise model resource ARN for the Edge Connector for Kinesis Video Streams"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "kinesisvideo:DescribeStream"
  ],
  "Resource": [

```

```

    "Kinesis Video Streams stream ARN"
  ],
  "Effect": "Allow"
},
...

```

## 外部資料來源之連接器的權限

如果您建立的元件類型使用連線至外部資料來源的函數，則必須授與服務角色使用實作函數的 Lambda 函數的權限。如需建立元件類型和函數的詳細資訊，請參閱 [< ???>](#)。

下列範例授予您的服務角色使用 Lambda 函數的權限。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucket*",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name",
        "arn:aws:s3:::bucket name/*"
      ]
    },
    {
      "Action": [
        "lambda:invokeFunction"
      ],
      "Resource": [
        "Lambda function ARN"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": [

```

```

    "s3:DeleteObject"
  ],
  "Resource": [
    "arn:aws:s3::*/DO_NOT_DELETE_WORKSPACE_*"
  ]
}
]
}

```

如需使用 IAM 主控台、和 IAM API 建立角色以及為其指派政策和信任關係的 AWS CLI 詳細資訊，請參閱 [建立角色以將許可委派給 AWS 服務](#)。

## 修改您的工作區 IAM 角色以使用 Athena 資料連接器

若要使用 A [AWS IoT TwinMaker thena 表格式資料連接器](#)，您必須更新 AWS IoT TwinMaker 工作區 IAM 角色。將下列許可新增至您的工作區 IAM 角色：

### Note

此 IAM 變更僅適用於與 AWS Glue 和 Amazon S3 一起存放的 Athena 表格式資料。若要搭配其他資料來源使用 Athena，您必須設定 Athena 的 IAM 角色，請參閱 [Athena 的身分識別與存取管理](#)。

```

{
  "Effect": "Allow",
  "Action": [
    "athena:GetQueryExecution",
    "athena:GetQueryResults",
    "athena:GetTableMetadata",
    "athena:GetWorkGroup",
    "athena:StartQueryExecution",
    "athena:StopQueryExecution"
  ],
  "Resource": [
    "athena resources arn"
  ]
}, // Athena permission
{
  "Effect": "Allow",

```

```
"Action": [
  "glue:GetTable",
  "glue:GetTables",
  "glue:GetDatabase",
  "glue:GetDatabases"
],
"Resource": [
  "glue resources arn"
]
}, // This is an example for accessing aws glue
{
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:GetObject"
  ],
  "Resource": [
    "Amazon S3 data source bucket resources arn"
  ]
}, // S3 bucket for storing the tabular data.
{
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketLocation",
    "s3:GetObject",
    "s3:ListBucket",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts",
    "s3:AbortMultipartUpload",
    "s3:CreateBucket",
    "s3:PutObject",
    "s3:PutBucketPublicAccessBlock"
  ],
  "Resource": [
    "S3 query result bucket resources arn"
  ]
} // Storing the query results
```

如需有關 [Athena IAM 組態的詳細資訊](#)，請閱讀 [Athena 中的身分與存取管理](#)。

## 建立工作區

若要建立和配置您的第一個工作區，請使用下列步驟。

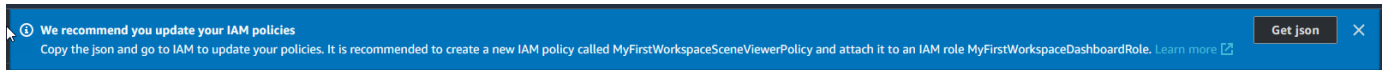
### Note

本主題說明如何使用單一資源建立簡單的工作區。[對於具有多個資源的功能齊全的工作區，請嘗試 Github 存儲庫示例中的示例設置。AWS IoT TwinMaker](#)

1. 在[AWS IoT TwinMaker 主控台](#)首頁上，選擇左側導覽窗格中的「工作區」。
2. 在「工作區」頁面上，選擇建立工作區。
3. 在「建立工作區」頁面上，輸入工作區的名稱。
4. (選擇性) 新增工作區的說明。
5. 在 S3 資源下，選擇建立 S3 儲存貯體。此選項會建立 Amazon S3 儲存貯體，其中 AWS IoT TwinMaker 存放與工作區相關的資訊和資源。每個工作區都需要自己的存儲桶。
6. 在「執行角色」下，選擇「自動產生新角色」或為此工作區建立的自訂 IAM 角色。

如果您選擇自動產生新角色，請將政策 AWS IoT TwinMaker 附加到角色，該角色授與新服務角色存取其他服 AWS 務的權限，包括讀取和寫入您在上一步中指定之 Amazon S3 儲存貯體的權限。如需有關將權限指派給此角色的資訊，請參閱[???](#)。

7. 選擇建立工作區。下列橫幅會顯示在「工作區」頁面的頂端。



8. 選擇「取得 JSON」。建議您將看到的 IAM 政策新增至針對檢視 Grafana 儀表板的使用者和帳戶 AWS IoT TwinMaker 建立的 IAM 角色。此角色的名稱遵循以下模式：`#####DashboardRole`，如需如何建立原則並將其附加至角色的指示，請參閱[修改角色權限原則 \(主控台\)](#)。

下列範例包含要新增至儀表板角色的原則。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::iottwinmaker-workspace-workspace-name-lower-case-account-id",
```



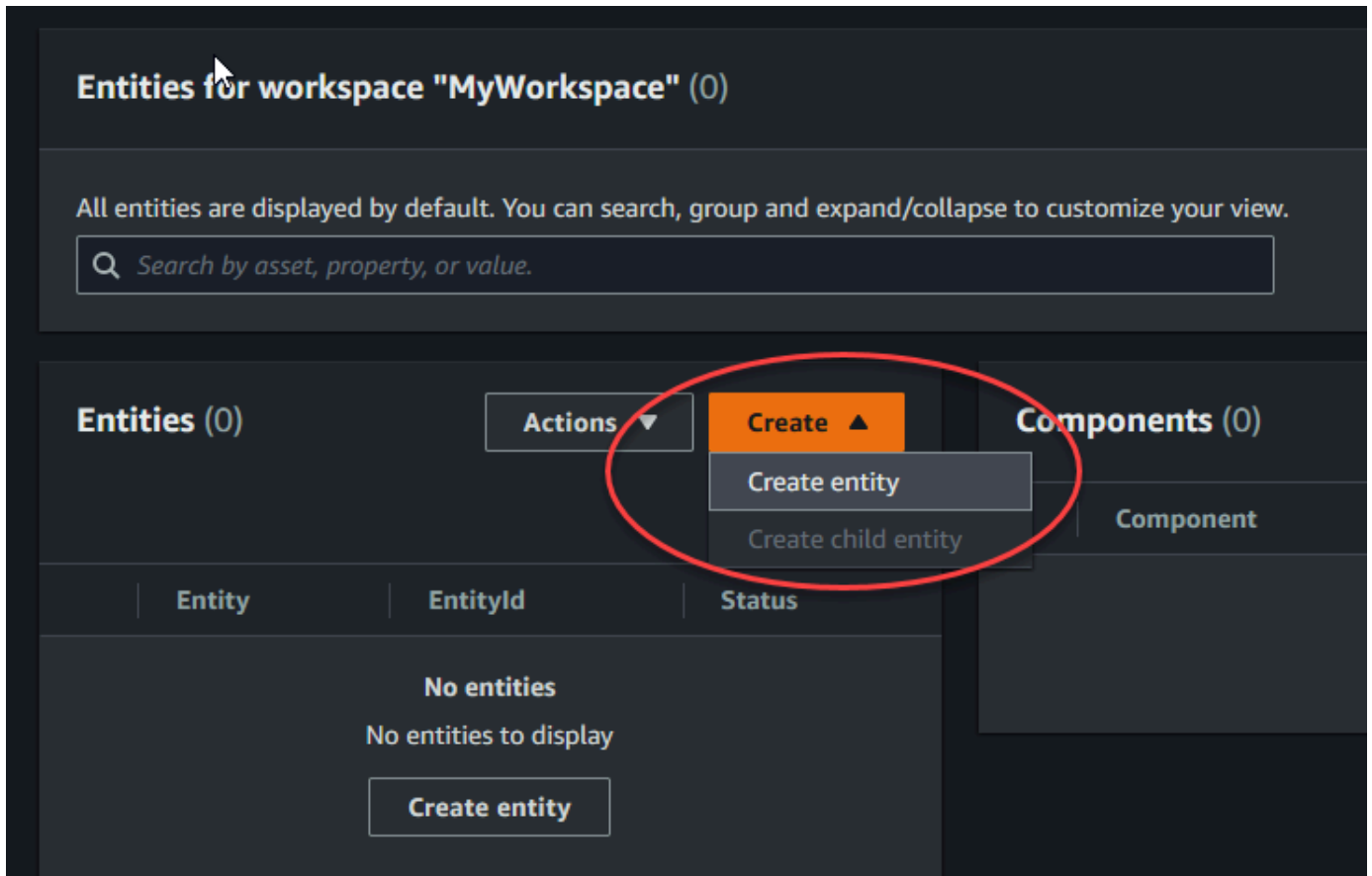
```
    "arn:aws:s3:::iottwinmaker-workspace-workspace-name-lower-case-account-id/
*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:Get*",
    "iottwinmaker:List*"
  ],
  "Resource": [
    "arn:aws:iottwinmaker:us-east-1:account-id:workspace/workspace-name",
    "arn:aws:iottwinmaker:us-east-1:account-id:workspace/workspace-name/*"
  ]
},
{
  "Effect": "Allow",
  "Action": "iottwinmaker:ListWorkspaces",
  "Resource": "*"
}
]
```

您現在可以開始使用第一個實體為工作區建立資料模型。如需如何執行此動作的詳細資訊，請參閱[建立您的第一個實體](#)。

## 建立您的第一個實體

若要建立您的第一個實體，請使用下列步驟。

1. 在 [工作區] 頁面上，選擇您的工作區，然後在左窗格中選擇 [實體]。
2. 在 [實體] 頁面上，選擇 [建立]，然後選擇 [建立實體]。



3. 在「建立實體」視窗中，輸入實體的名稱。此範例使用**CookieMixer**實體。
4. (選擇性) 輸入實體的說明。
5. 選擇建立實體

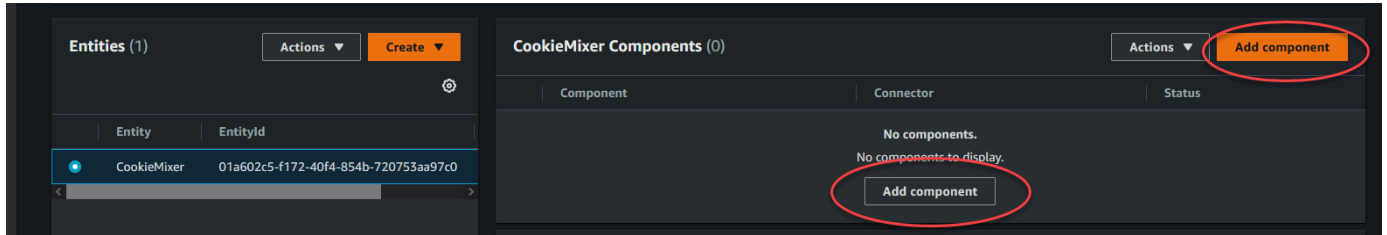
實體包含工作區中每個項目的相關資料。您可以透過加入元件將資料放入實體中。AWS IoT TwinMaker 提供下列內建元件類型。

- 參數：新增一組鍵值屬性。
- 文件：為包含實體相關資訊的文件新增名稱和 URL。
- 警報：連線至警示時間序列資料來源。
- SiteWise 連接器：提取 AWS IoT SiteWise 資產中定義的時間序列性質。
- Kinesis Video Streams 的邊緣連接器 AWS IoT Greengrass：從 K AWS IoT Greengrass VS 的邊緣連接器提取視訊資料。如需詳細資訊，請參閱 [AWS IoT TwinMaker 視訊整合](#)。

您可以在左窗格中選擇元件類型，來查看這些元件類型及其定義。您也可以在此 [元件類型] 頁面上建立新的元件類型。如需建立元件類型的詳細資訊，請參閱 [使用和建立元件類型](#)。

在此範例中，我們建立了一個簡單的文件元件，以新增實體的描述性資訊。

1. 在 [實體] 頁面上，選擇實體，然後選擇 [新增元件]。



2. 在「新增元件」視窗中，輸入元件的名稱。由於此範例使用 Cookie 混合器實體，因此我們 **MixerDescription** 在「名稱」欄位中輸入。

**Add component** ✕

Name

MixerDescription

Type

Types of components include documents, time-series data, structured data, and unstructured data.

com.amazon.iottwinmaker.documents ▼

Edit form  Edit JSON

Document editor

No docs associated to the entity

**Add a doc**

▼ Properties

Property	Data type	is Timeseries	Storage
documents	Map ▼	False ▼	Internal ▼

Value

\_\_\_\_\_

Add another property

Cancel **Add component**

3. 選擇「新增文件」，然後輸入文件名稱和外部 URL 的值。透過文件元件，您可以儲存包含實體重要資訊的外部 URL 清單。
4. 選擇 [新增元件]。

您現在已準備好建立您的第一個場景。如需如何執行此動作的詳細資訊，請參閱[建立和編輯 AWS IoT TwinMaker 場景](#)。

## 設置一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

## 註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 <https://aws.amazon.com/> 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

## 建立具有管理權限的使用者

註冊後，請保護 AWS 帳戶 AWS 帳戶根使用者、啟用和建立系統管理使用者 AWS IAM Identity Center，這樣您就不會將 root 使用者用於日常工作。

### 保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。[AWS Management Console](#)在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的[以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的[為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

### 建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM 身分中心中，將管理存取權授予使用者。

[若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。](#)

### 以具有管理權限的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

如需使用 IAM 身分中心使用者[登入的說明，請參閱使用AWS 登入 者指南中的登入 AWS 存取入口網站](#)。

### 指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[建立權限集](#)」。

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[新增群組](#)」。

## 使用和建立元件類型

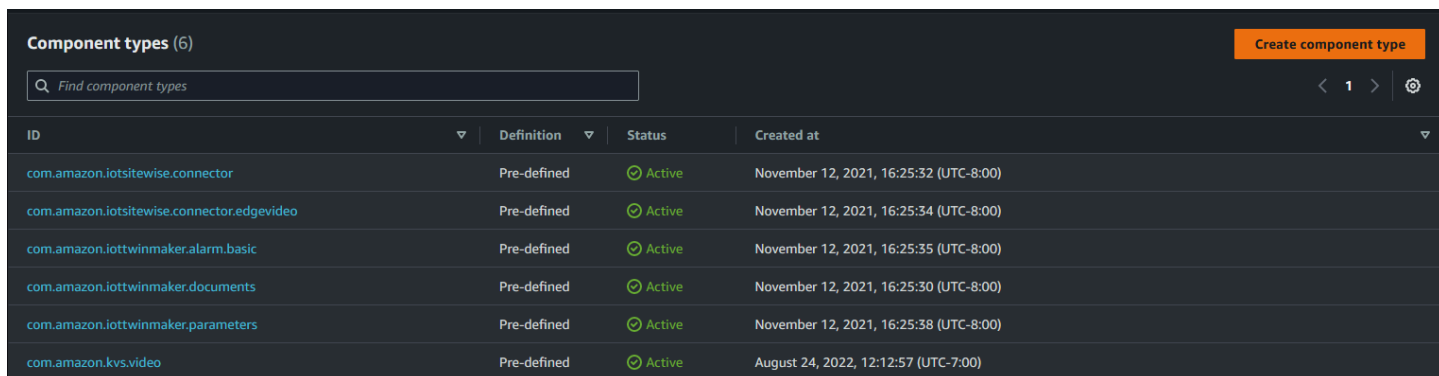
本主題將逐步引導您瞭解用來建立 AWS IoT TwinMaker 元件類型的值 and 結構。它會示範如何建立要求物件，您可以傳遞至 [CreateComponentType](#) API 或使用 AWS IoT TwinMaker 主控台中的元件類型編輯器。

組件為其關聯實體提供屬性和數據的上下文。

### 內建元件類型

在 AWS IoT TwinMaker 主控台中，當您選擇工作區，然後在左窗格中選擇 [元件類型] 時，您會看到下列元件類型。

- 自動同步 AWS IoT SiteWise 資產和資產模型，並將其轉換為實體元件和元件類型的元件類型的元件類型。AWS IoT TwinMaker 如需有關使用 AWS IoT SiteWise 資產同步的詳細資訊，請參閱[與資產同步 AWS IoT SiteWise](#)。
- 基本：一個基本的警報組件，可將警報數據從外部源提取到實體。此組件不包含連接到特定數據源的函數。這意味著警報組件是抽象的，並且可以由另一個組件類型繼承，該組件類型指定了數據源和從該源讀取的函數。
- 文件：包含實體相關資訊之文件的標題至 URL 的簡單對應。
- 連接器. 邊緣視訊：使用適用於運動視訊串流的邊緣連接器，從 IoT 裝置提取視訊至實體的元件。[AWS IoT Greengrass Kinesis Video Streams AWS IoT Greengrass 元件的邊緣連接器](#)不是 AWS IoT TwinMaker 元件，而是部署在 IoT 裝置本機上的預先建置 AWS IoT Greengrass 元件。
- 連接器：將數據提取到實體中的組件。AWS IoT SiteWise
- 參數：將靜態鍵值對添加到實體的組件。
- 視頻：從 Kinesis Video Streams 提取視頻到一個實體的組件。AWS IoT TwinMaker



The screenshot shows the 'Component types' section in the AWS IoT TwinMaker console. It features a search bar, a 'Create component type' button, and a table listing six pre-defined component types, all of which are active.

ID	Definition	Status	Created at
com.amazon.iotsitewise.connector	Pre-defined	Active	November 12, 2021, 16:25:32 (UTC-8:00)
com.amazon.iotsitewise.connector.edgevideo	Pre-defined	Active	November 12, 2021, 16:25:34 (UTC-8:00)
com.amazon.iottwinmaker.alarm.basic	Pre-defined	Active	November 12, 2021, 16:25:35 (UTC-8:00)
com.amazon.iottwinmaker.documents	Pre-defined	Active	November 12, 2021, 16:25:30 (UTC-8:00)
com.amazon.iottwinmaker.parameters	Pre-defined	Active	November 12, 2021, 16:25:38 (UTC-8:00)
com.amazon.kvs.video	Pre-defined	Active	August 24, 2022, 12:12:57 (UTC-7:00)



# AWS IoT TwinMaker 元件類型的核心功能

下列清單說明元件類型的核心功能。

- 屬性定義：[PropertyDefinitionRequest](#)物件定義了一個屬性，您可以在場景撰寫器中填入該屬性，也可以使用從外部資料來源提取的資料填入該屬性。您設定的靜態屬性會儲存在中 AWS IoT TwinMaker。從資料來源提取的時間序列屬性和其他屬性會儲存在外部。

您可以在對PropertyDefinitionRequest映的字串內指定性質定義。每個字串對於地圖必須是唯一的。

- 函數：[FunctionRequest](#)物件指定 Lambda 函數，該函數可從外部資料來源讀取和寫入外部資料來源。

包含具有儲存在外部但沒有對應函數來擷取值的屬性之屬性的組件型別是抽象組件型別。您可以從抽象組件類型擴展具體組件類型。您無法將抽象組件類型添加到實體。它們不會出現在場景作曲家中。

您可以在要對FunctionRequest映的字串內指定函數。字串必須指定下列其中一種預先定義的函數類型。

- dataReader：從外部源提取數據的函數。
- dataReaderByEntity：從外部源提取數據的函數。

當您使用這種類型的資料讀取器時，[GetPropertyValueHistory](#)API 作業僅支援此元件類型中屬性的實體特定查詢。（您只能請求 componentName + 的屬性值歷史記錄entityId。）

- dataReaderByComponentType：從外部源提取數據的函數。

當您使用這種類型的資料讀取器時，[GetPropertyValueHistory](#)API 作業僅支援此元件類型中屬性的跨實體查詢。（您只能要求的屬性值歷史記錄componentTypeId。）

- dataWriter：將資料寫入外部來源的函數。
- schemaInitializer：每當您建立包含元件類型的實體時，會自動初始化屬性值的函數。

在非抽象組件類型中，需要三種類型的數據讀取器函數之一。

如需實作時間串流遙測元件（包括警示）的 Lambda 函數範例，請參閱[AWS IoT TwinMaker 範例](#)中的資料讀取器。

**Note**

由於警示連接器繼承自抽象警示元件類型，因此 Lambda 函數必須傳回alarm\_key值。如果您沒有傳回此值，Grafana 將無法將其識別為鬧鐘。這是所有返回警報的組件都必需的。

- 繼承：組件類型通過繼承促進代碼可重用性。一個元件類型最多可以繼承 10 個父元件類型。

使用extendsFrom參數可指定元件類型從中繼承性質和函數的元件類型。

- ISSingleton：某些元件包含的屬性，例如位置座標，這些屬性不能包含在實體中多次。將isSingleton參數值設定為，true以指示元件類型只能包含在圖元中一次。

## 建立屬性定義

下表說明 a 的參數PropertyDefinitionRequest。

參數	描述
isExternalId	布林值；指定屬性是否為儲存在外部之屬性值的唯一識別碼 (例如 AWS IoT SiteWise 資產 ID)。  此屬性的預設值為 false。
isStoredExternally	布林值；指定屬性值是否儲存在外部。  此屬性的預設值為 false。
isTimeSeries	布林值；指定屬性是否儲存時間序列資料。  此屬性的預設值為 false
isRequiredInEntity	布林值；指定屬性在使用元件類型的實體中是否必須具有值。
dataType	指定屬性之資料類型 (例如字串、 <a href="#">DataType</a> 對映、清單和測量單位) 的物件。
defaultValue	指 <a href="#">DataValue</a> 定屬性預設值的物件。

參數	描述
configuration	一種 string-to-string 地圖，指定您需要連接到外部資料來源的其他資訊。

## 建立函數

下表說明 a 的參數FunctionRequest。

參數	描述
implementedBy	指 <a href="#">DataConnector</a> 定連線至外部資料來源之 Lambda 函數的物件。
requiredProperties	函數為了讀取和寫入外部資料來源所需的屬性清單。
scope	函數的範圍。用Workspace 於範圍涵蓋整個工作區的函數。用Entity於範圍限制為包含元件之實體的函數。

如需展示如何建立和延伸元件類型的範例，請參閱[???](#)。

## 範例元件類型

本主題包含示範如何實作元件類型重要概念的範例。

### 警報 (摘要)

下列範例是顯示在 AWS IoT TwinMaker 主控台內的抽象警示元件類型。它包含functions一個包含沒dataReader有implementedBy值的列表。

```
{
  "componentTypeId": "com.example.alarm.basic:1",
  "workspaceId": "MyWorkspace",
```

```
"description": "Abstract alarm component type",
"functions": {
  "dataReader": {
    "isInherited": false
  }
},
"isSingleton": false,
"propertyDefinitions": {
  "alarm_key": {
    "dataType": { "type": "STRING" },
    "isExternalId": true,
    "isRequiredInEntity": true,
    "isStoredExternally": false,
    "isTimeSeries": false
  },
  "alarm_status": {
    "dataType": {
      "allowedValues": [
        {
          "stringValue": "ACTIVE"
        },
        {
          "stringValue": "SNOOZE_DISABLED"
        },
        {
          "stringValue": "ACKNOWLEDGED"
        },
        {
          "stringValue": "NORMAL"
        }
      ],
      "type": "STRING"
    },
    "isRequiredInEntity": false,
    "isStoredExternally": true,
    "isTimeSeries": true
  }
}
}
```

備註：

componentTypeId和的值workspaceID是必要的。的值對您的工作區componentTypeId必須是唯一的。的值alarm\_key是唯一識別碼，函數可用來從外部來源擷取警示資料。密鑰的值是必需的，並存儲在中 AWS IoT TwinMaker。時alarm\_status間序列值儲存在外部來源中。

[範例中AWS IoT TwinMaker 提供更多範例。](#)

## 時間流遙測

下列範例是簡單的元件類型，可從外部來源擷取特定元件類型 (例如警示或 Cookie 混合器) 的遙測資料。它指定了組件類型繼承的 Lambda 函數。

```
{
  "componentTypeId": "com.example.timestream-telemetry",
  "workspaceId": "MyWorkspace",
  "functions": {
    "dataReader": {
      "implementedBy": {
        "lambda": {
          "arn": "LambdaArn"
        }
      }
    }
  },
  "propertyDefinitions": {
    "telemetryType": {
      "dataType": { "type": "STRING" },
      "isExternalId": false,
      "isStoredExternally": false,
      "isTimeSeries": false,
      "isRequiredInEntity": true
    },
    "telemetryId": {
      "dataType": { "type": "STRING" },
      "isExternalId": false,
      "isStoredExternally": false,
      "isTimeSeries": false,
      "isRequiredInEntity": true
    }
  }
}
```

## 報警 ( 從抽象報警繼承 )

下列範例會繼承抽象警示和時間串流遙測元件類型。它指定了自己的 Lambda 函數來檢索警報數據。

```
{
  "componentTypeId": "com.example.cookiefactory.alarm",
  "workspaceId": "MyWorkspace",
  "extendsFrom": [
    "com.example.timestream-telemetry",
    "com.amazon.iottwinmaker.alarm.basic"
  ],
  "propertyDefinitions": {
    "telemetryType": {
      "defaultValue": {
        "stringValue": "Alarm"
      }
    }
  },
  "functions": {
    "dataReader": {
      "implementedBy": {
        "lambda": {
          "arn": "lambdaArn"
        }
      }
    }
  }
}
```

### Note

由於警示連接器繼承自抽象警示元件類型，因此 Lambda 函數必須傳回 alarm\_key 值。如果您沒有傳回此值，Grafana 將無法將其識別為鬧鐘。這是所有返回警報的組件都必需的。

## 設備實例

本節中的範例說明如何建立潛在設備的模型。您可以使用這些範例來獲得有關如何在自己的流程中建模設備的一些想法。

## 餅乾攪拌機

下列範例會繼承自時間串流遙測元件類型。它為 Cookie 混合器的旋轉速率和溫度指定其他時間序列屬性。

```
{
  "componentTypeId": "com.example.cookiefactory.mixer",
  "workspaceId": "MyWorkspace",
  "extendsFrom": [
    "com.example.timestream-telemetry"
  ],
  "propertyDefinitions": {
    "telemetryType": {
      "defaultValue": { "stringValue": "Mixer" }
    },
    "RPM": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isStoredExternally": true
    },
    "Temperature": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isStoredExternally": true
    }
  }
}
```

## 水箱

下列範例會繼承自時間串流遙測元件類型。它為水箱的體積和流量指定其他時間序列屬性。

```
{
  "componentTypeId": "com.example.cookiefactory.watertank",
  "workspaceId": "MyWorkspace",
  "extendsFrom": [
    "com.example.timestream-telemetry"
  ],
  "propertyDefinitions": {
    "telemetryType": {
```

```

    "defaultValue" : { "stringValue": "WaterTank" }
  },
  "tankVolume1": {
    "dataType": { "type": "DOUBLE" },
    "isTimeSeries": true,
    "isStoredExternally": true
  },
  "tankVolume2": {
    "dataType": { "type": "DOUBLE" },
    "isTimeSeries": true,
    "isStoredExternally": true
  },
  "flowRate1": {
    "dataType": { "type": "DOUBLE" },
    "isTimeSeries": true,
    "isStoredExternally": true
  },
  "flowrate2": {
    "dataType": { "type": "DOUBLE" },
    "isTimeSeries": true,
    "isStoredExternally": true
  }
}
}

```

## 空間位置

下列範例包含屬性，其值會儲存在中 AWS IoT TwinMaker。由於這些值是由使用者指定並儲存在內部，因此不需要任何函數即可擷取它們。此範例也會使用資 RELATIONSHIP 料類型來指定與其他元件類型的關係。

此元件提供輕量型機制，用於將上下文加入至數位孿生。您可以使用它來添加元數據，指示某些東西的位置。您也可以邏輯上使用此資訊，以判斷哪些攝影機可以看到某個設備或空間，或知道如何將某人派往某個位置。

```

{
  "componentTypeId": "com.example.cookiefactory.space",
  "workspaceId": "MyWorkspace",
  "propertyDefinitions": {
    "position": { "dataType": { "nestedType": { "type": "DOUBLE"}, "type": "LIST"}},
    "rotation": { "dataType": { "nestedType": { "type": "DOUBLE"}, "type": "LIST"}},
  }
}

```



```
    "bounds": {"dataType": {"nestedType": {"type": "DOUBLE"}, "type": "LIST"}},  
    "parent_space" : { "dataType": {"type": "RELATIONSHIP"}}  
  }  
}
```

# AWS IoT TwinMaker 批次作業

使用 a metadataTransferJob 大規模傳輸和管理您的 AWS IoT TwinMaker 資源。A metadataTransferJob 可讓您在和 Amazon S3 之間執行批量操作 AWS IoT TwinMaker AWS IoT SiteWise 和傳輸資源。

您可以在下列案例中使用批次處理作業：

- 在帳戶之間大量遷移資產和資料，例如從開發帳戶遷移到生產帳戶。
- 大規模資產管理，例如大規模上傳和編輯 AWS IoT 資產。
- 將資產大量匯入 AWS IoT TwinMaker 和 AWS IoT SiteWise。
- 從現有本 AWS IoT TwinMaker 體檔案 (例如revit或BIM檔案) 大量匯入實體。

## 主題

- [關鍵概念和術語](#)
- [執行大量匯入和匯出作業](#)
- [AWS IoT TwinMaker 詮釋資料傳送工作網](#)

## 關鍵概念和術語

AWS IoT TwinMaker 批次作業使用下列概念和術語：

- 匯入：將資源移至工作區的動 AWS IoT TwinMaker 作。例如，從本機檔案、Amazon S3 儲存貯體中的檔案，或從 AWS IoT SiteWise AWS IoT TwinMaker 工作區。
- 匯出：將資源從工作 AWS IoT TwinMaker 區移至本機機器或 Amazon S3 儲存貯體的動作。
- 來源：您要移動資源的起始位置。

例如，Amazon S3 儲存貯體是匯入來源，而 AWS IoT TwinMaker 工作區則是匯出來源。

- 目的地：您要將資源移至的所需位置。

例如，Amazon S3 儲存貯體是匯出目的地，而 AWS IoT TwinMaker 工作區則是匯入目的地。

- AWS IoT SiteWise 綱要：用來匯入和匯出資源的綱要 AWS IoT SiteWise。
- AWS IoT TwinMaker 綱要：用來匯入和匯出資源的綱要 AWS IoT TwinMaker。
- AWS IoT TwinMaker 頂級資源：現有 API 中使用的資源。具體而言，實體或 ComponentType。

- AWS IoT TwinMaker 子層級資源：中繼資料定義中使用的巢狀資源類型。特別是一個組件。
- 元數據：成功導入或導出所需的關鍵信息以 AWS IoT SiteWise 及 AWS IoT TwinMaker 資源。
- metadataTransferJob：執行時建立的物件CreateMetadataTransferJob。

## AWS IoT TwinMaker metadataTransferJob 功能

本主題說明執行 AWS IoT TwinMaker 批次處理作業時的行為 — 如何 metadataTransferJob 處理 a。它也說明如何使用傳輸資源所需的中繼資料來定義結構描述。AWS IoT TwinMaker 批次作業支援下列功能：

- 頂層資源建立或取代：AWS IoT TwinMaker 將建立新資源或取代以資源 ID 唯一識別的所有現有資源。

例如，如果系統中存在實體，則實體定義將被在該Entity金鑰下的範本中定義的新實體定義所取代。

- 子資源創建或替換：

從 EntityComponent 層級中，您只能建立或取代元件。該實體必須已經存在，否則動作將產生一個 ValidationException。

在屬性或關係層級中，您只能建立或取代屬性或關係，且包含的內容 EntityComponent 必須已存在。

- 子資源刪除：

AWS IoT TwinMaker 還支持子資源刪除。子資源可以是元件、屬性或關係。

如果要刪除元件，則必須從圖元層級執行。

如果要刪除屬性或關係，則必須從「實體」或「EntityComponent 層級」執行此操作。

若要刪除子資源，請更新較高層級的資源，並省略子資源的定義。

- 不刪除頂層資源：永遠不 AWS IoT TwinMaker 會刪除頂層資源。頂層資源是指實體或 ComponentType。
- 在一個模板中沒有相同頂級資源的子資源定義：

您無法在相同範本中提供相同實體的完整實體定義和子資源 (例如屬性) 定義。

如果在實體中使用 EntityId，則無法在實體、EntityComponent 屬性或關聯中使用相同的 ID。

如果在中使用實體 ID 或元件名稱組合，則無法在 EntityComponent、屬性或關係中使用相同的 EntityComponent 組合。

如果在屬性或關係中使用實體 ID、元件名稱、屬性名稱組合，則無法在屬性或關係中使用相同的組合。

- ExternalId 是可選的 AWS IoT TwinMaker：ExternalId 可用於幫助您識別資源。

## 執行大量匯入和匯出作業

本主題說明如何執行大量匯入和匯出作業，以及如何處理移轉工作中的錯誤。它提供了使用 CLI 命令傳輸作業的示例。

AWS IoT TwinMaker API 參考包含有關 [CreateMetadataTransferJob](#) 和其他 API 動作的資訊。

### 主題

- [metadataTransferJob 前提](#)
- [IAM 許可](#)
- [執行大量作業](#)
- [錯誤處理](#)
- [匯入詮釋資料樣](#)
- [AWS IoT TwinMaker metadataTransferJob 例子](#)

## metadataTransferJob 前提

請先完成下列先決條件，然後再執行 metadataTransferJob：

- 建立 AWS IoT TwinMaker 工作區。工作區可以是匯入目標或匯出來源 metadataTransferJob。如需有關建立工作區的資訊，請參閱 [建立工作區](#)。
- 建立 Amazon S3 儲存貯體來存放資源。有關使用 Amazon S3 的更多信息，請參閱 [什麼是 Amazon S3 ?](#)

## IAM 許可

執行大量操作時，您需要建立具有許可的 IAM 政策，以便在 Amazon S3、AWS IoT TwinMaker AWS IoT SiteWise、和本機電腦之間交換 AWS 資源。如需建立 IAM 政策的詳細資訊，請參閱[建立 IAM 政策](#)。

AWS IoT SiteWise 和 Amazon S3 的政策聲明如 AWS IoT TwinMaker 下所示：

- AWS IoT TwinMaker 政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:GetObject",
      "s3:GetBucketLocation",
      "s3:ListBucket",
      "s3:AbortMultipartUpload",
      "s3:ListBucketMultipartUploads",
      "s3:ListMultipartUploadParts"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iottwinmaker:GetWorkspace",
      "iottwinmaker:CreateEntity",
      "iottwinmaker:GetEntity",
      "iottwinmaker:UpdateEntity",
      "iottwinmaker:GetComponentType",
      "iottwinmaker:CreateComponentType",
      "iottwinmaker:UpdateComponentType",
      "iottwinmaker:ListEntities",
      "iottwinmaker:ListComponentTypes",
      "iottwinmaker:ListTagsForResource",
      "iottwinmaker:TagResource",
      "iottwinmaker:UntagResource"
    ],
    "Resource": "*"
  }
}
```

```
]
}
```

- AWS IoT SiteWise 政策 :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:GetObject",
      "s3:GetBucketLocation",
      "s3:ListBucket",
      "s3:AbortMultipartUpload",
      "s3:ListBucketMultipartUploads",
      "s3:ListMultipartUploadParts"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:CreateAsset",
      "iotsitewise:CreateAssetModel",
      "iotsitewise:UpdateAsset",
      "iotsitewise:UpdateAssetModel",
      "iotsitewise:UpdateAssetProperty",
      "iotsitewise:ListAssets",
      "iotsitewise:ListAssetModels",
      "iotsitewise:ListAssetProperties",
      "iotsitewise:ListAssetModelProperties",
      "iotsitewise:ListAssociatedAssets",
      "iotsitewise:DescribeAsset",
      "iotsitewise:DescribeAssetModel",
      "iotsitewise:DescribeAssetProperty",
      "iotsitewise:AssociateAssets",
      "iotsitewise:DisassociateAssets",
      "iotsitewise:AssociateTimeSeriesToAssetProperty",
      "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
      "iotsitewise:BatchPutAssetPropertyValue",
      "iotsitewise:BatchGetAssetPropertyValue",
      "iotsitewise:TagResource",
      "iotsitewise:UntagResource",

```

```

        "iotsitewise:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}

```

- Amazon S3 政策：

```

{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}

```

或者，您可以將 Amazon S3 政策的範圍設定為僅存取單一 Amazon S3 儲存貯體，請參閱以下政策。

#### Amazon S3 單儲存桶範圍政策

```

{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": [
    "arn:aws:s3:::bucket name",
    "arn:aws:s3:::bucket name/*"
  ]
}

```

```
}
```

## 設定存取控制 metadataTransferJob

若要控制使用者可存取的工作類型，請將下列 IAM 政策新增至用於呼叫的角色 AWS IoT TwinMaker。

### Note

此政策僅允許存取將資源傳 AWS IoT TwinMaker 入和傳出 Amazon S3 的匯入和匯出任務。

```
{
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:*DataTransferJob*"
  ],
  "Resource": "*",
  "Condition": {
    "StringLikeIfExists": {
      "iottwinmaker:sourceType": [
        "s3",
        "iottwinmaker"
      ],
      "iottwinmaker:destinationType": [
        "iottwinmaker",
        "s3"
      ]
    }
  }
}
```

## 執行大量作業

本節介紹如何執行大量匯入和匯出作業。

將資料從 Amazon S3 匯入到 AWS IoT TwinMaker

1. 指定要使用結構描述傳輸的 AWS IoT TwinMaker metadataTransferJob 資源。建立您的結構描述檔案並將其存放在 Amazon S3 儲存貯體中。

如需資料架構範例，請參閱[匯入詮釋資料樣](#)。



2. 建立要求主體並將其儲存為 JSON 檔案。要求主體會指定傳送工作的來源和目的地。請務必將 Amazon S3 儲存貯體指定為來源，並將 AWS IoT TwinMaker 工作區指定為目的地。

以下是要求主體的範例：

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::your-S3-bucket-name/your_import_data.json"
    }
  ]},
  "destination": {
    "type": "iottwinmaker",
    "iotTwinMakerConfiguration": {
      "workspace": "arn:aws:iottwinmaker:us-east-1:111122223333:workspace/your-worksapce-name"
    }
  }
}
```

記錄您為請求主體提供的文件名，下一步將需要它。在這個例子中，請求主體被命名為 `createMetadataTransferJobImport.json`。

3. 運行以下 CLI 命令以調用 `CreateMetadataTransferJob` (將輸入 json 文件名替換為您給請求主體的名稱)：

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
--cli-input-json file://createMetadataTransferJobImport.json
```

這會建立 `metadataTransferJob` 並開始移轉所選資源的程序。

將資料從 AWS IoT TwinMaker Amazon S3 匯出

1. 使用適當的篩選器建立 JSON 要求主體，以選擇您要匯出的資源。在這個例子中，我們使用：

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iottwinmaker",
```

```
    "iotTwinMakerConfiguration": {
      "workspace": "arn:aws:iottwinmaker:us-
east-1:111122223333:workspace/your-workspace-name",
      "filters": [{
        "filterByEntity": {
          "entityId": "parent"
        }
      },
      {
        "filterByEntity": {
          "entityId": "child"
        }
      },
      {
        "filterByComponentType": {
          "componentTypeId": "component.type.minimal"
        }
      }
    ]
  }
},
"destination": {
  "type": "s3",
  "s3Configuration": {
    "location": "arn:aws:s3:::your-S3-bucket-location"
  }
}
}
```

`filters` 陣列可讓您指定要匯出的資源。在此範例中，我們篩選了 `entity`、`componentType`。

確保將您的工作 AWS IoT TwinMaker 區指定為來源，並將 Amazon S3 儲存貯體指定為中繼資料傳輸任務的目的地。

保存您的請求正文並記錄文件名，您將在下一步中使用它。在這個例子中，我們命名我們的請求體 `createMetadataTransferJobExport.json`。

2. 運行以下 CLI 命令以調用 `CreateMetadataTransferJob` ( 將輸入 `json` 文件名替換為您給請求主體的名稱 ) :

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
--cli-input-json file://createMetadataTransferJobExport.json
```

這會建立 `metadataTransferJob` 並開始移轉所選資源的程序。

若要檢查或更新移轉工作的狀態，請使用下列指令：

- 若要取消工作，請使用 [CancelMetadataTransferJob](#) API 動作。當您呼叫時 `CancelMetadataTransferJob`，API 只會取消執行中 `metadataTransferJob`，且任何已匯出或匯入的資源都不會受此 API 呼叫的影響。
- 若要擷取特定工作的資訊，請使用 [GetMetadataTransferJob](#) API 動作。

或者，您也可以使用下列 CLI 指令呼叫 `GetMetadataTransferJob` 現有的傳送工作：

```
aws iottwinmaker get-metadata-transfer-job --job-id ExistingJobId
```

如果您呼叫 `GetMetadataTransferJob` 不存在的 AWS IoT TwinMaker 匯入或匯出工作，您會收到回應 `ResourceNotFoundException` 錯誤。

- 若要列出目前的工作，請使用 [ListMetadataTransferJobs](#) API 動作。

以下是呼叫 `ListMetadataTransferJobs` AWS IoT TwinMaker 作為目標類型和 `s3` `sourceType` 的 CLI 範例：

```
aws iottwinmaker list-metadata-transfer-jobs --destination-type iottwinmaker --source-type s3
```

#### Note

您可以變更「`sourceType`」和「目的地類型」參數的值，以符合匯入或匯出工作的來源和目的地。

如需呼叫這些 API 動作的 CLI 命令的更多範例，請參閱 [AWS IoT TwinMaker metadataTransferJob 例子](#)。

如果在傳送工作期間遇到任何錯誤，請參閱 [錯誤處理](#)。

## 錯誤處理

建立並執行移轉工作之後，您可以呼叫 `GetMetadataTransferJob` 以診斷發生的任何錯誤：

```
aws iottwinmaker get-metadata-transfer-job \  
--metadata-transfer-job-id your_metadata_transfer_job_id \  

```

```
--region us-east-1
```

一旦您看到工作轉向的狀態COMPLETED，就可以驗證工作的結果。GetMetadataTransferJob 會傳回名為的物件 [MetadataTransferJobProgress](#)，其中包含下列欄位：

- 失敗計數：表示在傳輸過程中失敗的資源數目。
- 跳過計數：表示在移轉處理期間略過的資源數目。
- 成功計數：表示在傳輸過程中成功的資源數目。
- 總計數：表示移轉處理中涉及的資源總數。

此外，會傳回包含預先簽署網址的 ReportUrl 元素。如果您的移轉工作有您想要進一步調查的錯誤，您可以使用此 URL 下載完整的錯誤報告。

## 匯入詮釋資料樣

您可以使用單一大量匯入作業匯入許多元件、元件類型或實體。本節中的範例說明如何執行此操作。

template: Importing entities

匯入實體的工作使用下列範本格式：

```
{
  "entities": [
    {
      "description": "string",
      "entityId": "string",
      "entityName": "string",
      "parentEntityId": "string",
      "tags": {
        "string": "string"
      },
      "components": {
        "string": {
          "componentTypeId": "string",
          "description": "string",
          "properties": {
            "string": {
              "definition": {
                "configuration": {
                  "string": "string"
                }
              }
            }
          }
        }
      }
    }
  ]
}
```

```

        "dataType": "DataType",
        "defaultValue": "DataValue",
        "displayName": "string",
        "isExternalId": "boolean",
        "isRequiredInEntity": "boolean",
        "isStoredExternally": "boolean",
        "isTimeSeries": "boolean"
    },
    "value": "DataValue"
}
},
"propertyGroups": {
    "string": {
        "groupType": "string",
        "propertyNames": [
            "string"
        ]
    }
}
}
}
}
]
}
}

```

## template: Importing componentTypes

對於匯入元件類型的工作，請使用下列範本格式：

```

{
  "componentTypes": [
    {
      "componentTypeId": "string",
      "componentTypeName": "string",
      "description": "string",
      "extendsFrom": [
        "string"
      ],
      "functions": {
        "string": {
          "implementedBy": {
            "isNative": "boolean",
            "lambda": {
              "functionName": "Telemetry-tsDataReader",

```

```

        "arn": "Telemetry-tsDataReaderARN"
      }
    },
    "requiredProperties": [
      "string"
    ],
    "scope": "string"
  }
},
"isSingleton": "boolean",
"propertyDefinitions": {
  "string": {
    "configuration": {
      "string": "string"
    },
    "dataType": "DataType",
    "defaultValue": "DataValue",
    "displayName": "string",
    "isExternalId": "boolean",
    "isRequiredInEntity": "boolean",
    "isStoredExternally": "boolean",
    "isTimeSeries": "boolean"
  }
},
"propertyGroups": {
  "string": {
    "groupType": "string",
    "propertyNames": [
      "string"
    ]
  }
},
"tags": {
  "string": "string"
}
}
]
}

```

### template: Importing components

對於匯入元件的工作，請使用下列範本格式：

```
{
```

```
"entityComponents": [  
  {  
    "entityId": "string",  
    "componentName": "string",  
    "componentTypeId": "string",  
    "description": "string",  
    "properties": {  
      "string": {  
        "definition": {  
          "configuration": {  
            "string": "string"  
          },  
          "dataType": "DataType",  
          "defaultValue": "DataValue",  
          "displayName": "string",  
          "isExternalId": "boolean",  
          "isRequiredInEntity": "boolean",  
          "isStoredExternally": "boolean",  
          "isTimeSeries": "boolean"  
        },  
        "value": "DataValue"  
      }  
    },  
    "propertyGroups": {  
      "string": {  
        "groupType": "string",  
        "propertyNames": [  
          "string"  
        ]  
      }  
    }  
  }  
]
```

## AWS IoT TwinMaker metadataTransferJob 例子

使用下列指令來管理中繼資料傳輸：

- [CreateMetadataTransferJob](#) API 動作。

CLI 指令範例：

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \  
--cli-input-json file://yourTransferFileName.json
```

- 若要取消工作，請使用 [CancelMetadataTransferJob](#) API 動作。

CLI 指令範例：

```
aws iottwinmaker cancel-metadata-transfer-job  
--region us-east-1 \  
--metadata-transfer-job-id job-to-cancel-id
```

通話時 `CancelMetadataTransferJob`，只會取消特定的中繼資料傳輸工作，而任何已匯出或匯入的資源都不會受到影響。

- 若要擷取特定工作的資訊，請使用 [GetMetadataTransferJob](#) API 動作。

CLI 指令範例：

```
aws iottwinmaker get-metadata-transfer-job \  
--metadata-transfer-job-id your_metadata_transfer_job_id \  
--region us-east-1 \  

```

- 若要列出目前的工作，請使用 [ListMetadataTransferJobs](#) API 動作。

您可以 `ListMetadataTransferJobs` 使用 JSON 檔案篩選傳回的結果。請參閱使用 CLI 的下列程序：

1. 建立 CLI 輸入 JSON 檔案以指定您要使用的篩選器：

```
{  
  "sourceType": "s3",  
  "destinationType": "iottwinmaker",  
  "filters": [{  
    "workspaceId": "workspaceforbulkimport"  
  }],  
  {  
    "state": "COMPLETED"  
  }  
}]
```

保存並記錄文件名，輸入 CLI 命令時將需要它。

2. 使用 JSON 檔案做為下列 CLI 命令的引數：



```
aws iottwinmaker list-metadata-transfer-job --region us-east-1 \  
--cli-input-json file://ListMetadataTransferJobsExample.json
```

## AWS IoT TwinMaker 詮釋資料傳送工作網

metadataTransferJob 匯入結構描述：當您將資料上傳到 Amazon S3 儲存貯體時，請使用此 AWS IoT TwinMaker 中繼資料架構來驗證資料：

```
{  
  "$schema": "https://json-schema.org/draft/2020-12/schema",  
  "title": "IoT TwinMaker",  
  "description": "Metadata transfer job resource schema for IoT TwinMaker",  
  "definitions": {  
    "ExternalId": {  
      "type": "string",  
      "minLength": 1,  
      "maxLength": 128,  
      "pattern": "[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+"  
    },  
    "Description": {  
      "type": "string",  
      "minLength": 0,  
      "maxLength": 512  
    },  
    "DescriptionWithDefault": {  
      "type": "string",  
      "minLength": 0,  
      "maxLength": 512,  
      "default": ""  
    },  
    "ComponentTypeName": {  
      "description": "A friendly name for the component type.",  
      "type": "string",  
      "pattern": ".*[^\u0000-\\u001F\\u007F]*.*",  
      "minLength": 1,  
      "maxLength": 256  
    },  
    "ComponentTypeId": {  
      "description": "The ID of the component type.",  
      "type": "string",  
      "pattern": "[a-zA-Z_\\.\\-0-9:]+",  
    }  
  }  
}
```

```

    "minLength": 1,
    "maxLength": 256
  },
  "ComponentName": {
    "description": "The name of the component.",
    "type": "string",
    "pattern": "[a-zA-Z_\\-0-9]+",
    "minLength": 1,
    "maxLength": 256
  },
  "EntityId": {
    "description": "The ID of the entity.",
    "type": "string",
    "minLength": 1,
    "maxLength": 128,
    "pattern": "[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|^[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+"
  },
  "EntityName": {
    "description": "The name of the entity.",
    "type": "string",
    "minLength": 1,
    "maxLength": 256,
    "pattern": "[a-zA-Z_0-9-\\.][a-zA-Z_0-9-\\. ]*[a-zA-Z0-9]+"
  },
  "ParentEntityId": {
    "description": "The ID of the parent entity.",
    "type": "string",
    "minLength": 1,
    "maxLength": 128,
    "pattern": "\\$ROOT|^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|^[a-zA-Z0-9][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9]+",
    "default": "$ROOT"
  },
  "DisplayName": {
    "description": "A friendly name for the property.",
    "type": "string",
    "pattern": ".*[^\u0000-\u001F\u007F]*.*",
    "minLength": 0,
    "maxLength": 256
  },
  "Tags": {
    "description": "Metadata that you can use to manage the entity / componentType",
    "patternProperties": {

```

```

    "^(\\p{L}\\p{Z}\\p{N}_./=+\\-@)*$": {
      "type": "string",
      "minLength": 1,
      "maxLength": 256
    }
  },
  "existingJavaType": "java.util.Map<String,String>",
  "minProperties": 0,
  "maxProperties": 50
},
"Relationship": {
  "description": "The type of the relationship.",
  "type": "object",
  "properties": {
    "relationshipType": {
      "description": "The type of the relationship.",
      "type": "string",
      "pattern": ".*",
      "minLength": 1,
      "maxLength": 256
    },
    "targetComponentTypeId": {
      "description": "The ID of the target component type associated with this
relationship.",
      "$ref": "#/definitions/ComponentTypeId"
    }
  },
  "additionalProperties": false
},
"DataValue": {
  "description": "An object that specifies a value for a property.",
  "type": "object",
  "properties": {
    "booleanValue": {
      "description": "A Boolean value.",
      "type": "boolean"
    },
    "doubleValue": {
      "description": "A double value.",
      "type": "number"
    },
    "expression": {
      "description": "An expression that produces the value.",
      "type": "string",

```

```

    "pattern": "(^\\$\\{Parameters\\. [a-zA-z]+([a-zA-z_0-9]*)\\}$)",
    "minLength": 1,
    "maxLength": 316
  },
  "integerValue": {
    "description": "An integer value.",
    "type": "integer"
  },
  "listValue": {
    "description": "A list of multiple values.",
    "type": "array",
    "minItems": 0,
    "maxItems": 50,
    "uniqueItems": false,
    "insertionOrder": false,
    "items": {
      "$ref": "#/definitions/DataValue"
    },
    "default": null
  },
  "longValue": {
    "description": "A long value.",
    "type": "integer",
    "existingJavaType": "java.lang.Long"
  },
  "stringValue": {
    "description": "A string value.",
    "type": "string",
    "pattern": ".*",
    "minLength": 1,
    "maxLength": 256
  },
  "mapValue": {
    "description": "An object that maps strings to multiple DataValue objects.",
    "type": "object",
    "patternProperties": {
      "[a-zA-Z_\\-0-9]+": {
        "$ref": "#/definitions/DataValue"
      }
    },
    "additionalProperties": {
      "$ref": "#/definitions/DataValue"
    }
  },
},

```

```

    "relationshipValue": {
      "description": "A value that relates a component to another component.",
      "type": "object",
      "properties": {
        "TargetComponentName": {
          "type": "string",
          "pattern": "[a-zA-Z_\\-0-9]+",
          "minLength": 1,
          "maxLength": 256
        },
        "TargetEntityId": {
          "type": "string",
          "pattern": "[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|
^[a-zA-Z0-9][a-zA-Z_\\-0-9.]*[a-zA-Z0-9]+",
          "minLength": 1,
          "maxLength": 128
        }
      },
      "additionalProperties": false
    },
    "additionalProperties": false
  },
  "DataType": {
    "description": "An object that specifies the data type of a property.",
    "type": "object",
    "properties": {
      "allowedValues": {
        "description": "The allowed values for this data type.",
        "type": "array",
        "minItems": 0,
        "maxItems": 50,
        "uniqueItems": false,
        "insertionOrder": false,
        "items": {
          "$ref": "#/definitions/DataValue"
        }
      },
      "default": null
    },
    "nestedType": {
      "description": "The nested type in the data type.",
      "$ref": "#/definitions/DataType"
    },
    "relationship": {

```

```
    "description": "A relationship that associates a component with another
component.",
    "$ref": "#/definitions/Relationship"
  },
  "type": {
    "description": "The underlying type of the data type.",
    "type": "string",
    "enum": [
      "RELATIONSHIP",
      "STRING",
      "LONG",
      "BOOLEAN",
      "INTEGER",
      "DOUBLE",
      "LIST",
      "MAP"
    ]
  },
  "unitOfMeasure": {
    "description": "The unit of measure used in this data type.",
    "type": "string",
    "pattern": ".*",
    "minLength": 1,
    "maxLength": 256
  }
},
"required": [
  "type"
],
"additionalProperties": false
},
"PropertyDefinition": {
  "description": "An object that specifies information about a property.",
  "type": "object",
  "properties": {
    "configuration": {
      "description": "An object that specifies information about a property.",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "type": "string",
          "pattern": "[a-zA-Z_\\-0-9]+",
          "minLength": 1,
          "maxLength": 256
        }
      }
    }
  }
}
```

```
    },
    "existingJavaType": "java.util.Map<String,String>"
  },
  "dataType": {
    "description": "An object that contains information about the data type.",
    "$ref": "#/definitions/DataType"
  },
  "defaultValue": {
    "description": "An object that contains the default value.",
    "$ref": "#/definitions/DataValue"
  },
  "displayName": {
    "description": "An object that contains the default value.",
    "$ref": "#/definitions/DisplayName"
  },
  "isExternalId": {
    "description": "A Boolean value that specifies whether the property ID comes
from an external data store.",
    "type": "boolean",
    "default": null
  },
  "isRequiredInEntity": {
    "description": "A Boolean value that specifies whether the property is
required.",
    "type": "boolean",
    "default": null
  },
  "isStoredExternally": {
    "description": "A Boolean value that specifies whether the property is stored
externally.",
    "type": "boolean",
    "default": null
  },
  "isTimeSeries": {
    "description": "A Boolean value that specifies whether the property consists
of time series data.",
    "type": "boolean",
    "default": null
  }
},
"additionalProperties": false
},
"PropertyDefinitions": {
  "type": "object",
```

```

    "patternProperties": {
      "[a-zA-Z_\\-0-9]+": {
        "$ref": "#/definitions/PropertyDefinition"
      }
    },
    "additionalProperties": {
      "$ref": "#/definitions/PropertyDefinition"
    }
  },
  "Property": {
    "type": "object",
    "properties": {
      "definition": {
        "description": "The definition of the property",
        "$ref": "#/definitions/PropertyDefinition"
      },
      "value": {
        "description": "The value of the property.",
        "$ref": "#/definitions/DataValue"
      }
    },
    "additionalProperties": false
  },
  "Properties": {
    "type": "object",
    "patternProperties": {
      "[a-zA-Z_\\-0-9]+": {
        "$ref": "#/definitions/Property"
      }
    },
    "additionalProperties": {
      "$ref": "#/definitions/Property"
    }
  },
  "PropertyName": {
    "type": "string",
    "pattern": "[a-zA-Z_\\-0-9]+"
  },
  "PropertyGroup": {
    "description": "An object that specifies information about a property group.",
    "type": "object",
    "properties": {
      "groupType": {
        "description": "The type of property group.",

```



```
    "type": "string",
    "enum": [
      "TABULAR"
    ]
  },
  "propertyNames": {
    "description": "The list of property names in the property group.",
    "type": "array",
    "minItems": 1,
    "maxItems": 256,
    "uniqueItems": true,
    "insertionOrder": false,
    "items": {
      "$ref": "#/definitions/PropertyName"
    },
    "default": null
  }
},
"additionalProperties": false
},
"PropertyGroups": {
  "type": "object",
  "patternProperties": {
    "[a-zA-Z_\\-0-9]+": {
      "$ref": "#/definitions/PropertyGroup"
    }
  },
  "additionalProperties": {
    "$ref": "#/definitions/PropertyGroup"
  }
},
"Component": {
  "type": "object",
  "properties": {
    "componentTypeId": {
      "$ref": "#/definitions/ComponentTypeId"
    },
    "description": {
      "$ref": "#/definitions/Description"
    },
    "properties": {
      "description": "An object that maps strings to the properties to set in the component type. Each string in the mapping must be unique to this object.",
      "$ref": "#/definitions/Properties"
    }
  }
}
```

```

    },
    "propertyGroups": {
      "description": "An object that maps strings to the property groups to set in
the entity component. Each string in the mapping must be unique to this object.",
      "$ref": "#/definitions/PropertyGroups"
    }
  },
  "required": [
    "componentTypeId"
  ],
  "additionalProperties": false
},
"RequiredProperty": {
  "type": "string",
  "pattern": "[a-zA-Z_\\-0-9]+"
},
"LambdaFunction": {
  "type": "object",
  "properties": {
    "arn": {
      "type": "string",
      "pattern": "arn:((aws)|(aws-cn)|(aws-us-gov)|(\\"{partition})):lambda:(([a-
z0-9-]+)|(\\"{region})):([0-9]{12}|(\\"{accountId})):function:[/a-zA-Z0-9_-]+",
      "minLength": 1,
      "maxLength": 128
    }
  },
  "additionalProperties": false,
  "required": [
    "arn"
  ]
},
"DataConnector": {
  "description": "The data connector.",
  "type": "object",
  "properties": {
    "isNative": {
      "description": "A Boolean value that specifies whether the data connector is
native to IoT TwinMaker.",
      "type": "boolean"
    },
    "lambda": {
      "description": "The Lambda function associated with this data connector.",
      "$ref": "#/definitions/LambdaFunction"
    }
  }
}

```

```
    }
  },
  "additionalProperties": false
},
"Function": {
  "description": "The function of component type.",
  "type": "object",
  "properties": {
    "implementedBy": {
      "description": "The data connector.",
      "$ref": "#/definitions/DataConnector"
    },
    "requiredProperties": {
      "description": "The required properties of the function.",
      "type": "array",
      "minItems": 1,
      "maxItems": 256,
      "uniqueItems": true,
      "insertionOrder": false,
      "items": {
        "$ref": "#/definitions/RequiredProperty"
      },
      "default": null
    },
    "scope": {
      "description": "The scope of the function.",
      "type": "string",
      "enum": [
        "ENTITY",
        "WORKSPACE"
      ]
    }
  },
  "additionalProperties": false
},
"Entity": {
  "type": "object",
  "properties": {
    "description": {
      "description": "The description of the entity.",
      "$ref": "#/definitions/DescriptionWithDefault"
    },
    "entityId": {
      "$ref": "#/definitions/EntityId"
    }
  }
}
```

```
    },
    "entityExternalId": {
      "description": "The external ID of the entity.",
      "$ref": "#/definitions/ExternalId"
    },
    },
    "entityName": {
      "$ref": "#/definitions/EntityName"
    },
    },
    "parentEntityId": {
      "$ref": "#/definitions/ParentEntityId"
    },
    },
    "tags": {
      "$ref": "#/definitions/Tags"
    },
    },
    "components": {
      "description": "A map that sets information about a component.",
      "type": "object",
      "patternProperties": {
        "[a-zA-Z_\\-0-9]+": {
          "$ref": "#/definitions/Component"
        }
      },
      },
      "additionalProperties": {
        "$ref": "#/definitions/Component"
      }
    }
  },
  "required": [
    "entityId",
    "entityName"
  ],
  "additionalProperties": false
},
"ComponentType": {
  "type": "object",
  "properties": {
    "description": {
      "description": "The description of the component type.",
      "$ref": "#/definitions/DescriptionWithDefault"
    },
    },
    "componentTypeId": {
      "$ref": "#/definitions/ComponentTypeId"
    },
    },
    "componentTypeExternalId": {
```

```
    "description": "The external ID of the component type.",
    "$ref": "#/definitions/ExternalId"
  },
  "componentTypeName": {
    "$ref": "#/definitions/ComponentTypeName"
  },
  "extendsFrom": {
    "description": "Specifies the parent component type to extend.",
    "type": "array",
    "minItems": 1,
    "maxItems": 256,
    "uniqueItems": true,
    "insertionOrder": false,
    "items": {
      "$ref": "#/definitions/ComponentTypeId"
    },
    "default": null
  },
  "functions": {
    "description": "A Map of functions in the component type. Each function's key
must be unique to this map.",
    "type": "object",
    "patternProperties": {
      "[a-zA-Z_\\-0-9]+": {
        "$ref": "#/definitions/Function"
      }
    },
    "additionalProperties": {
      "$ref": "#/definitions/Function"
    }
  },
  "isSingleton": {
    "description": "A Boolean value that specifies whether an entity can have
more than one component of this type.",
    "type": "boolean",
    "default": false
  },
  "propertyDefinitions": {
    "description": "An map of the property definitions in the component type.
Each property definition's key must be unique to this map.",
    "$ref": "#/definitions/PropertyDefinitions"
  },
  "propertyGroups": {
```

```
    "description": "An object that maps strings to the property groups to set in
the component type. Each string in the mapping must be unique to this object.",
    "$ref": "#/definitions/PropertyGroups"
  },
  "tags": {
    "$ref": "#/definitions/Tags"
  }
},
"required": [
  "componentTypeId"
],
"additionalProperties": false
},
"EntityComponent": {
  "type": "object",
  "properties": {
    "entityId": {
      "$ref": "#/definitions/EntityId"
    },
    "componentName": {
      "$ref": "#/definitions/ComponentName"
    },
    "componentExternalId": {
      "description": "The external ID of the component.",
      "$ref": "#/definitions/ExternalId"
    },
    "componentTypeId": {
      "$ref": "#/definitions/ComponentTypeId"
    },
    "description": {
      "description": "The description of the component.",
      "$ref": "#/definitions/Description"
    },
    "properties": {
      "description": "An object that maps strings to the properties to set in the
component. Each string in the mapping must be unique to this object.",
      "$ref": "#/definitions/Properties"
    },
    "propertyGroups": {
      "description": "An object that maps strings to the property groups to set in
the component. Each string in the mapping must be unique to this object.",
      "$ref": "#/definitions/PropertyGroups"
    }
  }
},
```

```

    "required": [
      "entityId",
      "componentTypeId",
      "componentName"
    ],
    "additionalProperties": false
  }
},
"additionalProperties": false,
"properties": {
  "entities": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/Entity"
    }
  },
  "componentTypes": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/ComponentType"
    }
  },
  "entityComponents": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/EntityComponent"
    },
    "default": null
  }
}
}
}

```

下面是創建一個名為的新 ComponEntType component.type.initial 並創建一個名為的實體的示例：initial

```

{
  "componentTypes": [
    {
      "componentTypeId": "component.type.initial",
      "tags": {

```

```
        "key": "value"
      }
    }
  ],
  "entities": [
    {
      "entityName": "initial",
      "entityId": "initial"
    }
  ]
}
```

以下是更新現有實體的範例：

```
{
  "componentTypes": [
    {
      "componentTypeId": "component.type.initial",
      "description": "updated"
    }
  ],
  "entities": [
    {
      "entityName": "parent",
      "entityId": "parent"
    },
    {
      "entityName": "child",
      "entityId": "child",
      "components": {
        "testComponent": {
          "componentTypeId": "component.type.initial",
          "properties": {
            "testProperty": {
              "definition": {
                "configuration": {
                  "alias": "property"
                },
                "dataType": {
                  "relationship": {
                    "relationshipType": "parent",
                    "targetComponentTypeId": "test"
                  }
                }
              }
            }
          }
        }
      }
    }
  ]
}
```



```
        "type": "STRING",
        "unitOfMeasure": "t"
    },
    "displayName": "displayName"
}
}
}
},
"parentEntityId": "parent"
}
],
"entityComponents": [
{
    "entityId": "initial",
    "componentTypeId": "component.type.initial",
    "componentName": "entityComponent",
    "description": "additionalDescription",
    "properties": {
        "additionalProperty": {
            "definition": {
                "configuration": {
                    "alias": "additionalProperty"
                },
                "dataType": {
                    "type": "STRING"
                },
                "displayName": "additionalDisplayName"
            },
            "value": {
                "stringValue": "test"
            }
        }
    }
}
]
}
```

# AWS IoT TwinMaker 資料連接器

AWS IoT TwinMaker 使用以連接器為基礎的架構，以便您可以將資料從自己的資料倉庫連接到。AWS IoT TwinMaker 這意味著您不需要在使用之前遷移數據 AWS IoT TwinMaker。目前，AWS IoT TwinMaker 支援的第一方連接器 AWS IoT SiteWise。如果您將建模和屬性資料儲存在中 AWS IoT SiteWise，則不需要實作自己的連接器。如果您將模型或屬性資料儲存在其他資料存放區 (例如 Timestream、DynamoDB 或 Snowflake) 中，則必須使用 AWS IoT TwinMaker 資料 AWS Lambda 連接器介面實作連接器，AWS IoT TwinMaker 以便在必要時呼叫連接器。

## 主題

- [AWS IoT TwinMaker 資料連接器](#)
- [AWS IoT TwinMaker Athena 表格資料連接](#)
- [開發 AWS IoT TwinMaker 時間序列資料連接器](#)

# AWS IoT TwinMaker 資料連接器

連接器需要存取您的基礎資料存放區，才能解決傳送的查詢，並傳回結果或錯誤。

若要瞭解可用的連接器、其要求介面及其回應介面，請參閱下列主題。

如需有關連接器介面中使用之內容的詳細資訊，請參閱 [GetPropertyValueHistory](#) API 動作。

### Note

某些連接器在請求和回應介面中都有兩個時間戳記欄位，用於開始時間和結束時間屬性。startDateTime 和 endDateTime 使用較長的數字來表示第二個紀元，這不再受支持。為了保持向後兼容性，我們仍然會向該字段發送時間戳記值，但我們建議使用與 API 時間戳記格式一致的 startTime 和 endTime 字段。

## 主題

- [結構描述初始設定式](#)
- [DataReaderByEntity](#)
- [DataReaderByComponentType](#)
- [DataReader](#)

- [AttributePropertyValueReaderByEntity](#)
- [DataWriter](#)
- [範例](#)

## 結構描述初始設定式

您可以在元件類型或實體生命週期中使用結構描述初始設定式，從基礎資料來源擷取元件類型或元件屬性。結構描述初始設定式會自動匯入元件類型或元件屬性，而無需明確呼叫 API 動作進行設定。

### SchemaInitializer 請求接口

```
{
  "workspaceId": "string",
  "entityId": "string",
  "componentName": "string",
  "properties": {
    // property name as key,
    // value is of type PropertyRequest
    "string": "PropertyRequest"
  }
}
```

#### Note

在這個請求接口屬性的映射是一個PropertyRequest。如需詳細資訊，請參閱[PropertyRequest](#)。

### SchemaInitializer 響應接口

```
{
  "properties": {
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
  }
}
```

**Note**

在這個請求接口屬性的映射是一個PropertyResponse。如需詳細資訊，請參閱[PropertyResponse](#)。

## DataReaderByEntity

DataReaderByEntity 是資料平面連接器，用來取得單一元件中屬性的時間序列值。

如需有關此連接器的內容類型、語法和格式的資訊，請參閱 [GetPropertyValueHistory](#) API 動作。

### DataReaderByEntity 請求接口

```
{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": {
    // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
  },
  "workspaceId": "string",
  "selectedProperties": List:"string",
  "propertyFilters": List:PropertyFilter,
  "entityId": "string",
  "componentName": "string",
  "componentTypeId": "string",
  "interpolation": InterpolationParameters,
  "nextToken": "string",
  "maxResults": int,
  "orderByTime": "string"
}
```

### DataReaderByEntity 響應接口

```
{
  "propertyValues": [
    {
```

```

    "entityPropertyReference": EntityPropertyReference, // The same
as EntityPropertyReference
    "values": [
      {
        "timestamp": long, // Epoch sec, deprecated
        "time": "string", // ISO-8601 timestamp format
        "value": DataValue // The same as DataValue
      }
    ]
  }
],
"nextToken": "string"
}

```

## DataReaderByComponentType

若要取得來自相同元件類型之一般屬性的時間序列值，請使用資料平面連接器 `DataReaderByEntity`。例如，如果您在元件類型中定義時間序列屬性，而且您有多個使用該元件類型的元件，則您可以查詢指定時間範圍內所有元件的這些屬性。一個常見的使用案例是當您想要查詢多個元件的警示狀態以取得實體的全域檢視時。

如需有關此連接器的內容類型、語法和格式的資訊，請參閱 [GetPropertyValueHistory](#) API 動作。

### DataReaderByComponentType 請求接口

```

{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": { // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyResponse
    "string": "PropertyResponse"
  },
  "workspaceId": "string",
  "selectedProperties": List:"string",
  "propertyFilters": List:PropertyFilter,
  "componentTypeId": "string",
  "interpolation": InterpolationParameters,
  "nextToken": "string",
  "maxResults": int,
  "orderByTime": "string"
}

```

```
}

```

## DataReaderByComponentType 響應接口

```
{
  "propertyValues": [
    {
      "entityPropertyReference": EntityPropertyReference, // The same
      as EntityPropertyReference
      "entityId": "string",
      "componentName": "string",
      "values": [
        {
          "timestamp": long, // Epoch sec, deprecated
          "time": "string", // ISO-8601 timestamp format
          "value": DataValue // The same as DataValue
        }
      ]
    }
  ],
  "nextToken": "string"
}
```

## DataReader

DataReader 是可同時處理和案例的資料平面連接 DataReaderByEntity 器 DataReaderByComponentType。

如需有關此連接器的內容類型、語法和格式的資訊，請參閱 [GetPropertyValueHistory](#) API 動作。

## DataReader 請求接口

EntityId 和 componentName 是選擇性使用的。

```
{
  "startDateTime": long, // In epoch sec, deprecated
  "startTime": "string", // ISO-8601 timestamp format
  "endDateTime": long, // In epoch sec, deprecated
  "endTime": "string", // ISO-8601 timestamp format
  "properties": { // A map of properties as in the get-entity API response
    // property name as key,
    // value is of type PropertyRequest
    "string": "PropertyRequest"
  }
}
```

```

},

"workspaceId": "string",
"selectedProperties": List:"string",
"propertyFilters": List:PropertyFilter,
"entityId": "string",
"componentName": "string",
"componentTypeId": "string",
"interpolation": InterpolationParameters,
"nextToken": "string",
"maxResults": int,
"orderByTime": "string"
}

```

## DataReader 響應接口

```

{
  "propertyValues": [
    {
      "entityPropertyReference": EntityPropertyReference, // The same
as EntityPropertyReference
      "values": [
        {
          "timestamp": long, // Epoch sec, deprecated
          "time": "string", // ISO-8601 timestamp format
          "value": DataValue // The same as DataValue
        }
      ]
    }
  ],
  "nextToken": "string"
}

```

## AttributePropertyValueReaderByEntity

AttributePropertyValueReaderByEntity 是可用來擷取單一實體中靜態屬性值的資料平面連接器。

如需有關此連接器的內容類型、語法和格式的資訊，請參閱 [GetPropertyValue](#) API 動作。

## AttributePropertyValueReaderByEntity 請求接口

```

{
  "properties": {

```

```

// property name as key,
// value is of type PropertyResponse
"string": "PropertyResponse"
}

"workspaceId": "string",
"entityId": "string",
"componentName": "string",
"selectedProperties": List:"string",
}

```

## AttributePropertyValueReaderByEntity 響應接口

```

{
  "propertyValues": {
    "string": { // property name as key
      "propertyReference": EntityPropertyReference, // The same
as EntityPropertyReference
      "propertyValue": DataValue // The same as DataValue
    }
  }
}

```

## DataWriter

DataWriter 是一種資料平面連接器，可用來將時間序列資料點寫回單一元件中性質的基礎資料倉庫。

如需有關此連接器的內容類型、語法和格式的資訊，請參閱 [BatchPutPropertyValuesAPI](#) 動作。

## DataWriter 請求接口

```

{
  "workspaceId": "string",
  "properties": {
    // entity id as key
    "String": {
      // property name as key,
      // value is of type PropertyResponse
      "string": PropertyResponse
    }
  },
  "entries": [
    {
      "entryId": "string",

```



```

    "entityPropertyReference": EntityPropertyReference, // The same
as EntityPropertyReference
    "propertyValues": [
      {
        "timestamp": long, // Epoch sec, deprecated
        "time": "string", // ISO-8601 timestamp format
        "value": DataValue // The same as DataValue
      }
    ]
  }
]
}

```

## DataWriter 響應接口

```

{
  "errorEntries": [
    {
      "errors": List:BatchPutPropertyError // The value is a list of
type BatchPutPropertyError
    }
  ]
}

```

## 範例

下列 JSON 範例是多個連接器的回應和要求語法範例。

- SchemaInitializer:

下列範例顯示元件類型生命週期中的結構描述初始設定式。

要求:

```

{
  "workspaceId": "myWorkspace",
  "properties": {
    "modelId": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": true,
        "isFinal": true,

```

```

        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": true,
        "isStoredExternally": false,
        "isTimeSeries": false,
        "defaultValue": {
            "stringValue": "myModelId"
        }
    },
    "value": {
        "stringValue": "myModelId"
    }
},
"tableName": {
    "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": false,
        "isFinal": false,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": false,
        "isStoredExternally": false,
        "isTimeSeries": false,
        "defaultValue": {
            "stringValue": "myTableName"
        }
    },
    "value": {
        "stringValue": "myTableName"
    }
}
}
}
}

```

回應：

```

{
  "properties": {
    "myProperty1": {
      "definition": {
        "dataType": {
          "type": "DOUBLE",
          "unitOfMeasure": "%"
        }
      }
    }
  }
}

```

```

    },
    "configuration": {
      "myProperty1Id": "idValue"
    },
    "isTimeSeries": true
  }
},
"myProperty2": {
  "definition": {
    "dataType": { "type": "STRING" },
    "isTimeSeries": false,
    "defaultValue": {
      "stringValue": "property2Value"
    }
  }
}
}
}
}
}

```

- 實體生命週期中的模式初始化程式

要求:

```

{
  "workspaceId": "myWorkspace",
  "entityId": "myEntity",
  "componentName": "myComponent",
  "properties": {
    "assetId": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": true,
        "isFinal": true,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": true,
        "isStoredExternally": false,
        "isTimeSeries": false
      },
      "value": {
        "stringValue": "myAssetId"
      }
    },
    "tableName": {

```

```
    "definition": {
      "dataType": { "type": "STRING" },
      "isExternalId": false,
      "isFinal": false,
      "isImported": false,
      "isInherited": false,
      "isRequiredInEntity": false,
      "isStoredExternally": false,
      "isTimeSeries": false
    },
    "value": {
      "stringValue": "myTableName"
    }
  }
}
```

回應：

```
{
  "properties": {
    "myProperty1": {
      "definition": {
        "dataType": {
          "type": "DOUBLE",
          "unitOfMeasure": "%"
        },
        "configuration": {
          "myProperty1Id": "idValue"
        },
        "isTimeSeries": true
      }
    },
    "myProperty2": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isTimeSeries": false
      },
      "value": {
        "stringValue": "property2Value"
      }
    }
  }
}
```

```
}
```

- `DataReaderByEntity` 以及 `DataReader` :

要求:

```
{
  "workspaceId": "myWorkspace",
  "entityId": "myEntity",
  "componentName": "myComponent",
  "selectedProperties": [
    "Temperature",
    "Pressure"
  ],
  "startTime": "2022-04-07T04:04:42Z",
  "endTime": "2022-04-07T04:04:45Z",
  "maxResults": 4,
  "orderByTime": "ASCENDING",
  "properties": {
    "assetId": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": true,
        "isFinal": true,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": true,
        "isStoredExternally": false,
        "isTimeSeries": false
      },
      "value": {
        "stringValue": "myAssetId"
      }
    },
    "Temperature": {
      "definition": {
        "configuration": {
          "temperatureId": "xyz123"
        },
        "dataType": {
          "type": "DOUBLE",
          "unitOfMeasure": "DEGC"
        }
      },
      "isExternalId": false,

```

```

        "isFinal": false,
        "isImported": true,
        "isInherited": false,
        "isRequiredInEntity": false,
        "isStoredExternally": false,
        "isTimeSeries": true
    }
},
"Pressure": {
    "definition": {
        "configuration": {
            "pressureId": "xyz456"
        },
        "dataType": {
            "type": "DOUBLE",
            "unitOfMeasure": "MPA"
        },
        "isExternalId": false,
        "isFinal": false,
        "isImported": true,
        "isInherited": false,
        "isRequiredInEntity": false,
        "isStoredExternally": false,
        "isTimeSeries": true
    }
}
}
}
}

```

回應：

```

{
  "propertyValues": [
    {
      "entityPropertyReference": {
        "entityId": "myEntity",
        "componentName": "myComponent",
        "propertyName": "Temperature"
      },
      "values": [
        {
          "time": "2022-04-07T04:04:42Z",
          "value": {

```

```
        "doubleValue": 588.168
      }
    },
    {
      "time": "2022-04-07T04:04:43Z",
      "value": {
        "doubleValue": 592.4224
      }
    }
  ]
},
"nextToken": "qwertyuiop"
}
```

- `AttributePropertyValueReaderByEntity`:

要求:

```
{
  "workspaceId": "myWorkspace",
  "entityId": "myEntity",
  "componentName": "myComponent",
  "selectedProperties": [
    "manufacturer",
  ],
  "properties": {
    "assetId": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": true,
        "isFinal": true,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": true,
        "isStoredExternally": false,
        "isTimeSeries": false
      },
      "value": {
        "stringValue": "myAssetId"
      }
    },
    "manufacturer": {
      "definition": {
```

```

    "dataType": { "type": "STRING" },
    "configuration": {
      "manufacturerPropId": "M001"
    },
    "isExternalId": false,
    "isFinal": false,
    "isImported": false,
    "isInherited": false,
    "isRequiredInEntity": false,
    "isStoredExternally": true,
    "isTimeSeries": false
  }
}
}
}

```

回應：

```

{
  "propertyValues": {
    "manufacturer": {
      "propertyReference": {
        "propertyName": "manufacturer",
        "entityId": "myEntity",
        "componentName": "myComponent"
      },
      "propertyValue": {
        "stringValue": "Amazon"
      }
    }
  }
}

```

- **DataWriter:**

要求:

```

{
  "workspaceId": "myWorkspaceId",
  "properties": {
    "myEntity": {
      "Temperature": {

```



```
    "definition": {
      "configuration": {
        "temperatureId": "xyz123"
      },
      "dataType": {
        "type": "DOUBLE",
        "unitOfMeasure": "DEGC"
      },
      "isExternalId": false,
      "isFinal": false,
      "isImported": true,
      "isInherited": false,
      "isRequiredInEntity": false,
      "isStoredExternally": false,
      "isTimeSeries": true
    }
  }
},
"entries": [
  {
    "entryId": "myEntity",
    "entityPropertyReference": {
      "entityId": "myEntity",
      "componentName": "myComponent",
      "propertyName": "Temperature"
    },
    "propertyValues": [
      {
        "timestamp": 1626201120,
        "value": {
          "doubleValue": 95.6958
        }
      },
      {
        "timestamp": 1626201132,
        "value": {
          "doubleValue": 80.6959
        }
      }
    ]
  }
]
```

```
}
```

回應：

```
{
  "errorEntries": [
    {
      "errors": [
        {
          "errorCode": "409",
          "errorMessage": "Conflict value at same timestamp",
          "entry": {
            "entryId": "myEntity",
            "entityPropertyReference": {
              "entityId": "myEntity",
              "componentName": "myComponent",
              "propertyName": "Temperature"
            },
            "propertyValues": [
              {
                "time": "2022-04-07T04:04:42Z",
                "value": {
                  "doubleValue": 95.6958
                }
              }
            ]
          }
        }
      ]
    }
  ]
}
```

## AWS IoT TwinMaker Athena 表格資料連接

透過 Athena 表格式資料連接器，您可以在中存取和使用您的 Athena 資料存放區 AWS IoT TwinMaker。您可以使用 Athena 資料建立數位雙胞胎，而不需要花費大量的資料移轉工作。您可以使用預先建立的連接器，也可以建立自訂的 Athena 連接器來存取來自 Athena 資料來源的資料。

### AWS IoT TwinMaker Athena 資料連接器的

使用 Athena 表格式資料連接器之前，請先完成下列先決條件：

- 建立受管的 Athena 表格及其相關聯的 Amazon S3 資源。如需使用 Athena 的相關資訊，請參閱 [Athena 文件](#)。
- 建立 AWS IoT TwinMaker 工作區。您可以在 [AWS IoT TwinMaker 主控台](#) 中建立工作區。
- 使用 Athena 許可更新您的工作區 IAM 角色。如需詳細資訊，請參閱 [修改您的工作區 IAM 角色以使用 Athena 資料連接器](#)。
- 熟悉 AWS IoT TwinMaker 實體元件系統以及如何建立實體。如需詳細資訊，請參閱 [建立您的第一個實體](#)。
- 熟悉 AWS IoT TwinMaker 的資料連接器。如需詳細資訊，請參閱 [AWS IoT TwinMaker 資料連接器](#)。

## 使用 Athena 資料連接器

若要使用 Athena 資料連接器，您必須使用 Athena 連接器做為元件類型來建立元件。然後將零組件附加到場景中的圖元，以便在中使用 AWS IoT TwinMaker。

### 使用 Athena 資料連接器建立元件類型

使用此程序來建立具有 Athena 表格式資料連接器的 AWS IoT TwinMaker 元件類型：

1. 導覽至 [AWS IoT TwinMaker 主控台](#)。
2. 開啟現有工作區或 [建立新工作區](#)。
3. 從左側導覽功能表中選擇 [元件類型]，然後選取 [建立元件類型] 以開啟元件類型建立頁面。
4. 在 [建立元件類型] 頁面上，在 ID 欄位中填入符合您使用案例的 ID。

**Component type information**

ID

com.test.athena.connector.example

Description

Example athena connector child component type

Must be less than 2048 characters

Base Type

Choose a pre-defined Component Type or create your own

com.amazon.athena.connector ▼

5. 選擇「基礎」類型。從下拉式清單中，選取標示為 Athena 表格式資料連接器。
6. 針對下列欄位選擇 Athena 資源，以設定元件類型的資料來源：
  - 選擇 Athena 資料來源。
  - 選擇一個 Athena 資料庫。
  - 選擇表格名稱。
  - 選擇 Athena 工作群組。
7. 選取要用作資料來源的 Athena 資源後，請從表格中選擇要包含的欄。
8. 選取「外部 ID」欄名稱。從上一個步驟中選取要做為外部 ID 欄的欄。外部 ID 是用來代表 Athena 資產並將其對應至 AWS IoT TwinMaker 實體的 ID。

## Athena Data Connector

### Athena datasource

Select an Athena datasource

AwsDataCatalog

### Athena Database

tabular\_test\_database

### Table Name

tabular\_test\_data\_service\_record

### Column Names

Select columns to include

<input checked="" type="checkbox"/>	Table name	Data type
<input checked="" type="checkbox"/>	recordid	bigint
<input type="checkbox"/>	assetid	string
<input checked="" type="checkbox"/>	description	string
<input checked="" type="checkbox"/>	dateperformed	string
<input checked="" type="checkbox"/>	performedby	string
<input checked="" type="checkbox"/>	datevalidated	string
<input checked="" type="checkbox"/>	validatedby	string
<input checked="" type="checkbox"/>	comments	string
<input checked="" type="checkbox"/>	nextservicedate	string
<input checked="" type="checkbox"/>	servicerecordurl	string

### External ID Column

assetid

### Athena workgroup

Select an Athena workgroup

TestWorkgroup

9. (選擇性) 將 AWS 標籤新增至這些資源，以便您可以將它們分組和組織。
10. 選擇「建立元件類型」以完成建立元件類型。

使用 Athena 資料連接器類型建立元件，並將其附加至實體

使用此程序來建立具有 Athena 表格式資料連接器的 AWS IoT TwinMaker 元件，並將其附加至實體：

**Note**

您必須擁有使用 Athena 表格式資料連接器做為資料來源的現有元件類型，才能完成此程序。在開始本逐步解說之前，請參閱先前的程序使用 Athena 資料連接器建立元件類型。

1. 導覽至 [AWS IoT TwinMaker 主控台](#)。
2. 開啟現有工作區或[建立新工作區](#)。
3. 從左側導覽功能表中選擇實體，然後選取要新增元件或建立新實體的實體。
4. [建立新實體](#)。
5. 接下來，選擇添加組件。」中，在「元件名稱」欄位中填入與您的使用案例相符的名稱。
6. 從元件類型下拉式功能表中，選取您在上一個程序中建立的元件類型 ID。
7. 輸入元件資訊、元件名稱，然後選取先前 ComponentType 建立的子項。這是 ComponentType 您使用 Athena 資料連接器建立的。
8. 在「性質」區段中，輸入元件的 athenaComponentExternalID。

Property	Data type	isTimeSeries	Storage	isRequired	Value
athenaComponentExt	String	False	Internal	True	A0001

Add another property

9. 選擇「新增元件」以完成建立元件。

您現在已成功建立具有 Athena 資料連接器做為元件類型的元件，並將其附加至實體。

## 使用 Athena 表格式資料連接器 JSON 參考

下列範例是 Athena 表格式資料連接器的完整 JSON 參考。使用此資源作為建立自訂資料連接器和元件類型的資源。

```
{
  "componentTypeId": "com.amazon.athena.connector",
  "description": "Athena connector for syncing tabular data",
  "workspaceId": "AmazonOwnedTypesWorkspace",
  "propertyGroups": {
    "tabularPropertyGroup": {
      "groupType": "TABULAR",
      "propertyNames": []
    }
  },
  "propertyDefinitions": {
    "athenaDataSource": {
      "dataType": { "type": "STRING" },
      "isRequiredInEntity": true
    },
    "athenaDatabase": {
      "dataType": { "type": "STRING" },
      "isRequiredInEntity": true
    },
    "athenaTable": {
      "dataType": { "type": "STRING" },
      "isRequiredInEntity": true
    },
    "athenaWorkgroup": {
      "dataType": { "type": "STRING" },
      "isRequiredInEntity": true
    },
    "athenaExternalIdColumnName": {
      "dataType": { "type": "STRING" },
      "isRequiredInEntity": true,
      "isExternalId": false
    },
    "athenaComponentExternalId": {
      "dataType": { "type": "STRING" },
      "isStoredExternally": false,
      "isRequiredInEntity": true,
      "isExternalId": true
    }
  }
}
```

```
    },  
    "functions": {  
      "tabularDataReaderByEntity": {  
        "implementedBy": {  
          "isNative": true  
        }  
      }  
    }  
  }  
}
```

## 使用 Athena 資料連接器

您可以在 Grafana 中顯示使用 Athena 表格的實體。如需詳細資訊，請參閱 [AWS IoT TwinMaker Grafana 儀表板整合](#)。

如需建立和使用 [Athena 表格儲存資料的相關資訊](#)，請閱讀 [Athena 說明文件](#)。

### 疑難排解 Athena 資料連接

本主題說明您在設定 Athena 資料連接器時可能會遇到的常見問題。

Athena 工作組位置：

建立 Athena 連接器元件類型時，Athena 工作群組必須設定輸出位置。請參閱 [工作群組的運作方式](#)。

缺少 IAM 角色許可：

在 AWS IoT TwinMaker 建立元件類型、將 Ca 元件新增至實體或執行 API 時，工作區角色可能缺少 Athena API 存取權限。GetProperty 若要更新 IAM 許可，請參閱 [建立和管理的服務角色 AWS IoT TwinMaker](#)。

## 在 Grafana 中視覺化 Athena 表格式資料

Grafana 外掛程式也可用來視覺化您在 Grafana 儀表板面板上的表格資料 AWS IoT TwinMaker，其中包含其他功能，例如根據選取的屬性進行排序和篩選，而無需對 Athena 進行 API 呼叫或與 Athena 互動。本主題說明如何設定 Grafana 以視覺化的方式呈現 Athena 表格式資料。

### 必要條件

在設定 Grafana 面板以視覺化 Athena 表格資料之前，請先檢閱下列先決條件：



- 你已經設置了一個 Grafana 環境。如需詳細資訊，請參閱 [AWS IoT TwinMaker Grafana 整合](#)。
- 您可以設定 Grafana 資料來源。如需詳細資訊，請參閱 [AWS IoT TwinMaker](#)。
- 您熟悉建立新管控面板並新增面板。

## 在 Grafana 中視覺化 Athena 表格式資料

此程序說明如何設定 Grafana 面板，以視覺化方式呈現 Athena 表格資料。

1. 開啟您的 AWS IoT TwinMaker Grafana 儀表板。
2. 在面板設定中選取「表格」面板。
3. 在查詢配置中選擇您的數據源。
4. 選取「取得性質值」查詢。
5. 選擇實體。
6. 選取具有延伸 Athena 基礎元件類型的元件類型的元件。
7. 選取 Athena 餐桌的屬性群組。
8. 從內容群組中選取任意數目的內容。
9. 透過篩選器和性質順序清單設定表格式條件。使用以下選項：
  - 篩選：定義性質值的表示式，以篩選資料。
  - OrderBy：指定資料是否應以遞增或遞減順序傳回屬性。

Panel Title						
crit {componentName=}	description {component	equipment_type {compo	status {componentNam	total {componentName=	won {componentName=	
5	Shutdown valve inspec...	VALVE	COMPLETED	90563	128355	
5	Damaged cable on SDV	VALVE	COMPLETED	90041	128461	
5	BYTN-04-TV-02385 do...	VALVE	COMPLETED	85611	128361	
5	Shutdown vlv inspection	VALVE	COMPLETED	73797	128531	
5	RYTN-02-XV-06517 do	VALVE	COMPLETED	71326	128458	

Query 1 Transform 0

Query type: Get Property value

Entity: TabularEntity1

Component Name: TabularComponent

Property Group: tabularPropertyGroup (TABULAR)

Selected Properties: won (INTEGER) × status (STRING) × total (INTEGER) × crit (INTEGER) × description (STRING) × equipment\_type (STRING) ×

Filter: crit (INTEGER) = 5

OrderBy: total (INTEGER) DESC

## 開發 AWS IoT TwinMaker 時間序列資料連接器

本節說明如何在程序中開發時間 step-by-step 序列資料連接器。此外，我們還提供以整個 Cookie Factory 範例為基礎的時間序列資料連接器範例，其中包括 3D 模型、實體、元件、警示和連接器。Cookie 工廠範例來源可在 [AWS IoT TwinMaker 範例 GitHub 儲存庫](#) 中取得。

### 主題

- [AWS IoT TwinMaker 時間序列資料連接器必要](#)
- [時間序列資料連接器背景](#)
- [開發時間序列資料連接器](#)
- [改善您的資料連接器](#)
- [測試您的連接器](#)
- [安全](#)

- [建立 AWS IoT TwinMaker 資源](#)
- [下一步是什麼](#)
- [AWS IoT TwinMaker 餅乾工廠示例時間序列連接器](#)

## AWS IoT TwinMaker 時間序列資料連接器必要

在開發時間序列資料連接器之前，建議您先完成下列工作：

- 建立[AWS IoT TwinMaker 工作區](#)。
- 建立[AWS IoT TwinMaker 元件類型](#)。
- 建立[AWS IoT TwinMaker 實體](#)。
- (選擇性) 讀取[使用並建立元件類型](#)。
- (選擇性) 讀取[AWS IoT TwinMaker 資料連接器介面](#)，以大致瞭解資 AWS IoT TwinMaker 料連接器。

### Note

如需完全實作的連接器範例，請參閱我們的 [Cookie 工廠範例實作](#)。

## 時間序列資料連接器背景

想像一下，您正在與一家擁有一套餅乾混合器和水箱的工廠合作。您想要建立這些實體的 AWS IoT TwinMaker 數位雙胞胎，以便您可以透過檢查各種時間序列指標來監視其運作狀態。

您已經設定了現場感應器，而且您已經將測量資料串流到 Timestream 資料庫中。您希望能夠以最小的額外負荷檢視和組織測量資料。AWS IoT TwinMaker 您可以使用時間序列資料連接器來完成此工作。下圖顯示範例遙測表格，透過使用時間序列連接器填入。

Rows returned (1000+)

Results are paginated. Scroll through the result pages to see more query results.

Filter

TelemetryAssetId	TelemetryAssetType	measure_name	time	measure_value:varchar	measure_value:double
Mixer_22_680b5b8e-1afe-4a77-87ab-834f5e5ba01e	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.4233207702637
Mixer_22_680b5b8e-1afe-4a77-87ab-834f5e5ba01e	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.9421195983887
Mixer_24_7f0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_25_cf42effc-ba19-48ba-bbc3-d21d2508ce31	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.8453979492188
Mixer_20_0568f25f-116c-429c-a974-5ceec065a6ac	Mixer	Temperature	2022-04-19 00:28:00.241000000	-	99.1292877197266
Mixer_24_7f0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.4532585144043
Mixer_15_0bb566cd-d6f3-4804-9fe1-7d2abca82d0	Mixer	RPM	2022-04-19 00:28:00.241000000	-	58.397144317627
Mixer_2_d8e76844-e739-4845-a748-a83983279376	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.206958770752
Mixer_6_b66db3d3-c144-47b5-afb9-3a0150c53456	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.206958770752

此螢幕擷取畫面中使用的資料集和 Timestream 表格可在 [AWS IoT TwinMaker 範例 GitHub 儲存庫](#) 中取得。另請參閱實施的 [cookie 工廠示例連接器](#)，該連接器會產生上述螢幕截圖中顯示的結果。

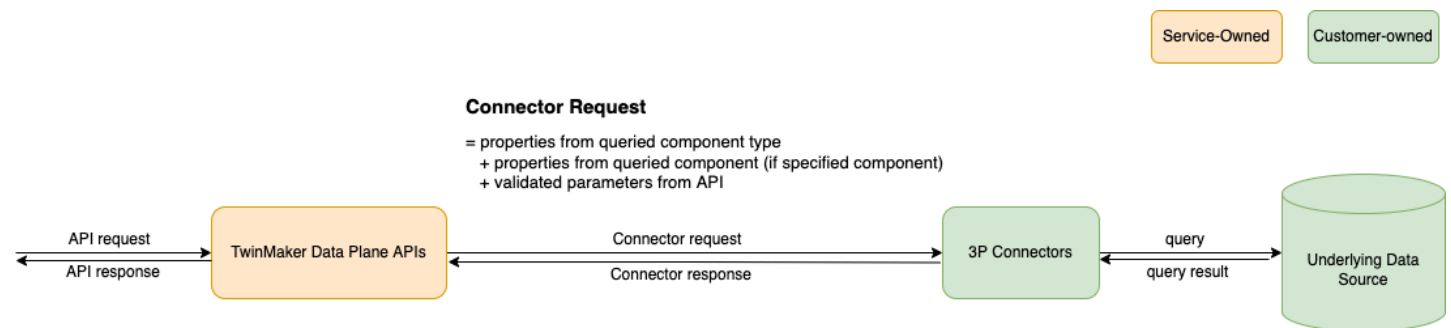
## 時間序列資料連接器資料流程

對於資料平面查詢，請從元件和元件類型定義中 AWS IoT TwinMaker 擷取元件和元件類型的對應屬性。AWS IoT TwinMaker 將屬性與查詢中的任何 API 查詢參數一起轉發給 AWS Lambda 函數。

AWS IoT TwinMaker 使用 Lambda 函數存取和解析來自資料來源的查詢，並傳回這些查詢的結果。Lambda 函數使用資料平面中的元件和元件類型屬性來解決初始請求。

Lambda 查詢的結果會對應至 API 回應並傳回給您。

AWS IoT TwinMaker 定義資料連接器介面，並使用該介面與 Lambda 函數互動。使用資料連接器，您可以從 AWS IoT TwinMaker API 查詢資料來源，而無需進行任何資料移轉工作。下列影像概述了前幾段所述的基本資料流程。



## 開發時間序列資料連接器

下列程序概述逐步建立至功能性時間序列資料連接器的開發模型。基本步驟如下：

### 1. 建立有效的基本元件類型

在元件類型中，您可以定義跨元件共用的通用性質。若要深入瞭解如何定義元件類型，請參閱[使用和建立元件類型](#)。

AWS IoT TwinMaker 使用[實體元件塑型陣列](#)，以便將每個元件貼附至一個圖元。我們建議您將每個實體項目建模為實體，並使用各自的元件類型來建模不同的資料來源。

下列範例會顯示具有一個屬性的 Timestream 範本元件類型：

```
{
  "componentTypeId": "com.example.timestream-telemetry",
  "workspaceId": "MyWorkspace",
  "functions": {
    "dataReader": {
      "implementedBy": {
        "lambda": {
          "arn": "lambdaArn"
        }
      }
    }
  },
  "propertyDefinitions": {
    "telemetryType": {
      "dataType": { "type": "STRING" },
      "isExternalId": false,
      "isStoredExternally": false,
      "isTimeSeries": false,
      "isRequiredInEntity": true
    },
    "telemetryId": {
      "dataType": { "type": "STRING" },
      "isExternalId": true,
      "isStoredExternally": false,
      "isTimeSeries": false,
      "isRequiredInEntity": true
    },
    "Temperature": {
      "dataType": { "type": "DOUBLE" },
      "isExternalId": false,
      "isTimeSeries": true,
      "isStoredExternally": true,
      "isRequiredInEntity": false
    }
  }
}
```

```
}
```

組件類型的關鍵元素如下：

- 此 `telemetryId` 屬性會識別對應資料來源中實體項目的唯一索引鍵。資料連接器使用此性質做為篩選條件，以僅查詢與指定項目相關聯的值。此外，如果您在資料平面 API 回應中包含 `telemetryId` 屬性值，則用戶端會取得 ID，並在必要時執行反向查詢。
- 此 `lambdaArn` 欄位會識別元件類型參與的 Lambda 函數。
- 此 `isRequiredInEntity` 旗標會強制建立 ID。此標誌是必需的，以便在創建組件時，也實例化項目的 ID。
- 會以外部 ID 的形式新增至元件類型，以便在「時間串流」表格中識別項目。TelemetryId

## 2. 使用元件類型建立元件

若要使用您建立的元件類型，您必須建立一個元件，並將其附加至您要從中擷取資料的圖元。下列步驟詳細說明建立該元件的程序：

- a. 導覽至 [AWS IoT TwinMaker 主控台](#)。
- b. 選取並開啟您在其中建立元件類型的相同工作區。
- c. 導覽至實體頁面。
- d. 建立新實體或從表格中選取現有實體。
- e. 選取要使用的實體之後，請選擇新增元件以開啟 [新增元件] 頁面。
- f. 為元件命名，並為「類型」選擇您在 1 中使用範本建立的元件類型。建立有效的基本元件類型。

## 3. 讓您的元件類型呼叫 Lambda 連接器

Lambda 連接器需要存取資料來源，並根據輸入產生查詢陳述式，並將其轉寄至資料來源。下列範例會顯示執行此動作的 JSON 要求範本。

```
{
  "workspaceId": "MyWorkspace",
  "entityId": "MyEntity",
  "componentName": "TelemetryData",
  "selectedProperties": ["Temperature"],
  "startTime": "2022-08-25T00:00:00Z",
  "endTime": "2022-08-25T00:00:05Z",
  "maxResults": 3,
  "orderByTime": "ASCENDING",
```

```
"properties": {
  "telemetryType": {
    "definition": {
      "dataType": { "type": "STRING" },
      "isExternalId": false,
      "isFinal": false,
      "isImported": false,
      "isInherited": false,
      "isRequiredInEntity": false,
      "isStoredExternally": false,
      "isTimeSeries": false
    },
    "value": {
      "stringValue": "Mixer"
    }
  },
  "telemetryId": {
    "definition": {
      "dataType": { "type": "STRING" },
      "isExternalId": true,
      "isFinal": true,
      "isImported": false,
      "isInherited": false,
      "isRequiredInEntity": true,
      "isStoredExternally": false,
      "isTimeSeries": false
    },
    "value": {
      "stringValue": "item_A001"
    }
  },
  "Temperature": {
    "definition": {
      "dataType": { "type": "DOUBLE", },
      "isExternalId": false,
      "isFinal": false,
      "isImported": true,
      "isInherited": false,
      "isRequiredInEntity": false,
      "isStoredExternally": false,
      "isTimeSeries": true
    }
  }
}
```

```
}
```

請求的關鍵要素：

- 這selectedProperties是一份清單，其中包含您要進行時間流測量的屬性。
- startDateTime、startTimeEndTime、和endTime欄位指定要求的時間範圍。這決定了返回的測量的樣本範圍。
- 這entityId是您要從中查詢資料的實體名稱。
- 這componentName是您要從中查詢資料之元件的名稱。
- 使用此orderByTime欄位來組織結果的顯示順序。

在前面的示例請求中，我們希望在給定項目的給定時間段內獲得所選屬性的一系列樣本，並具有所選時間順序。響應語句可以歸納為以下內容：

```
{
  "propertyValues": [
    {
      "entityPropertyReference": {
        "entityId": "MyEntity",
        "componentName": "TelemetryData",
        "propertyName": "Temperature"
      },
      "values": [
        {
          "time": "2022-08-25T00:00:00Z",
          "value": {
            "doubleValue": 588.168
          }
        },
        {
          "time": "2022-08-25T00:00:01Z",
          "value": {
            "doubleValue": 592.4224
          }
        },
        {
          "time": "2022-08-25T00:00:02Z",
          "value": {
            "doubleValue": 594.9383
          }
        }
      ]
    }
  ]
}
```



```

    }
  ]
}
],
"nextToken": "...
}

```

#### 4. 將元件類型更新為具有兩個屬性

下列 JSON 範本顯示具有兩個屬性的有效元件類型：

```

{
  "componentTypeId": "com.example.timestream-telemetry",
  "workspaceId": "MyWorkspace",
  "functions": {
    "dataReader": {
      "implementedBy": {
        "lambda": {
          "arn": "lambdaArn"
        }
      }
    }
  },
  "propertyDefinitions": {
    "telemetryType": {
      "dataType": { "type": "STRING" },
      "isExternalId": false,
      "isStoredExternally": false,
      "isTimeSeries": false,
      "isRequiredInEntity": true
    },
    "telemetryId": {
      "dataType": { "type": "STRING" },
      "isExternalId": true,
      "isStoredExternally": false,
      "isTimeSeries": false,
      "isRequiredInEntity": true
    },
    "Temperature": {
      "dataType": { "type": "DOUBLE" },
      "isExternalId": false,
      "isTimeSeries": true,
      "isStoredExternally": true,
      "isRequiredInEntity": false
    }
  }
}

```

```

    },
    "RPM": {
      "dataType": { "type": "DOUBLE" },
      "isExternalId": false,
      "isTimeSeries": true,
      "isStoredExternally": true,
      "isRequiredInEntity": false
    }
  }
}

```

## 5. 更新 Lambda 連接器以處理第二個屬性

AWS IoT TwinMaker 資料平面 API 支援在單一要求中查詢多個屬性，並透過提供清單來追 AWS IoT TwinMaker 縱連接器的單一要求 `selectedProperties`。

下列 JSON 要求會顯示已修改的範本，該範本現在支援兩個屬性的要求。

```

{
  "workspaceId": "MyWorkspace",
  "entityId": "MyEntity",
  "componentName": "TelemetryData",
  "selectedProperties": ["Temperature", "RPM"],
  "startTime": "2022-08-25T00:00:00Z",
  "endTime": "2022-08-25T00:00:05Z",
  "maxResults": 3,
  "orderByTime": "ASCENDING",
  "properties": {
    "telemetryType": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": false,
        "isFinal": false,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": false,
        "isStoredExternally": false,
        "isTimeSeries": false
      },
      "value": {
        "stringValue": "Mixer"
      }
    }
  },
}

```

```
    "telemetryId": {
      "definition": {
        "dataType": { "type": "STRING" },
        "isExternalId": true,
        "isFinal": true,
        "isImported": false,
        "isInherited": false,
        "isRequiredInEntity": true,
        "isStoredExternally": false,
        "isTimeSeries": false
      },
      "value": {
        "stringValue": "item_A001"
      }
    },
    "Temperature": {
      "definition": {
        "dataType": { "type": "DOUBLE" },
        "isExternalId": false,
        "isFinal": false,
        "isImported": true,
        "isInherited": false,
        "isRequiredInEntity": false,
        "isStoredExternally": false,
        "isTimeSeries": true
      }
    },
    "RPM": {
      "definition": {
        "dataType": { "type": "DOUBLE" },
        "isExternalId": false,
        "isFinal": false,
        "isImported": true,
        "isInherited": false,
        "isRequiredInEntity": false,
        "isStoredExternally": false,
        "isTimeSeries": true
      }
    }
  }
}
```

同樣地，對應的回應也會更新，如下列範例所示：

```
{
  "propertyValues": [
    {
      "entityPropertyReference": {
        "entityId": "MyEntity",
        "componentName": "TelemetryData",
        "propertyName": "Temperature"
      },
      "values": [
        {
          "time": "2022-08-25T00:00:00Z",
          "value": {
            "doubleValue": 588.168
          }
        },
        {
          "time": "2022-08-25T00:00:01Z",
          "value": {
            "doubleValue": 592.4224
          }
        },
        {
          "time": "2022-08-25T00:00:02Z",
          "value": {
            "doubleValue": 594.9383
          }
        }
      ]
    },
    {
      "entityPropertyReference": {
        "entityId": "MyEntity",
        "componentName": "TelemetryData",
        "propertyName": "RPM"
      },
      "values": [
        {
          "time": "2022-08-25T00:00:00Z",
          "value": {
            "doubleValue": 59
          }
        },
        {

```

```
    "time": "2022-08-25T00:00:01Z",
    "value": {
      "doubleValue": 60
    }
  },
  {
    "time": "2022-08-25T00:00:02Z",
    "value": {
      "doubleValue": 60
    }
  }
]
},
"nextToken": "...
}
```

#### Note

就這種情況的分頁而言，請求中的頁面大小適用於所有屬性。這表示查詢中有五個屬性，頁面大小為 100，如果來源中有足夠的資料點，您應該預期會看到每個屬性 100 個資料點，總共有 500 個資料點。

如需實作範例，請參閱中的[雪花連接器範例](#) GitHub。

## 改善您的資料連接器

### 處理例外狀況

Lambda 連接器擲回例外狀況是安全的。在資料平面 API 呼叫中，AWS IoT TwinMaker 服務會等待 Lambda 函數傳回回應。如果連接器實作擲回例外狀況，則會將例外狀況類型 AWS IoT TwinMaker 轉譯為 ConnectorFailure，讓 API 用戶端知道連接器內部發生了問題。

### 處理分頁

在這個例子中，Timestream 提供了一個[實用程序功能](#)，可以幫助本地支持分頁。但是，對於其他一些查詢接口（例如 SQL），可能需要額外的努力來實現有效的分頁算法。有一個[雪花](#)連接器範例可以在 SQL 介面中處理分頁。

AWS IoT TwinMaker 透過連接器回應介面傳回新權杖時，該權杖會先加密，然後再傳回 API 用戶端。當權杖包含在另一個請求中時，請先將其 AWS IoT TwinMaker 解密，然後再將其轉寄至 Lambda 連接器。我們建議您避免在權杖中新增敏感資訊。

## 測試您的連接器

雖然您仍然可以在連接器連結至元件類型之後更新實作，但我們強烈建議您在與之整合之前先驗證 Lambda 連接器 AWS IoT TwinMaker。

有多種方法可以測試 Lambda 連接器：您可以在 Lambda 主控台或本機中測試 Lambda 連接器 AWS CDK。

如需有關測試 Lambda 函數的詳細資訊，請參閱[測試 Lambda 函數](#)和[本機測試 AWS CDK 應用程式](#)。

## 安全

如需有關 Timestream 安全性最佳做法的說明文件，請參閱[時間串流中的安全性](#)。

如需 SQL 注入預防範例的範例，請參閱 AWS IoT TwinMaker 範例 GitHub 存放庫中的下列[Python 指令碼](#)。

## 建立 AWS IoT TwinMaker 資源

實作 Lambda 函數後，您可以透過[AWS IoT TwinMaker 主控台](#)或 API 建立元件類型、實體和元件等 AWS IoT TwinMaker 資源。

### Note

如果您遵循 GitHub 範例中的設定指示，則會自動提供所有 AWS IoT TwinMaker 資源。您可以檢查範[AWS IoT TwinMaker GitHub 例](#)中的元件類型定義。一旦任何元件使用元件類型，就無法更新元件類型的性質定義和函數。

## 集成測試

我們建議您進行整合測試，AWS IoT TwinMaker 以驗證資料平面查詢的運作方式 end-to-end。您可以通過 [GetPropertyValueHistory](#) API 執行，也可以輕鬆地在[AWS IoT TwinMaker 控制台](#)中執行。

在 AWS IoT TwinMaker 控制台中，轉到組件詳細信息，然後在測試下，您將看到組件中的所有屬性列在那裡。主控台的 [測試] 區域可讓您測試時間序列屬性和 non-time-series 屬性。對於時間序列屬性，您也可以使用 [GetPropertyValueHistory](#) API 和 non-time-series 屬性使用 [GetPropertyValue](#) API。如果您的 Lambda 連接器支援多個內容查詢，您可以選擇多個屬性。

## 下一步是什麼

您現在可以設定 [AWS IoT TwinMaker Grafana 儀表板以視覺化指標](#)。您也可以瀏覽範例 [GitHub 存放庫中的其他資料連接器AWS IoT TwinMaker 範例](#)，以查看它們是否符合您的使用案例。

## AWS IoT TwinMaker餅乾工廠示例時間序列連接器

[Cookie 工廠 Lambda 函數的完整程式碼可在上取得](#) GitHub。雖然您仍然可以在連接器連結至元件類型之後更新實作，但我們強烈建議您在與之整合之前先驗證 Lambda 連接器 AWS IoT TwinMaker。您可以在 Lambda 主控台或本機中測試 Lambda 函數 AWS CDK。如需有關測試 Lambda 函數的詳細資訊，請參閱[測試 Lambda 函數](#)和[本機測試 AWS CDK 應用程式](#)。

### 餅乾工廠組件類型示例

在組件類型中，我們定義跨組件共享的共同屬性。對於 cookie 工廠示例，相同類型的物理組件共享相同的測量，因此我們可以在組件類型中定義測量結構描述。例如，混音器類型在下列範例中定義。

```
{
  "componentTypeId": "com.example.cookiefactory.mixer"
  "propertyDefinitions": {
    "RPM": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "Temperature": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    }
  }
}
```

例如，實體元件可能在 Timestream 資料庫中具有測量值、SQL 資料庫中的維護記錄，或警示系統中的警示資料。建立多個元件並將其與實體產生關聯，會將不同的資料來源連結至實體，並填入實體元件圖形。在此內容中，每個元件都需要一個telemetryId性質來識別對應資料來源中元件的唯一索引鍵。指定telemetryId屬性有兩個好處：該內容可用於資料連接器作為篩選條件，以僅查詢指定元件



的值，而且如果您在資料平面 API 回應中包含telemetryId屬性值，則用戶端會取得 ID，並在必要時執行反向查閱。

如果您將元件類型新增為外部 ID，它會在TimeStream表格中識別元件。TelemetryId

```
{
  "componentTypeId": "com.example.cookiefactory.mixer"
  "propertyDefinitions": {
    "telemetryId": {
      "dataType": { "type": "STRING" },
      "isTimeSeries": false,
      "isRequiredInEntity": true,
      "isExternalId": true,
      "isStoredExternally": false
    },
    "RPM": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "Temperature": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    }
  }
}
```

同樣地，我們也有的元件類型WaterTank，如下面的 JSON 範例所示。

```
{
  "componentTypeId": "com.example.cookiefactory.watertank",
  "propertyDefinitions": {
    "flowRate1": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    }
  }
}
```

```

    },
    "flowrate2": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "tankVolume1": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "tankVolume2": {
      "dataType": { "type": "DOUBLE" },
      "isTimeSeries": true,
      "isRequiredInEntity": false,
      "isExternalId": false,
      "isStoredExternally": true
    },
    "telemetryId": {
      "dataType": { "type": "STRING" },
      "isTimeSeries": false,
      "isRequiredInEntity": true,
      "isExternalId": true,
      "isStoredExternally": false
    }
  }
}
}

```

如果它的目的TelemetryType是查詢實體範圍中的屬性值，則是組件類型中的選用屬性。如需範例，請參閱[AWS IoT TwinMaker 範例 GitHub 存放庫](#)中已定義的元件類型。同一個資料表中也內嵌了警示類型，因此TelemetryType已定義，您可以擷取常用屬性 (如TelemetryId和) TelemetryType 至父元件類型，以供其他子類型共用。

## 範例 Lambda

Lambda 連接器需要存取資料來源，並根據輸入產生查詢陳述式，並將其轉寄至資料來源。傳送至 Lambda 的範例要求會顯示在下列 JSON 範例中。

```
{
```

```
'workspaceId': 'CookieFactory',
'selectedProperties': ['Temperature'],
'startDateTime': 1648796400,
'startTime': '2022-04-01T07:00:00.000Z',
'endDateTime': 1650610799,
'endTime': '2022-04-22T06:59:59.000Z',
'properties': {
  'telemetryId': {
    'definition': {
      'dataType': { 'type': 'STRING' },
      'isTimeSeries': False,
      'isRequiredInEntity': True,
      'isExternalId': True,
      'isStoredExternally': False,
      'isImported': False,
      'isFinal': False,
      'isInherited': True,
    },
    'value': {
      'stringValue': 'Mixer_22_680b5b8e-1afe-4a77-87ab-834fbe5ba01e'
    }
  }
  'Temperature': {
    'definition': {
      'dataType': { 'type': 'DOUBLE' },
      'isTimeSeries': True,
      'isRequiredInEntity': False,
      'isExternalId': False,
      'isStoredExternally': True,
      'isImported': False,
      'isFinal': False,
      'isInherited': False
    }
  }
  'RPM': {
    'definition': {
      'dataType': { 'type': 'DOUBLE' },
      'isTimeSeries': True,
      'isRequiredInEntity': False,
      'isExternalId': False,
      'isStoredExternally': True,
      'isImported': False,
      'isFinal': False,
      'isInherited': False
    }
  }
}
```

```
    }  
  },  
  'entityId': 'Mixer_22_d133c9d0-472c-48bb-8f14-54f3890bc0fe',  
  'componentName': 'MixerComponent',  
  'maxResults': 100,  
  'orderByTime': 'ASCENDING'  
}
```

Lambda 函數的目標是查詢特定實體的歷史測量資料。AWS IoT TwinMaker 提供元件-屬性對映，您應該為元件 ID 指定具現化的值。例如，若要處理元件類型層級查詢 (警示使用案例很常見) 並傳回工作區中所有元件的警示狀態，則屬性對映具有元件類型性質定義。

對於最簡單的情況，如前面的請求，我們希望在給定的時間窗口內對給定的組件進行一系列溫度樣本，以遞增的時間順序進行。查詢語句可以歸納為以下內容：

```
...  
SELECT measure_name, time, measure_value::double  
  FROM {database_name}.{table_name}  
 WHERE time < from_iso8601_timestamp('{request.start_time}')  
 AND time >= from_iso8601_timestamp('{request.end_time}')  
 AND TelemetryId = '{telemetry_id}'  
 AND measure_name = '{selected_property}'  
 ORDER BY time {request.orderByTime}  
...
```

# 建立和編輯 AWS IoT TwinMaker 場景

場景是您的數字雙胞胎的三維可視化。它們是您編輯數位雙胞胎的主要方式。瞭解如何在場景中新增鬧鐘、時間序列資料、色彩覆疊、標籤和視覺規則，以使數位雙胞胎視覺效果與實際使用案例保持一致。

本節涵蓋下列主題：

- [建立第一個場景之前](#)
- [上傳資源至資 AWS IoT TwinMaker 源庫](#)
- [建立您的場景](#)
- [將固定攝影機加入至圖元](#)
- [場景增強編輯](#)
- [編輯您的場景](#)
- [3D 瓷磚模型格式](#)
- [動態場景](#)

## 建立第一個場景之前

場景依靠資源來代表您的數位雙胞胎。這些資源由 3D 模型、資料或紋理檔案組成。資源的大小和複雜性、場景中的元素 (例如照明) 以及電腦硬體都會影響場 AWS IoT TwinMaker 景的效能。使用本主題中的資訊可減少延遲、載入時間，並改善場景的影格速率。

## 將資源匯入之前，先將資源最佳化 AWS IoT TwinMaker

您可以使 AWS IoT TwinMaker 用與數字雙胞胎實時互動。為了獲得最佳場景體驗，我們建議您將資源最佳化，以便在即時環境中使用。

您的 3D 模型可能會對效能產生重大影響。複雜的模型幾何圖形和網格可能會降低效能。例如，工業 CAD 模型具有較高的詳細等級。我們建議您先壓縮這些模型的網格，並減少它們的多邊形數量，然後再在場景中 AWS IoT TwinMaker 使用它們。如果您要為其建立新的 3D 模型 AWS IoT TwinMaker，則應該建立詳細等級並在所有模型中維護它。從不影響使用案例視覺效果或解釋的模型中移除詳細資料。

若要壓縮模型並縮小檔案大小，請使用開放原始碼網格壓縮工具，例如 [DRACO 3D 資料壓縮](#)。

未最佳化的紋理也會影響效能。如果紋理中不需要任何透明度，請考慮選擇 PEG 圖像格式而不是 PNG 格式。您可以使用開放原始碼紋理壓縮工具 (例如 [Basis 通用紋理壓縮](#)) 來壓縮紋理檔案。

## 效能的最佳做法 AWS IoT TwinMaker

若要取得最佳效能 AWS IoT TwinMaker，請注意下列限制和最佳作法。

- AWS IoT TwinMaker 場景彩現效能取決於硬體。效能會因不同的電腦硬體組態而異。
- 我們建議您在您的所有物件中的多邊形總計數低於 100 萬個 AWS IoT TwinMaker。
- 我們建議每個場景共有 200 個物件。將場景中的物件數量增加到 200 以上可能會降低場景影格速率。
- 我們建議場景中所有獨特 3D 資產的總大小不超過 100 MB。否則，您可能會遇到緩慢的加載時間或性能下降，具體取決於您的瀏覽器和硬件。
- 依預設，場景具有環境光源。您可以在場景中增加額外的光源，以使某些物件成為焦點，或在物件上投射陰影。我們建議每個場景使用一個光源。在需要的地方使用光源，並避免在場景中複製真實世界的光源。

## 進一步了解

使用這些資源進一步瞭解可用來改善場景效能的最佳化技巧。

- [如何將 OBJ 模型轉換和壓縮為 GLTF 以便與一起使用 AWS IoT TwinMaker](#)
- [針對網頁內容最佳化 3D 模型](#)
- [優化場景以獲得更好的 WebGL 性能](#)

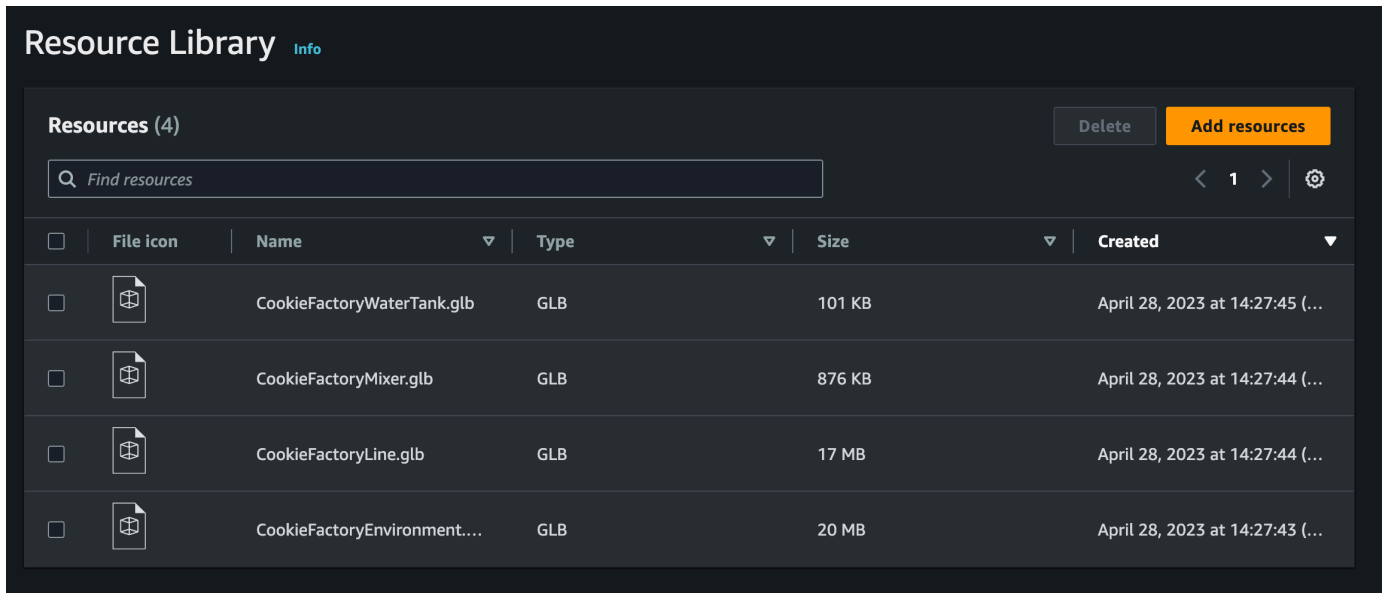
## 上傳資源至資 AWS IoT TwinMaker 源庫

您可以使用資源庫來控制和管理任何要放置在數位孿生應用程式場景中的資源。若要 AWS IoT TwinMaker 瞭解資源，請使用 [資源庫] 主控台頁面上傳資源。

## 使用主控台將檔案上傳至資源庫

請依照下列步驟使用 AWS IoT TwinMaker 主控台將檔案新增至「資源程式庫」。

1. 在左側導覽功能表的「工作區」下，選取「資源庫」。
2. 選取 [新增資源]，然後選擇您要上傳的檔案。



## 建立您的場景

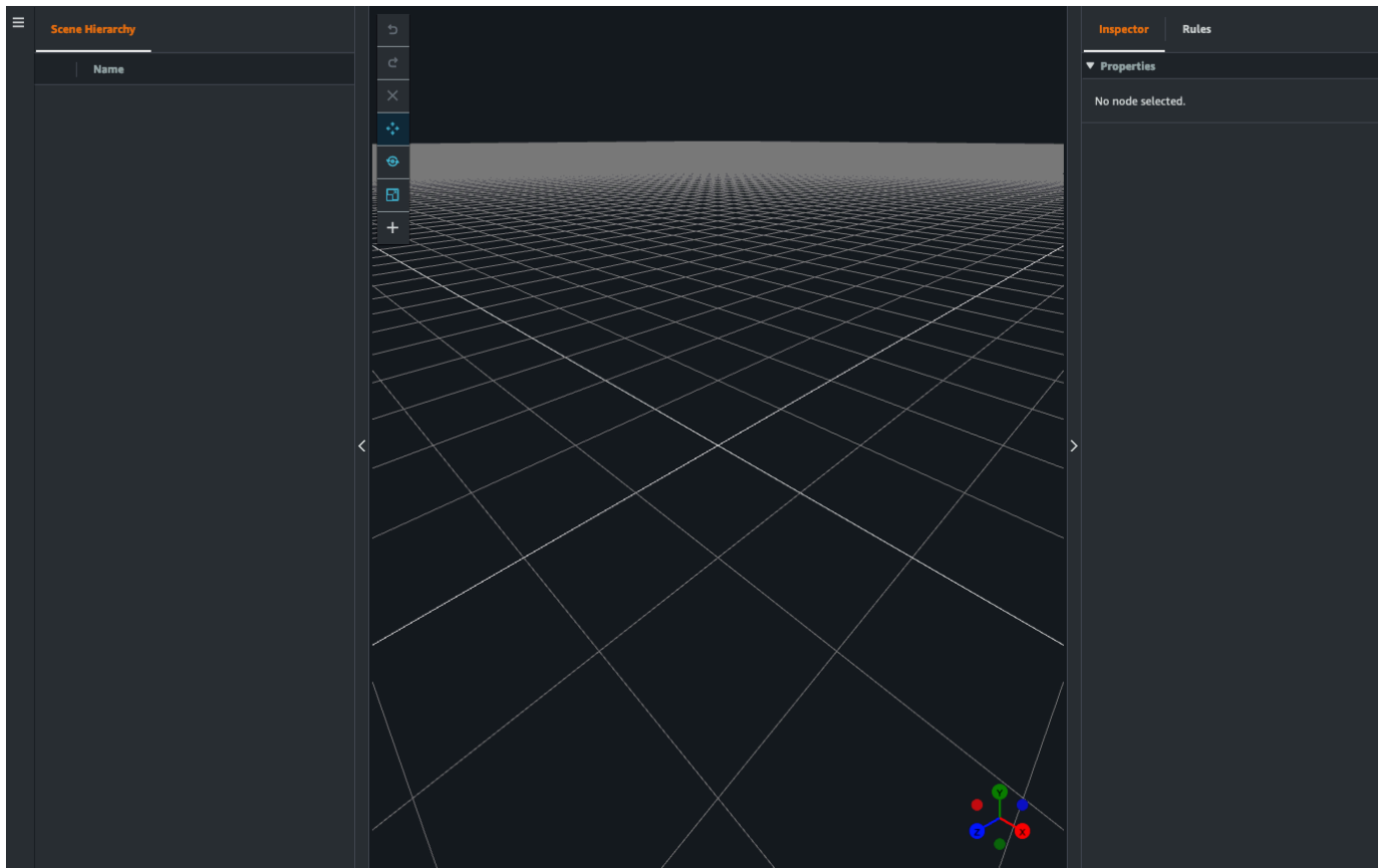
在本節中，您將設置一個場景，以便您可以編輯數字雙胞胎。您可以導入已上傳到[資源庫](#)的 3D 模型，然後添加小部件並將屬性數據綁定到對象以完成數字對象。場景物件可以包括整個建築或空間，或定位於其實體位置的個別設備。

### Note

建立場景之前，必須先建立工作區。

使用下列程序在中建立場景 AWS IoT TwinMaker。

1. 若要開啟場景窗格，請在工作區的左側導覽中選擇「場景」。
2. 請選擇 Create scene (建立場景)。新場景建立窗格隨即開啟。
3. 在場景建立窗格中，輸入新場景的名稱和描述。如果您有標準或分層組合定價方案，則可以選取場景類型。建議使用[動態場景](#)。
4. 當您準備好建立場景時，請選擇「建立場景」。新場景隨即開啟，並準備好讓您使用它。

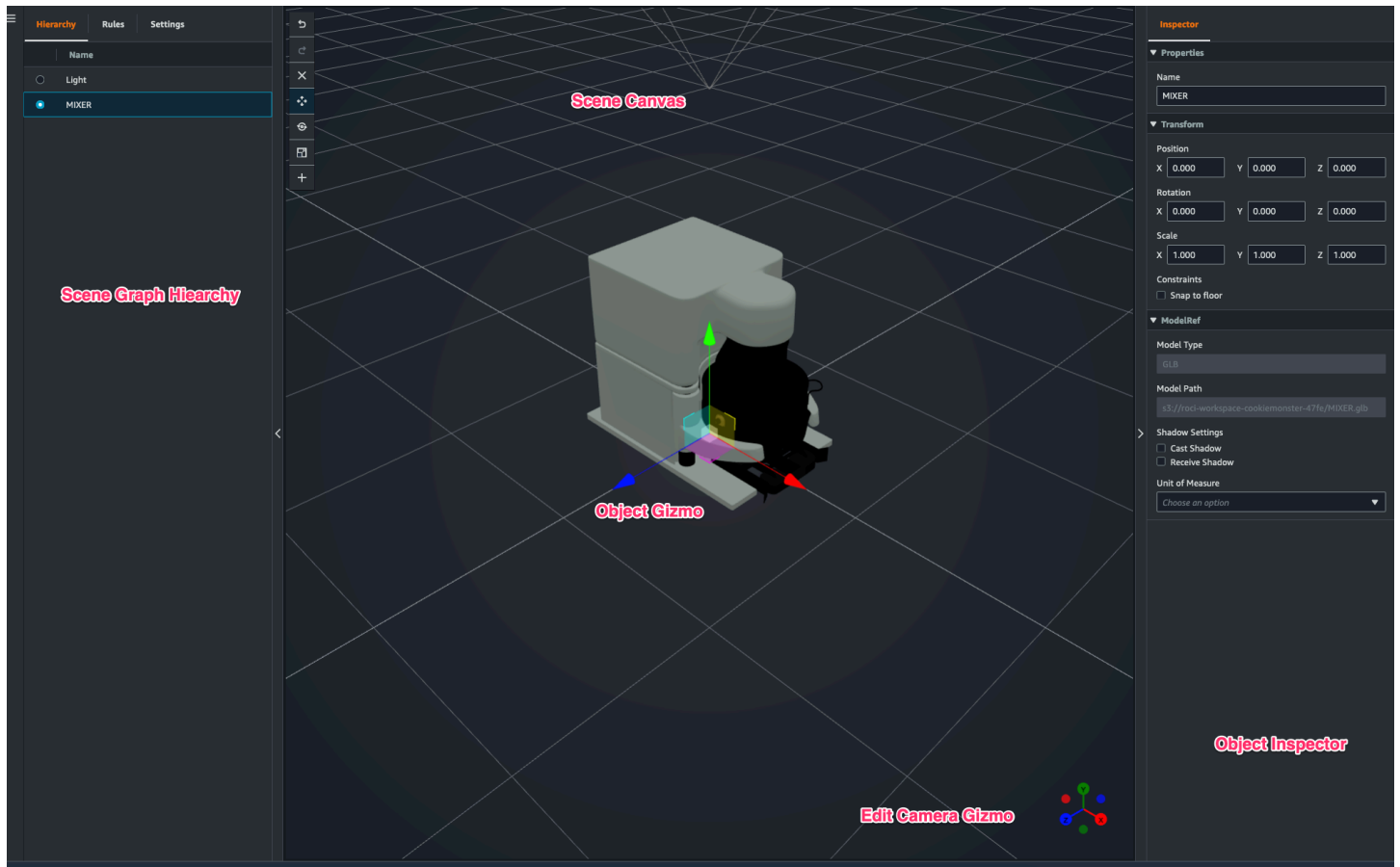


## 在場景中使用 3D 導覽 AWS IoT TwinMaker

AWS IoT TwinMaker 場景有一組導覽控制項，可讓您有效率地瀏覽場景的 3D 空間。若要與場景所代表的 3D 空間和物件互動，您可以使用下列 Widget 和選單選項。

- **Inspector**：使用「檢查器」視窗來檢視和編輯階層中所選實體或組件的屬性和設定。
- **場景畫布**：場景畫布是一種 3D 空間，您可以在其中定位和定向要使用的任何 3D 資源。
- **場景圖層次結構**：您可以使用此面板查看場景中存在的所有實體。它出現在窗口的左側。
- **物件控點**：使用此控點可以在畫布周圍移動物件。它會出現在場景畫布中所選 3D 物件的中心。
- **編輯相機控點**：使用「編輯相機控點」快速檢視場景視圖相機的目前方位並修改檢視角度。您可以在場景視圖的右下角找到此控點。
- **縮放控制項**：若要在場景畫布上導覽，請使用滑鼠右鍵按一下並朝您要移動的方向拖曳。要旋轉，請左鍵單擊並拖動以旋轉。若要縮放，請使用滑鼠上的滾輪，或在筆記型電腦的觸控板上捏合並移動手指。





層次結構窗格上的場景按鈕具有按鈕佈局的順序列出以下功能：

- 復原：復原場景中最後的變更。
- 重做：重做場景中的最後一次更改。
- 加號 (+)：使用此按鈕可存取下列動作：新增空白節點、新增 3D 模型、新增標籤、新增光源和新增模型著色器。
- 變更導覽方式：存取場景相機導覽選項「環轉」和「平移」。
- 垃圾桶 (刪除)：使用此按鈕可刪除場景中選取的物件。
- 物件操作工具：使用此按鈕可平移、旋轉和縮放選取的物件。

## 將固定攝影機加入至圖元

您可以將固定的攝影機視角附加到 AWS IoT TwinMaker 場景中的圖元。這些攝影機在 3D 模型上提供固定的透視，可讓您快速輕鬆地將場景中的透視轉移到目標圖元。

1. 在 [AWS IoT TwinMaker 主控台](#) 中導覽至您的場景。

2. 在場景階層選單中，選擇要附加攝影機的圖元。
3. 按 + 按鈕，然後從下拉選項中選擇從當前視圖添加相機。將具有目前透視的相機套用至圖元。
4. 在檢閱器中，您可以配置相機並調整以下設置：
  - 攝影機名稱
  - 相機位置和旋轉
  - 相機焦距
  - 縮放級別
  - 近截面和遠裁剪平面
5. 放置相機後存取相機。在階層架構中選取您加入攝影機的圖元。尋找圖元下列示的相機名稱。
6. 從實體中選擇置入的相機後，場景攝影機視圖將會對齊置入攝影機的設定透視。

## 場景增強編輯

AWS IoT TwinMaker 場景具有一組工具，可增強和編輯和操控場景中存在的資源。

下列主題將教導您如何在 AWS IoT TwinMaker 場景中使用增強的編輯功能。

- [場景物件的目標放置](#)
- [子模型選擇](#)
- [編輯場景階層中的圖元](#)

### 場景物件的目標放置

AWS IoT TwinMaker 可讓您精確地放置物件並將物件新增至場景。這項增強的編輯功能可讓您更好地控制在場景中放置標籤、實體燈光和模型的位置。

1. 在[AWS IoT TwinMaker 主控台](#)中導覽至您的場景。
2. 按 + 按鈕，然後從下拉選項中選擇其中一個選項。這可能是模型，燈光，標籤或 + 菜單中的任何內容。

當您在場景的 3d 空間中移動游標時，您應該會看到游標周圍的目標。

3. 使用目標在場景中精確地放置元素。

## 子模型選擇

AWS IoT TwinMaker 可讓您選取場景中 3D 模型的子模型，並將標準屬性套用至其中，例如標籤、光源或規則。

3D 模型檔案格式包含中繼資料，可將模型的子區域指定為較大模型內的子模型。例如，模型可以是過濾系統，系統各個部分（例如罐，管道或電機）被標記為過濾 3D 模型的子模型。

在場景中支持的 3D 文件格式：GLB 和 GLTF。

1. 在[AWS IoT TwinMaker 主控台](#)中導覽至您的場景。
2. 如果場景中沒有模型，請務必從「+」選單中選取選項來新增模型。
3. 選取場景階層中列出的模型，一旦選取階層，應該會在模型下顯示任何子模型。

### Note

如果您沒有看到列出任何子模型，則很可能該模型沒有配置為具有任何子模型。

4. 若要切換子模型的可見性，請按下位於階層中子模型名稱右側的眼睛圖示。
5. 若要編輯子模型資料（例如其名稱或位置），場景檢查器會在選取子模型時開啟。使用檢查器選單來更新或變更子模型資料。
6. 若要將標籤、光源、規則或其他性質加入至子模型，請在階層中選取子模型時按 +。

## 編輯場景階層中的圖元

AWS IoT TwinMaker 場景可讓您直接編輯階層表格中實體的屬性。下列程序顯示可透過階層功能表對實體執行哪些動作。

1. 在[AWS IoT TwinMaker 主控台](#)中導覽至您的場景。
2. 開啟場景階層，然後選取要操作之圖元的子元素。
3. 選取元素後，按下 + 按鈕，然後從下拉式清單中選取下列其中一個選項：
  - 新增空白節點
  - 加入 3D 模型
  - 增加光源
  - 從目前視圖加入相機

- 新增標籤
  - 新增模型著色器
  - 新增運動指示器
4. 從下拉式清單中選取其中一個選項後，選取項目將作為步驟 2 中所選元素的子項套用至場景。
  5. 您可以重新排序子元素和重新父元素，方法是選取子元素，然後在階層中拖曳至新的父元素。

## 將註釋新增至實體

AWS IoT TwinMaker 場景撰寫器可讓您為場景階層中的任何元素加上註解。註釋是以降價編寫的。

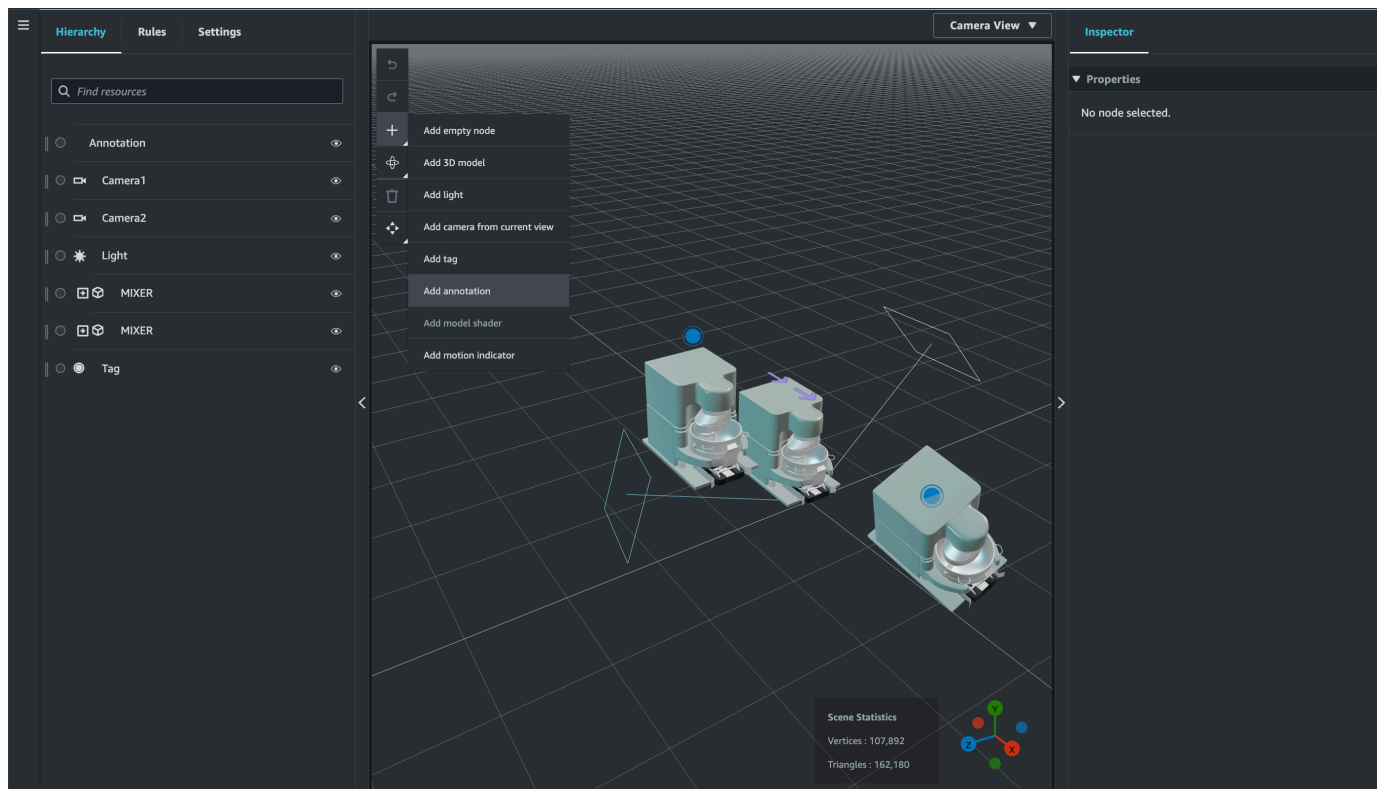
有關在 Markdown 中編寫的更多信息，請參閱關於降價語法，[基本語法](#)的官方文檔。

### Note

AWS IoT TwinMaker 註釋和覆蓋降價語法而不是 HTML。

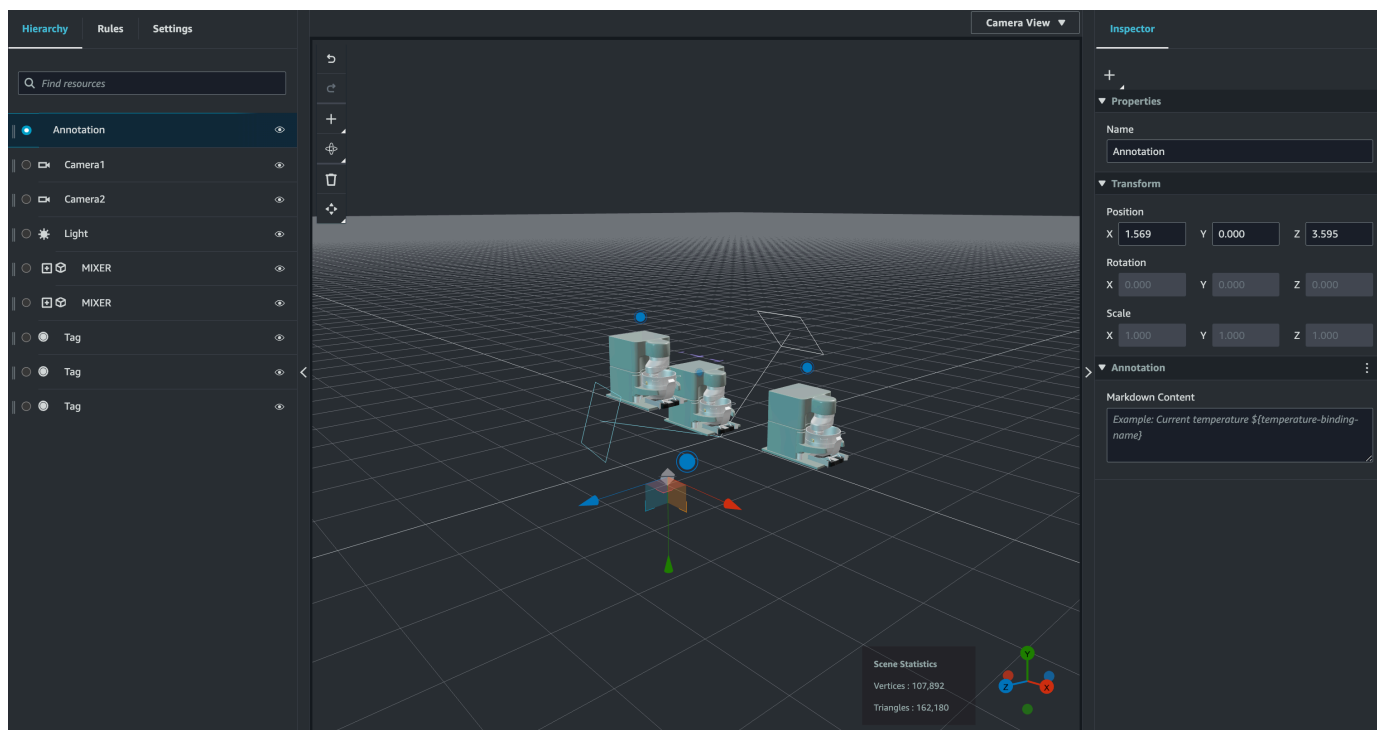
## 將註釋新增至實體

1. 在[AWS IoT TwinMaker 主控台](#)中導覽至您的場景。
2. 從場景階層中選取您要註解的元素。如果未選取階層中的元素，則您可以將註釋新增至根。
3. 按加號 + 按鈕，然後選擇添加註釋選項。



4. 在左側的 Inspector 窗口中，向下滾動到註釋部分。使用 Markdown 語法編寫您希望註釋顯示的文本。

有關在 Markdown 中編寫的更多信息，請參閱關於降價語法，[基本語法](#)的官方文檔。



5. 若要將 AWS IoT TwinMaker 場景資料繫結至註釋，請選擇 [新增資料繫結]，新增 [實體 ID]，然後選取要從中顯示資料的實體的 [元件名稱] 和 [屬性名稱]。您可以更新繫結名稱以將其用作 Markdown 變數，並在註釋中顯示資料。

### Inspector

+

▼ Properties

Name

Annotation

▼ Transform

Position

X	1.569	Y	0.000	Z	3.595
---	-------	---	-------	---	-------

Rotation

X	0.000	Y	0.000	Z	0.000
---	-------	---	-------	---	-------

Scale

X	1.000	Y	1.000	Z	1.000
---	-------	---	-------	---	-------

▼ Annotation

Markdown Content

Add data binding

*Example: Current temperature `${temperature-binding-name}`*

## Inspector

+

### ▼ Properties

Name

### ▼ Transform

Position

X	<input type="text" value="1.569"/>	Y	<input type="text" value="0.000"/>	Z	<input type="text" value="3.595"/>
---	------------------------------------	---	------------------------------------	---	------------------------------------

Rotation

X	<input type="text" value="0.000"/>	Y	<input type="text" value="0.000"/>	Z	<input type="text" value="0.000"/>
---	------------------------------------	---	------------------------------------	---	------------------------------------

Scale

X	<input type="text" value="1.000"/>	Y	<input type="text" value="1.000"/>	Z	<input type="text" value="1.000"/>
---	------------------------------------	---	------------------------------------	---	------------------------------------

### ▼ Annotation ⋮

Markdown Content

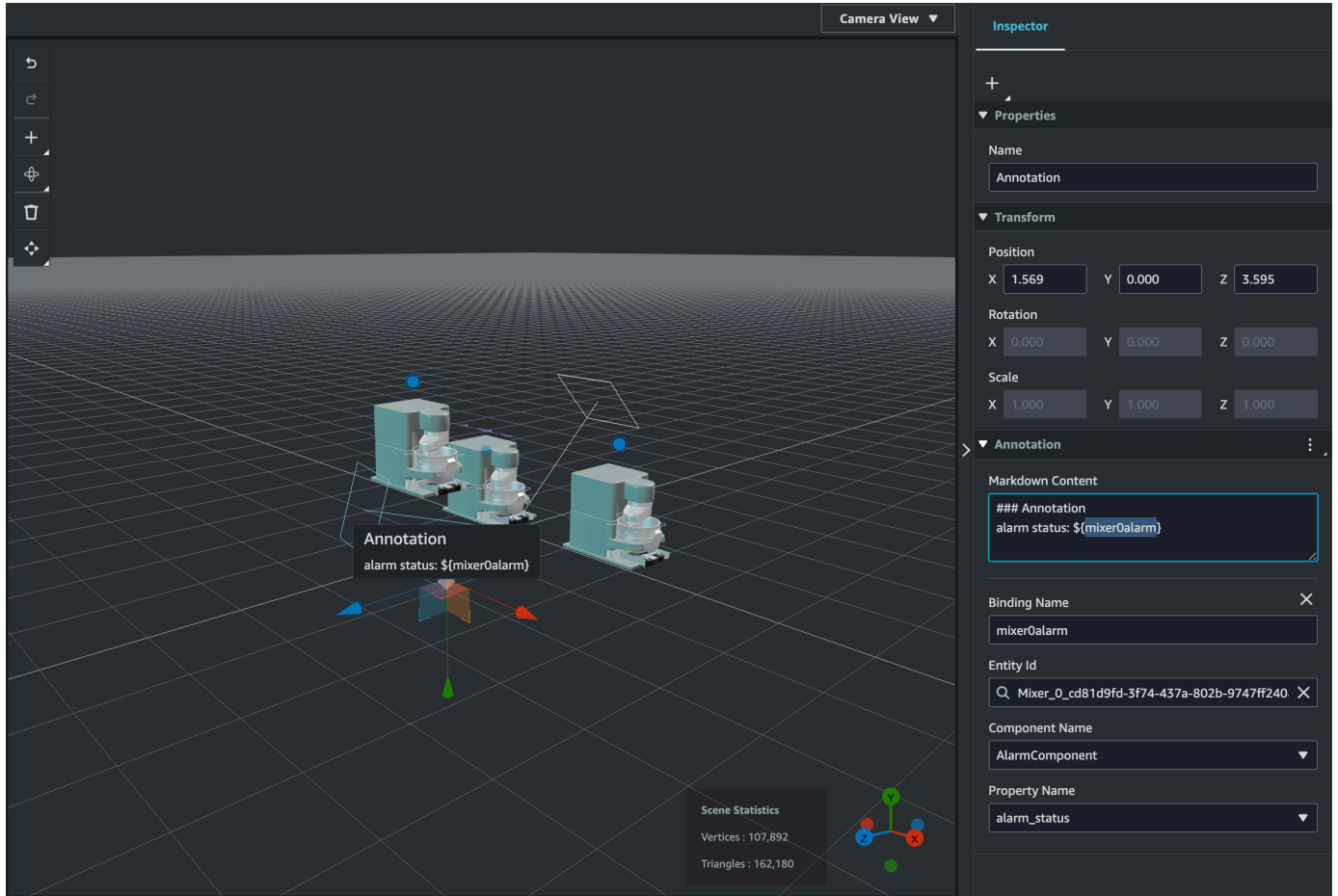
```
Example: Current temperature ${temperature-binding-name}
```



## 6. 綁定名稱用於表示註釋的變量。

輸入繫結名稱，透 AWS IoT TwinMaker 過變數語法在註釋中顯示實體時間序列的最新歷史值：`${variable-name}`

例如，此覆蓋圖會以語法在 `mixer0alarm` 註釋中顯示的值 `${mixer0alarm}`。



## 將覆蓋添加到標籤

您可以為 AWS IoT TwinMaker 場景建立疊加層。場景覆蓋與標籤相關聯，可用於顯示與場景圖元相關聯的重要資料。覆蓋圖是在降價中編寫和渲染的。

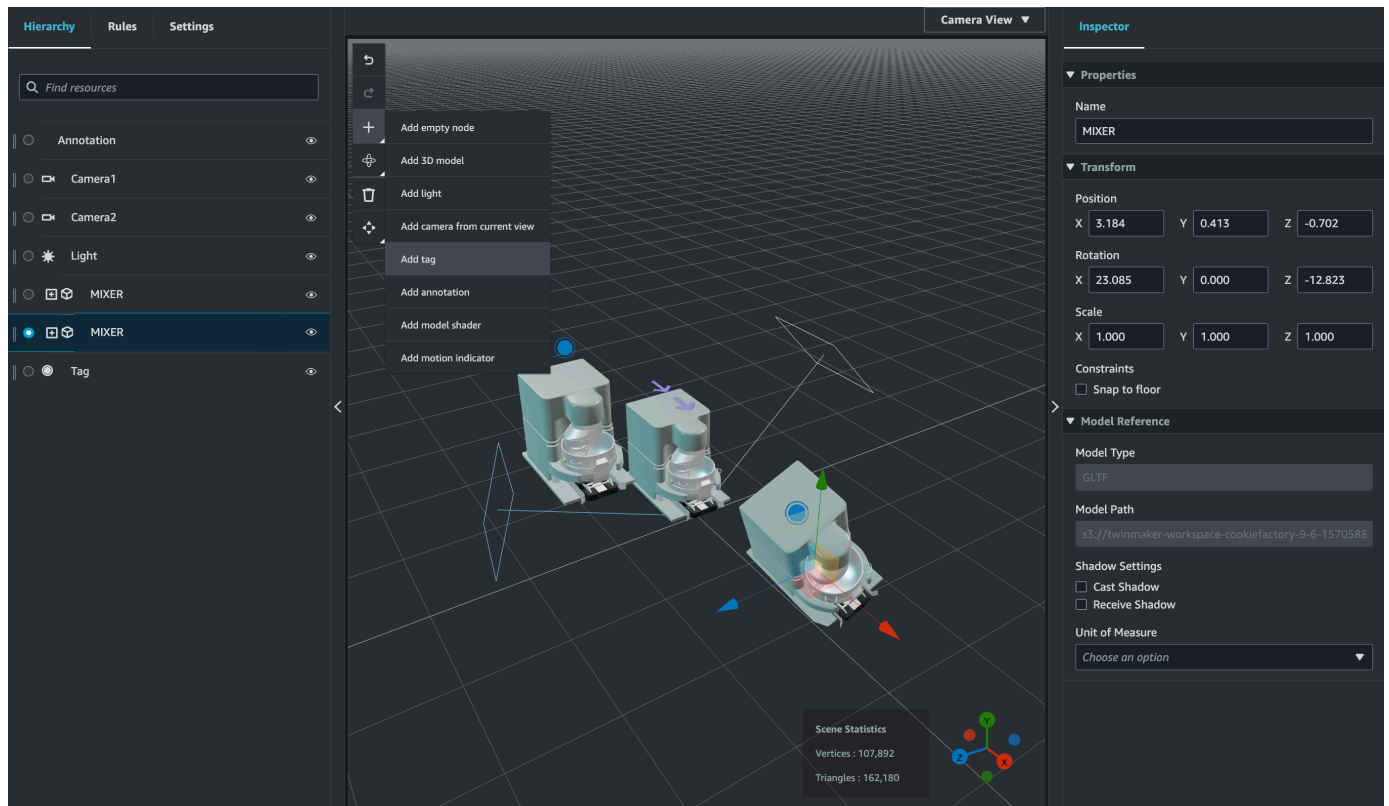
有關在 Markdown 中編寫的更多信息，請參閱關於降價語法，[基本語法](#)的官方文檔。

**Note**

依預設，只有在選取與其關聯的標籤時，覆蓋才會在場景中可見。您可以在場景設置中切換此選項，以便一次可見所有疊加層。

1. 在[AWS IoT TwinMaker 主控台](#)中導覽至您的場景。
2. AWS IoT TwinMaker 覆蓋與標籤場景相關聯，您可以更新現有標籤或添加新標籤。

按加號 + 按鈕，然後選擇添加標籤選項。



3. 在右側的「Inspector」面板中，選取「+」(加號) 按鈕，然後選取「新增覆蓋」。

## Inspector



Add overlay

Add entity binding

### Default Icon

Choose an icon

### Entity Id



### Component Name

Select an option

### Property Name

Select an option

### Rule Id

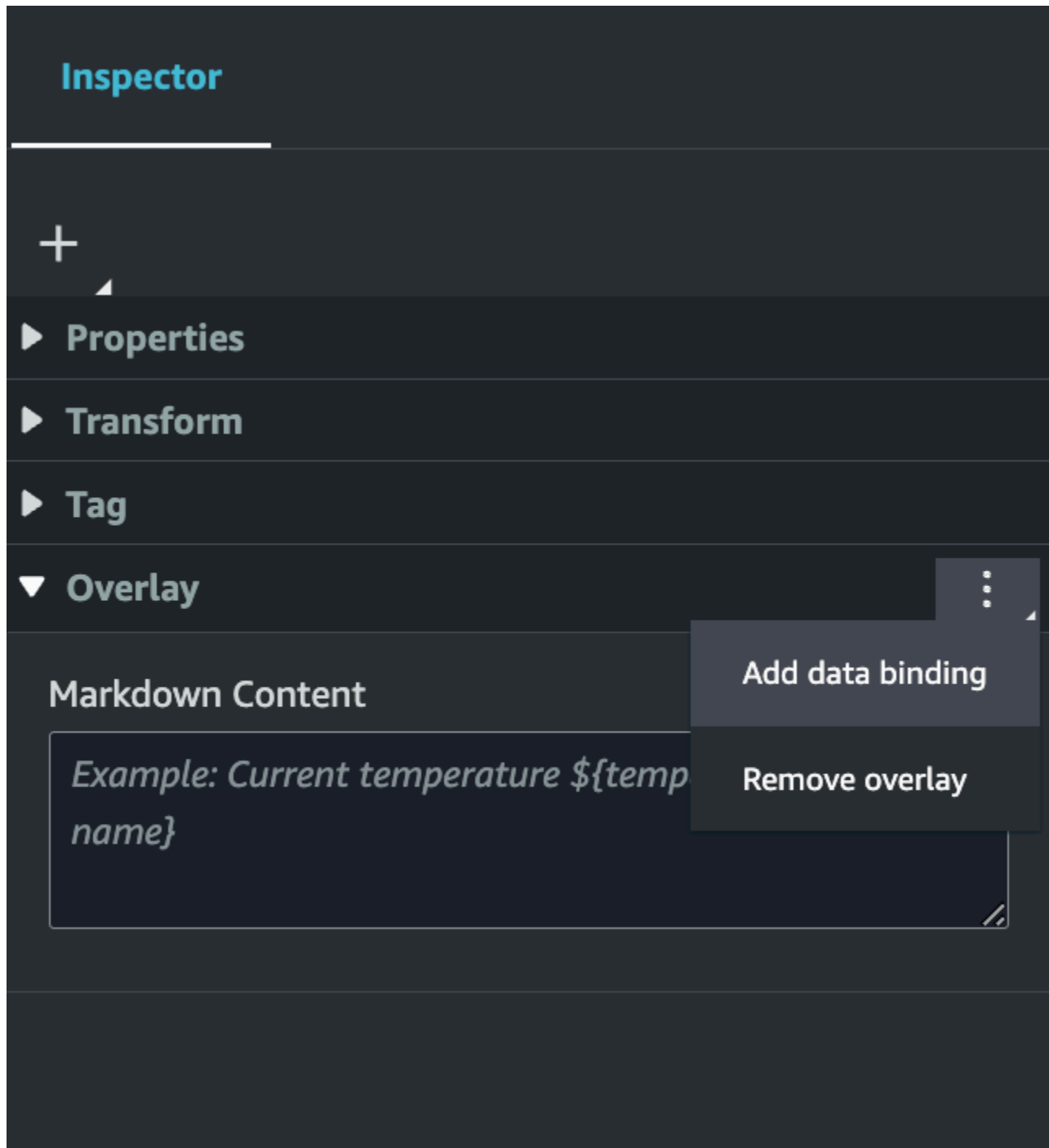
Choose a rule

### Link Target

4. 在 Markdown 語法中，編寫您希望疊加層顯示的文本。

有關在 Markdown 中編寫的更多信息，請參閱關於降價語法，[基本語法](#)的官方文檔。

5. 若要將 AWS IoT TwinMaker 場景資料繫結至覆蓋圖，請選取 [新增資料繫結]。



新增繫結名稱和實體 ID，然後選取要從中顯示資料的實體的元件名稱和屬性名稱。

- 您可以透過 AWS IoT TwinMaker 下列變數語法，在覆蓋中顯示實體時間序列資料的最新歷史值：`${variable-name}`。

例如，此覆蓋圖會在覆蓋 `mixer0alarm` 中以語法顯示的值 `${mixer0alarm}`。

# Inspector



▶ Properties

▶ Transform

▶ Tag

▼ Overlay ⋮

## Markdown Content

```
### Overlay  
alarm status: ${mixer0alarm}
```

Binding Name ✕

mixer0alarm

Entity Id

🔍 Mixer\_0\_cd81d9fd-3f74-437a-802b-9747ff240 ✕


Component Name

AlarmComponent ▼

Property Name

將覆蓋添加到標籤

7. 要啟用覆蓋可見性，請打開左上角的「設置」選項卡，並確保「覆蓋」開關已打開，以便一次可以看到所有覆蓋。

 Note

依預設，只有在選取與其關聯的標籤時，覆蓋才會在場景中可見。

**Hierarchy** | **Rules** | **Settings**

▼ **Current View Setting**

**Toggle visibility**

Motion indicator

Tags

Overlay

Annotation

▼ **Scene Settings**

Environment Preset

Choose an environment ▼

▼ **Tag Settings**

Scale

5.000  Fixed scaling

**Overlay**

Visibility settings

Always on

▼ **Data Binding Template**

sel\_entity

Select an option ▼

sel\_comp



## 編輯您的場景

建立場景後，您可以在場景中新增實體、元件和設定增強型 Widget。使用實體元件和 Widget 來建立數位孿生模型，並提供符合您使用案例的功能。

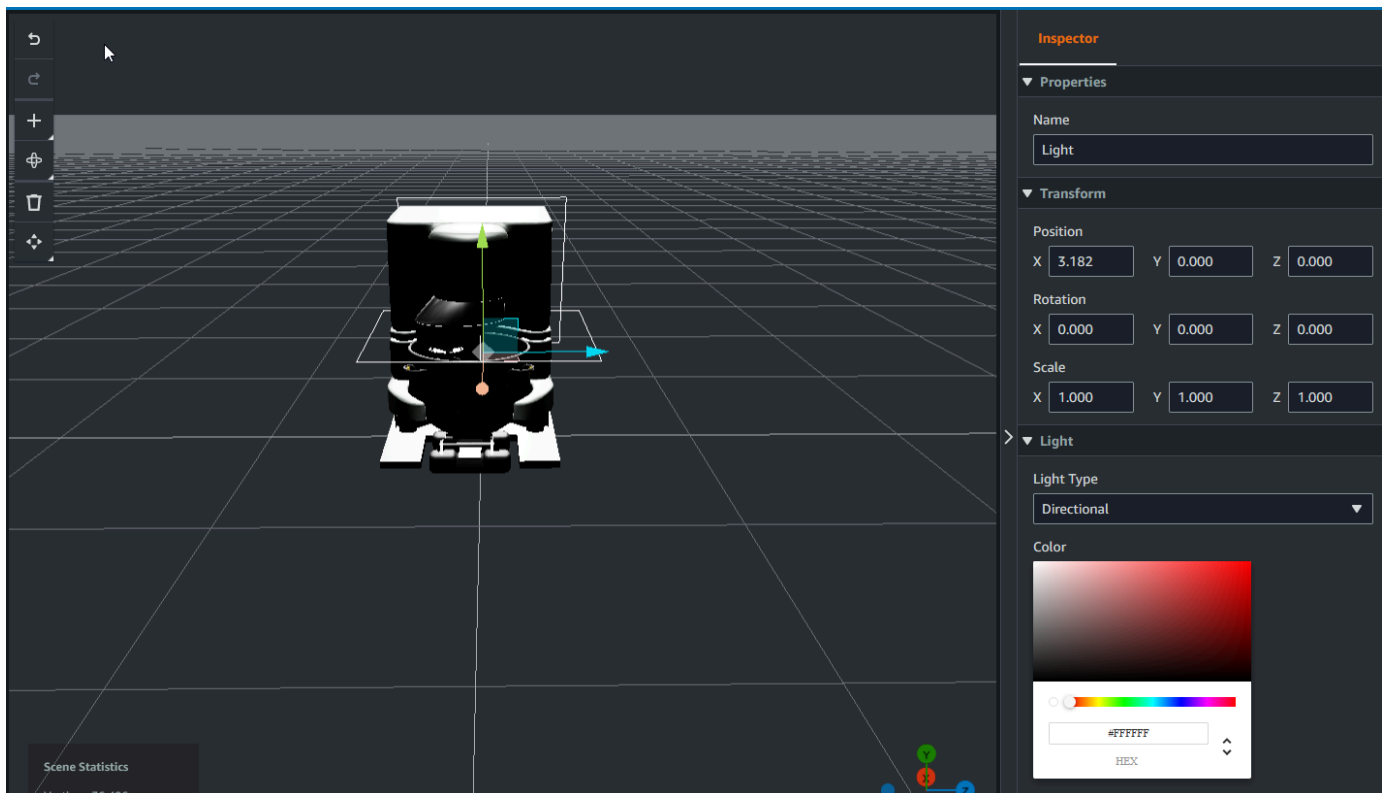
### 將模型新增至場景

若要將模型新增至您的場景，請使用下列步驟。

#### Note

要在場景中添加模型，您必須先將模型上傳到 AWS IoT TwinMaker 資源庫。如需詳細資訊，請參閱 [上傳資源至資 AWS IoT TwinMaker 源庫](#)。

1. 在場景撰寫器頁面上，選擇加號 (+)，然後選擇「新增 3D 模型」。
2. 在 [從資源庫新增資源] 視窗中，選擇 CookieFactorMixer.glb 檔案，然後選擇 [新增]。場景撰寫器開啟。
3. 選用性：選擇加號 (+)，然後選擇「增加光源」。
4. 選擇每個光源選項以查看它們如何影響場景。



**Note**

場景具有預設的環境光源。若要避免影格速率損失，請考慮限制場景中放置的其他光源數量。

## 將模型著色器增強 UI 小部件添加到您的場景

模型著色器 Widget 可以在您定義的條件下變更物件的顏色。例如，您可以建立顏色 Widget，根據混合器的溫度資料變更場景中 Cookie 混合器的顏色。

請使用下列程序，將模型著色器 Widget 新增至選取的物件。

1. 在階層中選取您要新增 Widget 的物件。按 + 按鈕，然後選擇「模型著色器」。
2. 若要新增視覺化規則群組，請先依照下列指示建立 ColorRule，然後在「規則 ID」物件的「Inspector」面板中選擇ColorRule。
3. 選取 EntityId ComponentName，並且 PropertyName 您想要將模型著色器繫結到的目標。

## 為場景建立視覺規則

您可以使用視覺化規則對應來指定變更增強 UI Widget (例如標籤或模型著色器) 視覺外觀的資料驅動條件。提供了範例規則，但您也可以建立自己的規則。下列範例顯示視覺化規則。

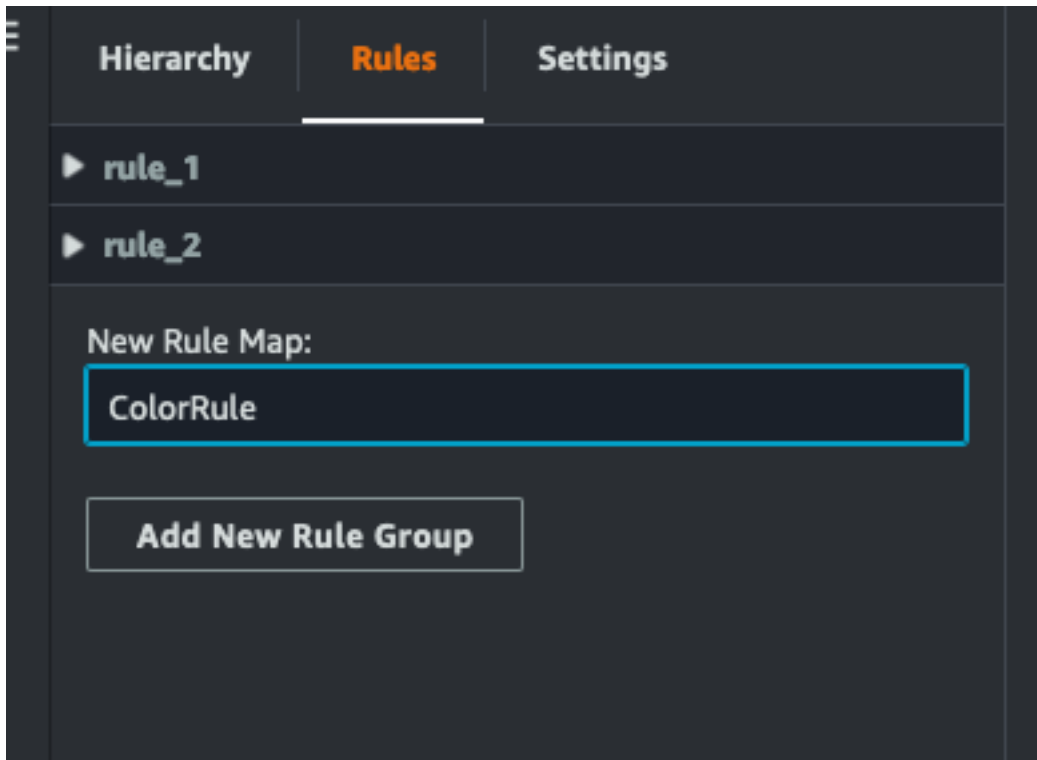
The screenshot displays the configuration interface for a rule in AWS IoT TwinMaker. It features a dark theme and a vertical scroll bar on the right. The interface is organized into sections for each statement and a bottom section for rule management.

- Statement 1:**
  - Expression: `temperature >= 40`
  - Target: Icon (dropdown), Error (dropdown), and a red 'X' icon.
  - Action: Remove statement
- Statement 2:**
  - Expression: `temperature >= 20`
  - Target: Icon (dropdown), Warning (dropdown), and a yellow warning icon.
  - Action: Remove statement
- Statement 3:**
  - Expression: `temperature < 20`
  - Target: Icon (dropdown), Info (dropdown), and a blue info icon.
  - Action: Remove statement
- Bottom Section:**
  - Action: Add new statement
  - Action: Remove Rule
  - Rule Name: `sampleTimeSeriesColorRule`
  - Rule Id: [Empty text input field]

上圖顯示了針對特定值檢查 ID 為 '溫度' 的先前定義的資料屬性時的規則。例如，如果「溫度」大於或等於 40，則狀態會將標籤的外觀變更為紅色圓圈。在 Grafana 儀表板中選擇目標時，會填入設定為使用相同資料來源的詳細資料面板。

下列程序說明如何為網格著色增強 UI 層新增視覺規則群組。

1. 在主控台的「規則」索引標籤下，輸入文字欄位 ColorRule 中的名稱，然後選擇「新增規則群組」。



2. 為您的使用案例定義新規則。例如，您可以根據資料屬性「溫度」建立一個，其中報告的值小於 20。對規則運算式使用下列語法：小於是 <、大於 >、小於或等於 <=、大於或等於 >=，而等於則為 ==。如需詳細資訊，請參閱[阿帕奇共享資源 JEXL 語法](#)。)
3. 將目標設定為顏色。若要定義顏色，例如 #fcba03，請使用十六進位值。如需十六進位值的詳細資訊，請參閱[十六進位](#)。)

## 為場景建立標籤

標籤是加入至場景特定  $x, y, z$  座標位置的註解。標籤使用實體屬性將場景零件連接到知識圖。您可以使用標籤來配置場景中項目的行為或視覺外觀，例如鬧鐘。

**Note**

若要將功能新增至標籤，您可以將視覺規則套用至標籤。

使用下列程序將標籤新增至場景。

1. 選取階層中的物件，選擇「+」按鈕，然後選擇「新增標記」。
2. 命名標籤。然後，若要套用視覺化規則，請選取視覺群組 ID。
3. 在下拉式清單中，選擇 EntityID ComponentName、和。 PropertyName
4. 若要填入「資料路徑」欄位，請選擇「建立」 DataFrameLabel。

## 3D 瓷磚模型格式

### 在場景中使用 3D 瓷磚

如果您在中載入 3D 場景時遇到很長的等待時間，AWS IoT TwinMaker 或者導覽複雜的 3D 模型時顯示效能不佳，則可能需要將模型轉換為 3D 拼貼。本節說明 3D 拼貼格式和可用的協力廠商工具。請繼續閱讀以決定 3D Tiles 是否適合您的使用案例，以及開始使用的說明。

### 複雜模型使用案例

如果模型為以下情況，AWS IoT TwinMaker 場景中的 3D 模型可能會導致效能問題，例如載入時間緩慢和導覽延遲：

- 大：它的文件大小大於 100MB。
- 密集：它由數百或數千個不同的網格組成。
- 複雜：網面幾何圖形具有數百萬個三角形，可形成複雜形狀。

### 3D 瓷磚格式

[3D 瓷磚格式](#)是用於串流模型幾何和改善 3D 渲染效能的解決方案。它可以在 AWS IoT TwinMaker 場景中立即載入 3D 模型，並根據相機視圖中可見的內容在模型的區塊中載入，以最佳化 3D 互動。

3D 瓷磚格式是由[鉅](#)創建的。鉅有一個管理的服務，將 3D 模型轉換為 3D 瓷磚叫[鉅離子](#)。這是目前建立 3D 瓷磚的最佳解決方案，我們建議您使用[支援的格式](#)複雜模型。您可以在 Cesium 的定價頁面上註冊 Cesium 並根據您的業務需求選擇合適的[訂閱計劃](#)。

要準備可以添加到 AWS IoT TwinMaker 場景的 3D 瓷磚模型，請按照鉅離子記錄的說明進行操作：

- [將模型導入鉅離子](#)

## 上傳鉅 3D 瓷磚 AWS

將模型轉換為 3D Tiles 後，請下載模型檔案，然後將它們上傳到您的 AWS IoT TwinMaker 工作區 Amazon S3 儲存貯體：

1. [創建並下載您的 3D 瓷磚模型存檔](#)。
2. 將存檔解壓縮到文件夾中。
3. 將整個 3D 圖塊資料夾上傳到與您的 AWS IoT TwinMaker 工作區相關聯的 Amazon S3 儲存貯體。請參閱 Amazon S3 使用者指南中的[上傳物件](#)。)
4. 如果您的 3D Tiles 模型已成功上傳，您將在 AWS IoT TwinMaker [資源庫中看到具有類型的 Amazon S3 資料夾路徑Tiles3D](#)。

### Note

資 AWS IoT TwinMaker 源庫不支持直接上傳 3D 圖塊模型。

## 在中使用 3D 瓷磚 AWS IoT TwinMaker

AWS IoT TwinMaker 知道任何上傳到您的工作區 S3 存儲桶的 3D 瓷磚模型。

模型必須在同一個 Amazon S3 目錄中提供一個tileset.json和所有相依檔案

(.gltf、.b3dm、.i3dm、.cmpt、.pnst)。Amazon S3 目錄路徑將以類型顯示在資源庫中Tiles3D。

若要將 3D 瓷磚模型新增至您的場景，請遵循下列步驟：

1. 在場景撰寫器頁面上，選擇加號 (+)，然後選擇「新增 3D 模型」。
2. 在 [從資源庫新增資源] 視窗中，選擇具有類型的 3D 磚模型路徑Tiles3D，然後選擇 [新增]。
3. 在畫布上按一下，將模型放置在場景中。

## 3D 瓷磚的差異

3D 瓷磚目前不支援幾何和語義中繼資料，這表示原始模型的網格階層不適用於子模型選取功能。您仍然可以將 Widget 新增至 3D Tiles 模型，但您無法使用微調至子模型的功能：模型著色器、分離的 3D 轉換或子模型網格的實體繫結。

建議將 3D 並排轉換用於做為場景背景背景的大型資產。如果您希望子模型進一步分解和註釋，則應將其作為單獨的 GLTF/GLB 資產提取並直接添加到場景中。這可以通過免費和常見的 3D 工具，如 [Blender](#) 來完成。

範例使用案例：

- 您擁有 1GB 的工廠模型，其中包含詳細的機房和地板、電箱和管道。當關聯的性質資料超過臨界值時，電氣箱和管需要發紅光。
- 您可以隔離模型中的方塊和管網格，並使用 Blender 將其匯出到單獨的 GLTF 中。
- 您可以將沒有電氣和衛工元素的工廠轉換為 3D Tiles 模型，並將其上傳到 S3。
- 您可以將 3D 並排模型和 GLTF 模型加入到原點 (0,0,0) 的 AWS IoT TwinMaker 全景中。
- 您可以將模型著色器元件加入 GLTF 的電子箱和管路子模型，以根據屬性規則使網格變成紅色。

## 動態場景

AWS IoT TwinMaker 場景通過在實體組件中存儲場景節點和設置來解鎖[知識圖](#)的力量。使用主 AWS IoT TwinMaker 控制台建立動態場景，以便更輕鬆地管理、建置和彩現 3D 場景。

主要功能：

- 所有 3D 場景節點物件、設定和資料繫結都會根據知識圖查詢「動態」呈現。
- 如果您在 Grafana 或自訂應用程式中使用唯讀場景檢視器，則可以每隔 30 秒取得場景的更新。

## 靜態與動態場景

靜態場景由存放在 S3 中的場景 JSON 檔案組成，該檔案包含所有場景節點和設定的詳細資料。場景的任何變更都必須對 JSON 文件進行，並儲存到 S3。如果您有[基本定價計劃](#)，靜態場景是唯一的選擇。

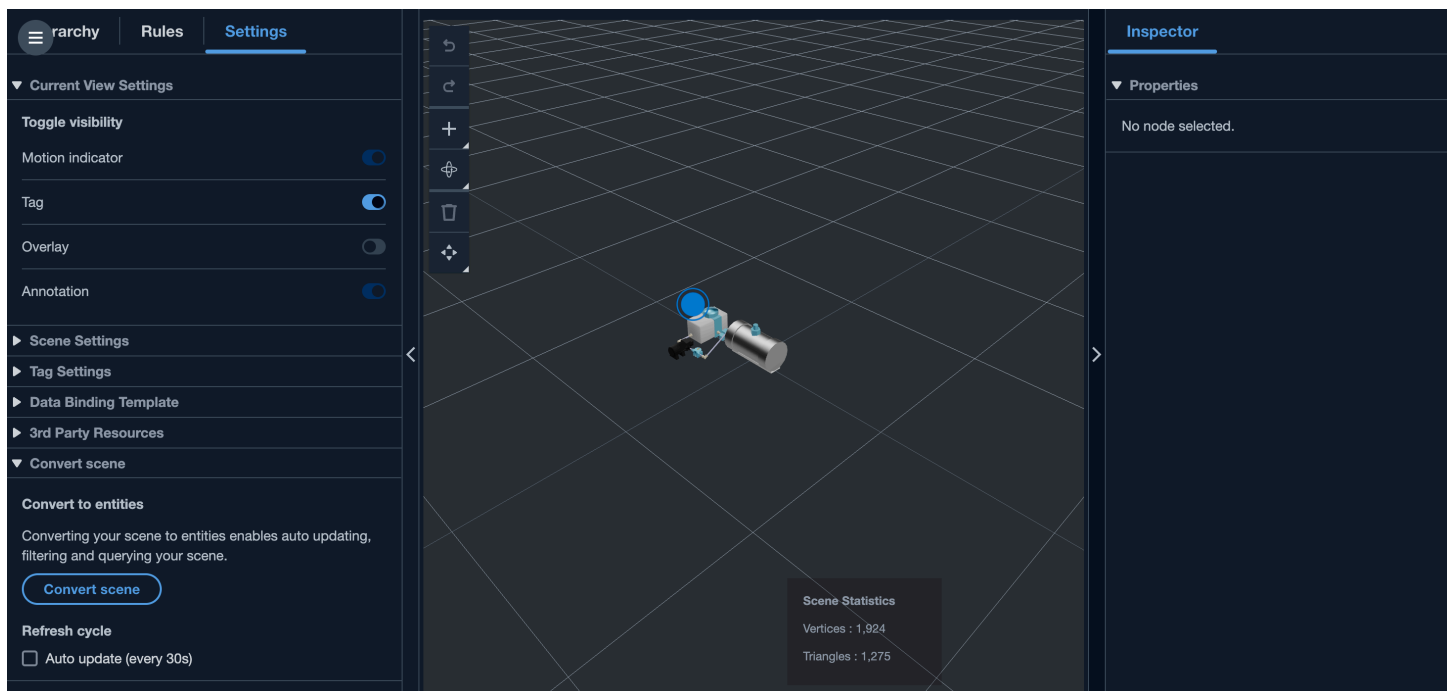
動態場景是由具有場景全域設定的場景 JSON 檔案所組成，而所有其他場景節點和節點設定則會儲存為知識圖中的實體元件。動態場景僅在標準和分層組合定價方案中受支援。如[切換 AWS IoT TwinMaker 定價模式](#)需如何升級定價方案的資訊，請參閱。

您可以按照以下步驟將現有的靜態場景轉換為動態場景：

- 在[AWS IoT TwinMaker 主控台](#)中導覽至您的場景。
- 在左側面板上，按一下「設定」標籤。
- 展開面板底部的「轉換場景」區段。
- 按一下轉換場景按鈕，然後按一下確認。

### Warning

從靜態場景轉換為動態場景是不可逆轉的。



## 全景零組件類型和圖元

為了建立場景特有的圖元元件，支援下列 1P 元件類型：

- [相機存儲攝像機小部件設置的組件類型。](#)
- [數據覆蓋一種組件類型，用於存儲註釋或標籤小器具覆蓋設置的組件類型。](#)
- 光源一種組件類型，用於存儲光源小部件的設置。
- 模型參考一種組件類型，用於存儲場景中使用的 3D 模型的設置和 S3 位置。



- [模型著色器在 3D 模型上儲存模型著色器設定的組件類型。](#)
- 運動指示器一種組件類型，用於儲存運動指示器小部件的設置。
- [組件. 子模型參考儲存 3D 模型之子模型設定的元件類型。](#)
- [標籤一種元件類型，用於儲存標籤元件的設定。](#)
- 節點一種組件類型，用於儲存場景節點的基本設置，例如其 3D 轉換，名稱和通用屬性。

## 動態場景概念

動態場景圖元儲存在標有標示\$SCENES的整體圖元之下。每個場景都由一個根實體和與場景節點層次結構相匹配的子實體層次結構組成。根下的每個場景節點都有一個 `com.amazon.iottwinmaker.3d.node` 組件和節點類型的組件（3D 模型，窗口小部件等）。

### Warning

請勿手動刪除任何場景圖元，否則您的場景可能處於中斷狀態。如果您想要部分或完全刪除場景，請使用場景撰寫器頁面來新增和刪除場景節點，然後使用場景頁面來選取和刪除場景。

# 使用 AWS IoT TwinMaker UI 組件創建自定義 Web 應用程序

AWS IoT TwinMaker 為 AWS IoT 應用程序開發人員提供開源 UI 組件。使用這些 UI 組件，開發人員可以構建自定義的 Web 應用程序，並為其數字雙胞胎啟用 AWS IoT TwinMaker 功能。

AWS IoT TwinMaker UI 元件是 Ap AWS IoT plication Kit 的一部分，這是一個開放原始碼的用戶端程式庫，可讓 IoT 應用程式開發人員簡化複雜 IoT 應用程式的開發

AWS IoT TwinMaker UI 組件包括：

- AWS IoT TwinMaker 來源：

一種資料連接器元件，可讓您擷取資料並與資 AWS IoT TwinMaker 料和數位雙胞胎互動。

如需詳細資訊，請參閱[AWS IoT TwinMaker 來源](#)文件。

- 場景檢視器：

構建的 3D 渲染組件，渲染您@react-three/fiber的數字學生，並使您能夠與它進行交互。

如需詳細資訊，請參閱[場景檢視器](#)文件。

- 視頻播放器：

一種視訊播放器元件，可讓您透過 AWS IoT TwinMaker Kinesis 視訊串流來串流視訊。

如需詳細資訊，請參閱[影片播放](#)程式文件。

要了解有關使用 AWS IoT 應用工具包的更多信息，請訪問[AWS IoT 應用程序工具包 Github](#) 頁

有關如何使用應用程序工具包啟動新 Web 應用程 AWS IoT 序的說明，請訪問官方的 [IoT 應用程序 Kit](#) 文檔頁面。

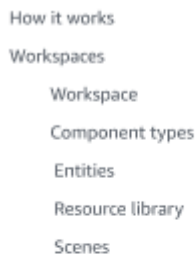
# 切換 AWS IoT TwinMaker 定價模式

AWS IoT TwinMaker 目前有三種定價模式：基本、標準或分層組合。標準定價模式設定為預設定價模式。





您可以隨時從以使用量為基礎的定價模式切換為階層式的定價模式，但變更會在下一個帳單週期開始生效。從以使用量為基準的訂價模式切換至以階層為基準的訂價模式後，您就無法針對接下來的三個使用週期切換回以使用量為基準的訂價模式。如果您從基本切換到標準，則變更會立即生效。如需詳細資訊和費用資訊，請參閱[AWS IoT TwinMaker 定價](#)

此程序說明如何在[AWS IoT TwinMaker 主控台](#)中切換定價模式：

1. 開啟 [AWS IoT TwinMaker 主控台](#)。
2. 在左側導覽窗格中，選取 [設定]。「定價」頁面隨即開啟。



How it works  
Workspaces  
    Workspace  
    Component types  
    Entities  
    Resource library  
    Scenes  
**Settings**

What's new   
Documentation   
FAQ   
Pricing 

3. 選擇 [變更價格模式]。
4. 選取「標準」或「階層」套裝軟體模式，如下列螢幕擷取畫面所示。

### Select price mode

**Basic**  
Basic pricing mode is determined by the data access calls sent during the current billing cycle. Does not include Knowledge Graph.

**Standard (current price mode)**  
Standard pricing mode is determined by the entities used, queries made, and data access calls sent during the current billing cycle.

**Tiered bundle**  
Tiered bundle pricing mode is based on 4 tiers of usage. Each tier is set by number of entities, and a usage threshold based on queries made.

---

### Standard pricing

The Standard pricing mode is determined by the entities used, queries made, and data access calls sent during the current billing cycle.

Pricing element	Pricing unit	Usage threshold
Unified data access calls	per MM	n/a
Queries	per 10K	n/a
Entities	per entity/month	n/a

Cancel Save

5. 選擇 [儲存] 以確認新的定價模式。
6. 您現在已變更定價模式。

#### i Note

您可以隨時從以使用量為基礎的定價模式切換為階層式的定價模式，但變更會在下一個帳單週期開始生效。從以使用量為基準的訂價模式切換至以階層為基準的訂價模式後，您就無法針對接下來的三個使用週期切換回以使用量為基準的訂價模式。如果您從基本切換到標準，則變更會立即生效。

# AWS IoT TwinMaker 知識圖

AWS IoT TwinMaker 知識圖表會組織 AWS IoT TwinMaker 工作區中包含的所有資訊，並以視覺化圖形格式呈現。您可以針對實體、元件和元件類型執行查詢，以產生可顯示 AWS IoT TwinMaker 資源之間關係的視覺圖形。

下列主題說明如何使用及整合知識圖表。

## 主題

- [AWS IoT TwinMaker 知識圖核心概念](#)
- [如何執行 AWS IoT TwinMaker 知識圖查詢](#)
- [知識圖場景整合](#)
- [如何使用 AWS IoT TwinMaker 知識圖與 Grafana](#)
- [AWS IoT TwinMaker 知識圖表其他資源](#)

## AWS IoT TwinMaker 知識圖核心概念

本主題涵蓋知識圖功能的關鍵概念和詞彙。

知識圖如何工作：

知識圖創建實體及其組件與現有 [CreateEntity](#) 或 [UpdateEntity](#) API 之間的關係。關係是在實體的一個組件上定義的特殊數據類型 [關係](#) 的只是一個屬性。AWS IoT TwinMaker 知識圖會呼叫 [ExecuteQuery](#) API，根據實體中的任何資料或它們之間的關係進行查詢。知識圖使用彈性的 PartiQL 查詢語言 (許多 AWS 服務使用)，新增的圖形比對語法支援可協助您撰寫查詢。進行呼叫後，您可以將結果視為表格或將其視覺化為連接節點和邊緣的圖形。

知識圖關鍵術語：

- **實體圖表**：工作區內節點和邊緣的集合。
- **節點**：工作區中的每個實體都會成為實體圖形中的節點。
- **邊緣**：在實體元件上定義的每個關聯屬性都會成為實體圖形中的邊緣。此外，使用實體 `parentEntityId` 欄位定義的階層式父子關係也會成為實體圖形中具有 "isChildOf" 關聯名稱的邊緣。所有邊均為方向性邊緣。
- **關係**：AWS IoT TwinMaker 關係是一個實體的組件的一種特殊類型的屬性。您可以使用 AWS IoT TwinMaker [CreateEntity](#) 或 [UpdateEntity](#) API 來定義和編輯關係。在中 AWS IoT

TwinMaker，必須在實體的元件中定義關係。關係無法定義為隔離的資源。關係必須是從一個圖元到另一個圖元的方向性。

## 如何執行 AWS IoT TwinMaker 知識圖查詢

使用 AWS IoT TwinMaker 知識圖表之前，請確定您已完成下列必要條件：

- 建立 AWS IoT TwinMaker 工作區。您可以在[AWS IoT TwinMaker 主控台](#)中建立工作區。
- 熟悉 AWS IoT TwinMaker 實體元件系統以及如何建立實體。如需詳細資訊，請參閱 [建立您的第一個實體](#)。
- 熟悉 AWS IoT TwinMaker 的資料連接器。如需詳細資訊，請參閱 [AWS IoT TwinMaker 資料連接器](#)。

### Note

若要使用 AWS IoT TwinMaker 知識圖表，您必須採用標準或分層組合定價模式。如需詳細資訊，請參閱 [切換 AWS IoT TwinMaker 定價模式](#)。

下列程序說明如何撰寫、執行、儲存和編輯查詢。

### 開啟查詢編輯器

若要導覽至知識圖查詢編輯器

1. 開啟 [AWS IoT TwinMaker 主控台](#)。
2. 開啟您要在其中使用知識圖的工作區。
3. 在左側導覽功能表中，選擇 [查詢編輯器]。
4. 查詢編輯器隨即開啟。您現在可以在工作區的資源上執行查詢了。

### 執行查詢

若要執行查詢並產生圖形

1. 在查詢編輯器中，選擇「編輯器」頁籤以開啟語法編輯器。
2. 在編輯器空間中，撰寫您希望針對工作區資源執行的查詢。

```
1 SELECT ahu, vav, r FROM EntityGraph
2 MATCH (vav)-[:feed]-(ahu)
3 WHERE vav.entityName LIKE 'vav_%'
```

Run Clear Ln: 3, Col: 34

Visual graph Query results Summary

在顯示的範例中，請求會搜尋名稱vav\_%中包含的實體，然後使用下列程式碼依實體之間的feed關係來組織這些實體。

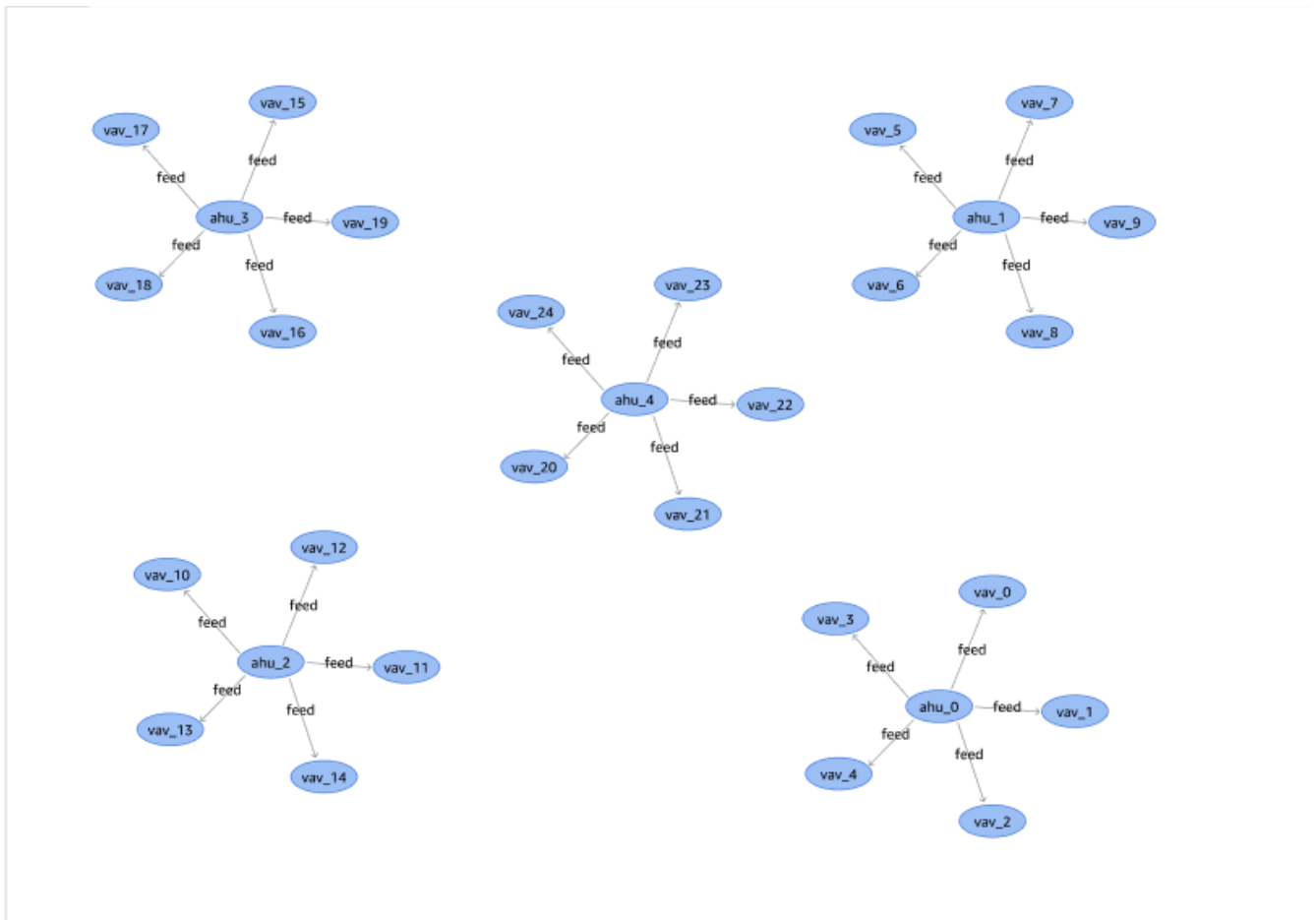
```
SELECT ahu, vav, r FROM EntityGraph
MATCH (vav)-[:feed]-(ahu)
WHERE vav.entityName LIKE 'vav_%'
```

#### Note

知識圖語法使用 [PartiQL](#)。如需有關此語法的資訊，請參閱[AWS IoT TwinMaker 知識圖表其他資源](#)。

3. 選擇「執行查詢」以執行您建立的請求。

系統會根據您的要求產生圖表。



上面顯示的示例圖是基於步驟 2 中的查詢示例。

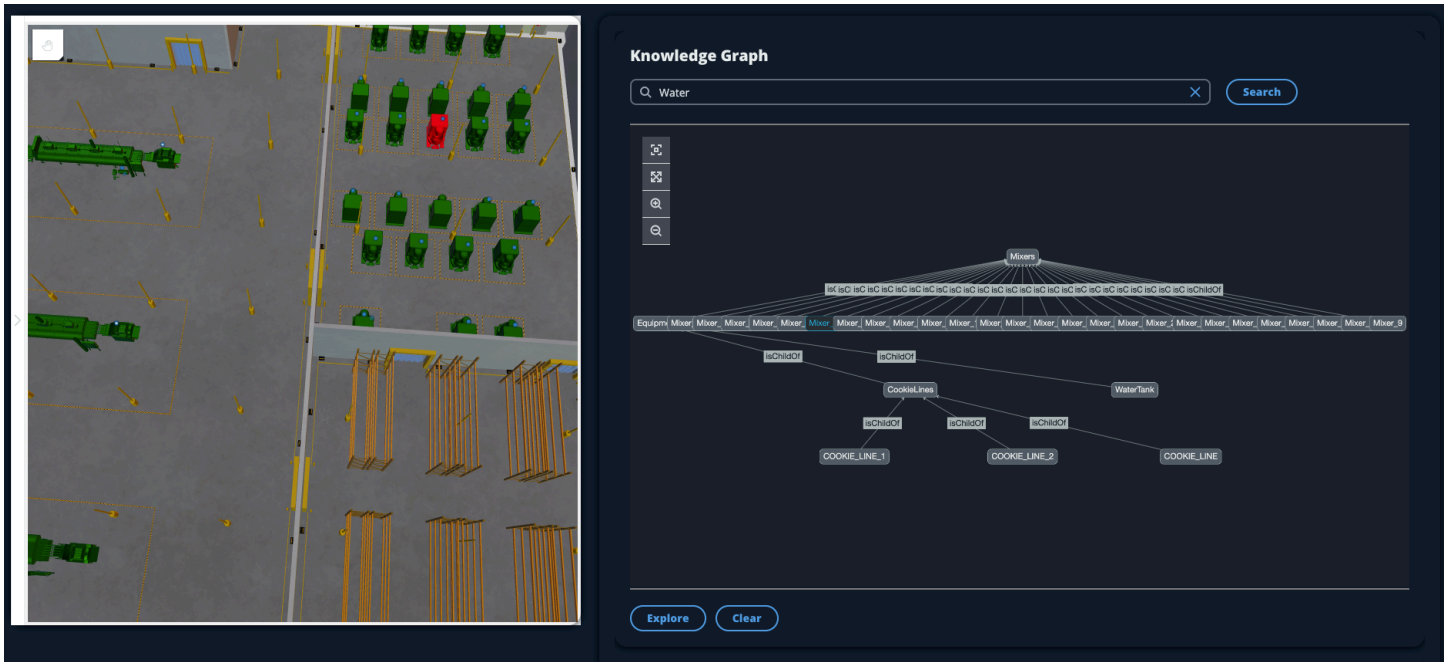
4. 查詢的結果也會顯示在清單中。選擇結果以檢視清單中的查詢結果。
5. 您也可以選擇「匯出為」，以 JSON 或 CSV 格式匯出查詢結果。

這涵蓋了控制台中知識圖的基本用途。如需展示知識圖語法的詳細資訊和範例，請參閱[AWS IoT TwinMaker 知識圖表其他資源](#)。

## 知識圖場景整合

您可以使用 AWS IoT 應用程式套件元件來建置將知識圖整合到您的 AWS IoT TwinMaker 場景中的 Web 應用程式。這可讓您根據場景中存在的 3D 節點 (代表您的設備或系統的 3D 模型) 產生圖形。若要建立繪製場景中 3D 節點圖形的應用程式，請先將 3D 節點繫結至工作區中的圖元。透過此對應，AWS IoT TwinMaker 繪製場景中出現的 3D 模型與工作區中實體之間的關係圖形。然後，您可以建立 Web 應用程式、使用場景選取 3D 模型，並以圖形格式探索它們與其他實體的關係。





有關利用 AWS IoT 用應用程序工具包組件在 AWS IoT TwinMaker 場景中生成圖形的工作 Web 應用程序的示例，請參閱 github 上的 [AWS IoT TwinMaker 示例 react 應用程序](#)。

## AWS IoT TwinMaker 場景圖先決條件

建立在場景中使用 AWS IoT TwinMaker 知識圖的 Web 應用程式之前，請先完成下列先決條件：

- 建立 AWS IoT TwinMaker 工作區。您可以在 [AWS IoT TwinMaker 主控台](#) 中建立工作區。
- 熟悉 AWS IoT TwinMaker 實體元件系統以及如何建立實體。如需詳細資訊，請參閱 [建立您的第一個實體](#)。
- 建立填入 3D 模型的 AWS IoT TwinMaker 場景。
- 熟悉 AWS IoT TwinMaker 的 AWS IoT 應用程式套件元件。如需 AWS IoT TwinMaker 元件的詳細資訊，請參閱 [使用 AWS IoT TwinMaker UI 組件創建自定義 Web 應用程序](#)。
- 通過知識圖概念和關鍵術語變得非凡。請參閱 [AWS IoT TwinMaker 知識圖核心概念](#)。

### Note

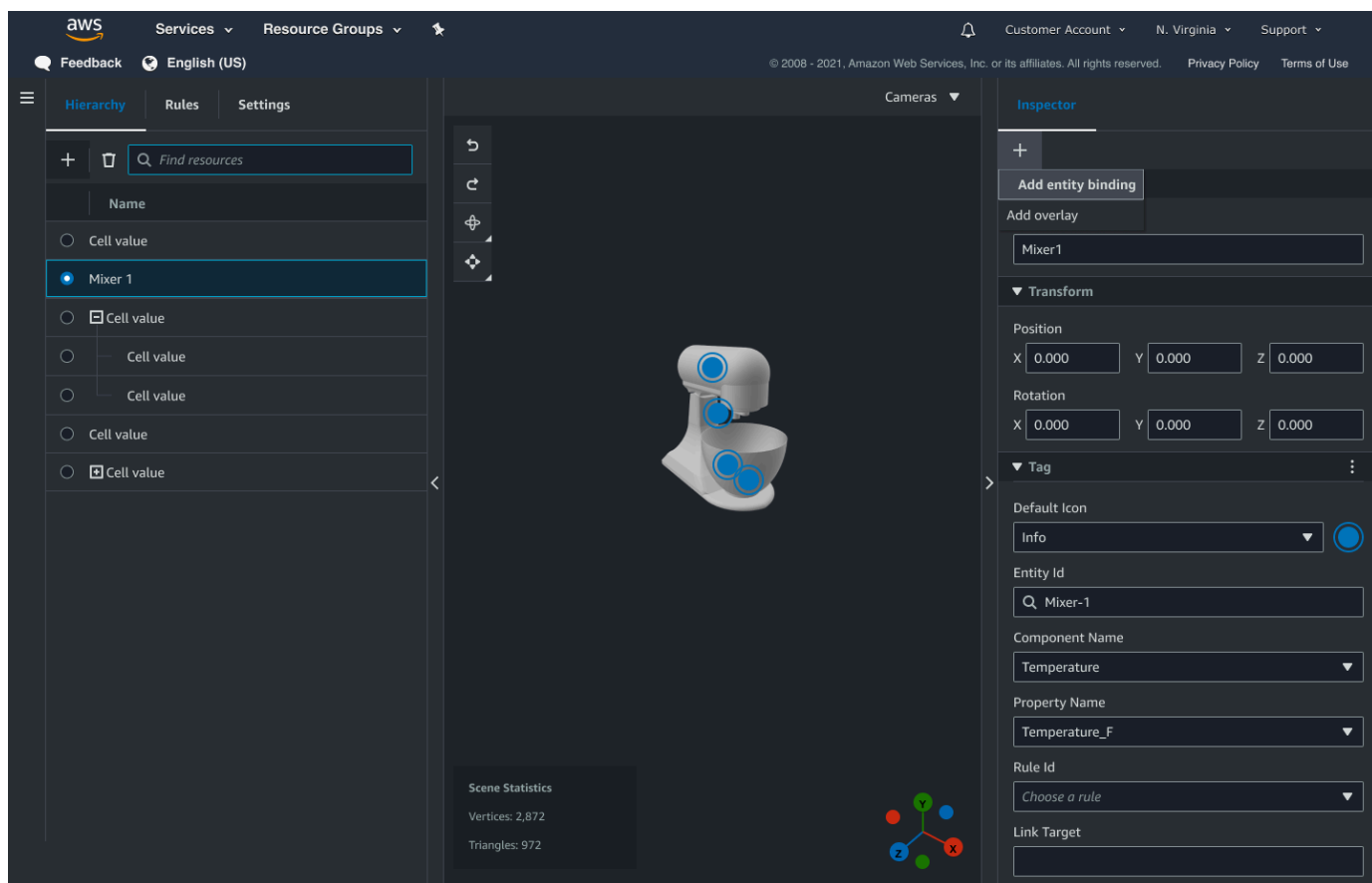
若要使用 AWS IoT TwinMaker 知識圖表 and 任何相關功能，您必須使用標準或分層組合定價模式。如需 AWS IoT TwinMaker 定價的詳細資訊，請參閱 [切換 AWS IoT TwinMaker 定價模式](#)。

## 在場景中綁定 3D 節點

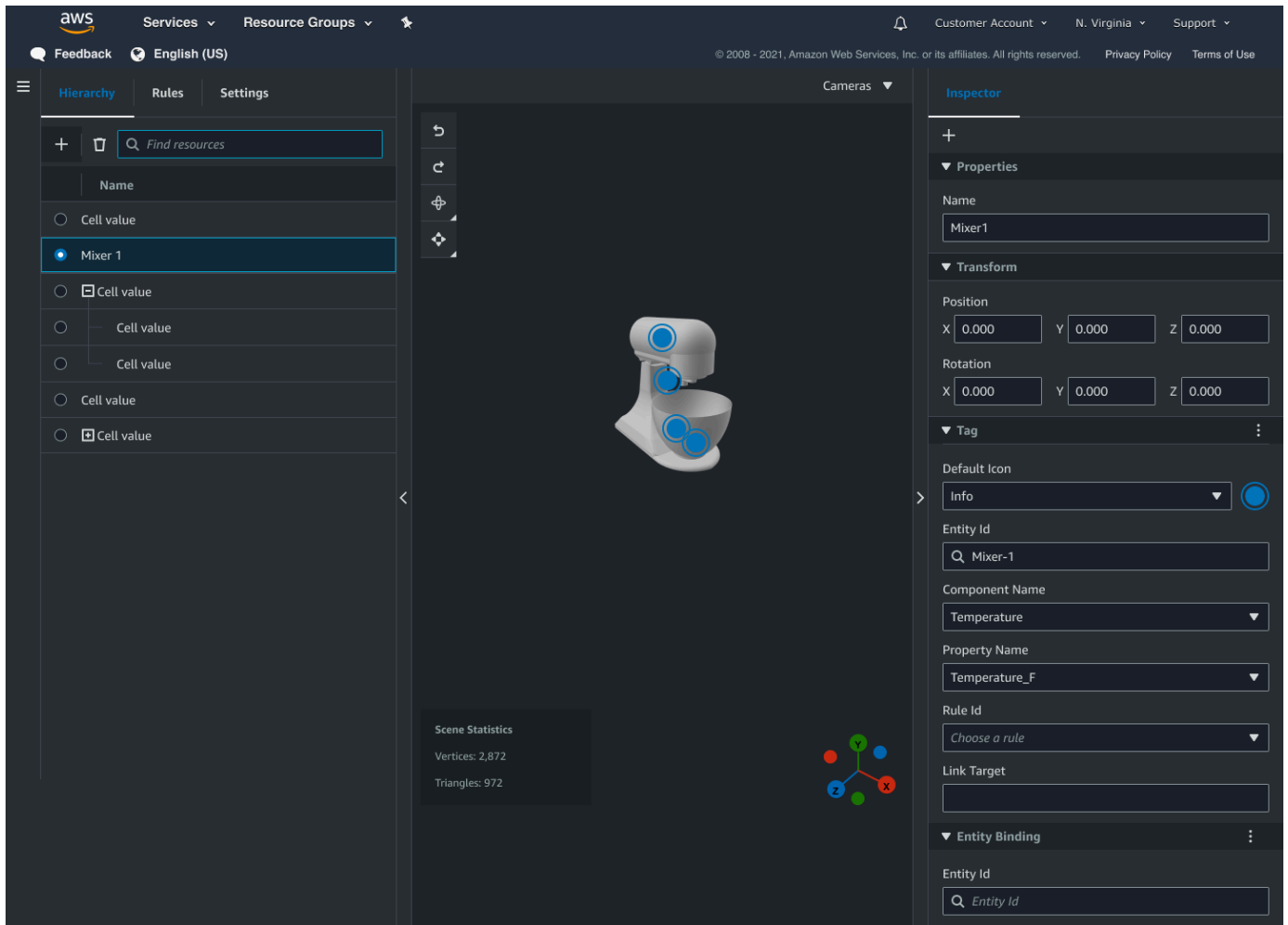
在建立將知識圖與場景整合的 Web 應用程式之前，請先將場景中存在的 3D 模型 (稱為 3D 節點) 繫結至相關聯的工作區實體。例如，如果您在場景中有混音器設備的模型，而對應的實體稱為 mixer\_0，請在混音器模型與代表混音器的實體之間建立資料繫結，以便繪製模型和實體。

若要執行資料繫結動作

1. 登入 [AWS IoT TwinMaker 主控台](#)。
2. 開啟您的工作區，然後選取包含您要繫結的 3D 節點的場景。
3. 在場景管理器中選取節點 (3D 模型)。當您選擇一個節點時，它將打開屏幕右側的檢查器面板。
4. 在「檢閱器」面板中，瀏覽至面板頂端，然後選取「+」按鈕。然後選擇新增實體繫結選項。這會開啟一個下拉式清單，您可以在其中選取要繫結至目前選取的節點的實體。



5. 從資料繫結下拉式功能表中，選取要對映至 3D 模型的實體 ID。在 [元件名稱] 和 [內容名稱] 欄位中，選取要繫結的元件和屬性。



選取「實體 ID」、「元件名稱」和「屬性名稱」欄位後，繫結即完成。

6. 對您要繪製圖形的所有模型和圖元重複此程序。

#### Note

可以在場景標籤上執行相同的數據綁定操作，只需選擇標籤而不是實體，然後按照相同的過程將標籤綁定到節點即可。

## 建立網路應用程式

繫結實體之後，請使用 AWS IoT 應用程式套件程式庫建置具有知識圖 Widget 的 Web 應用程式，讓您檢視場景並探索場景節點與實體之間的關係。

使用下列資源建立您自己的應用程式：

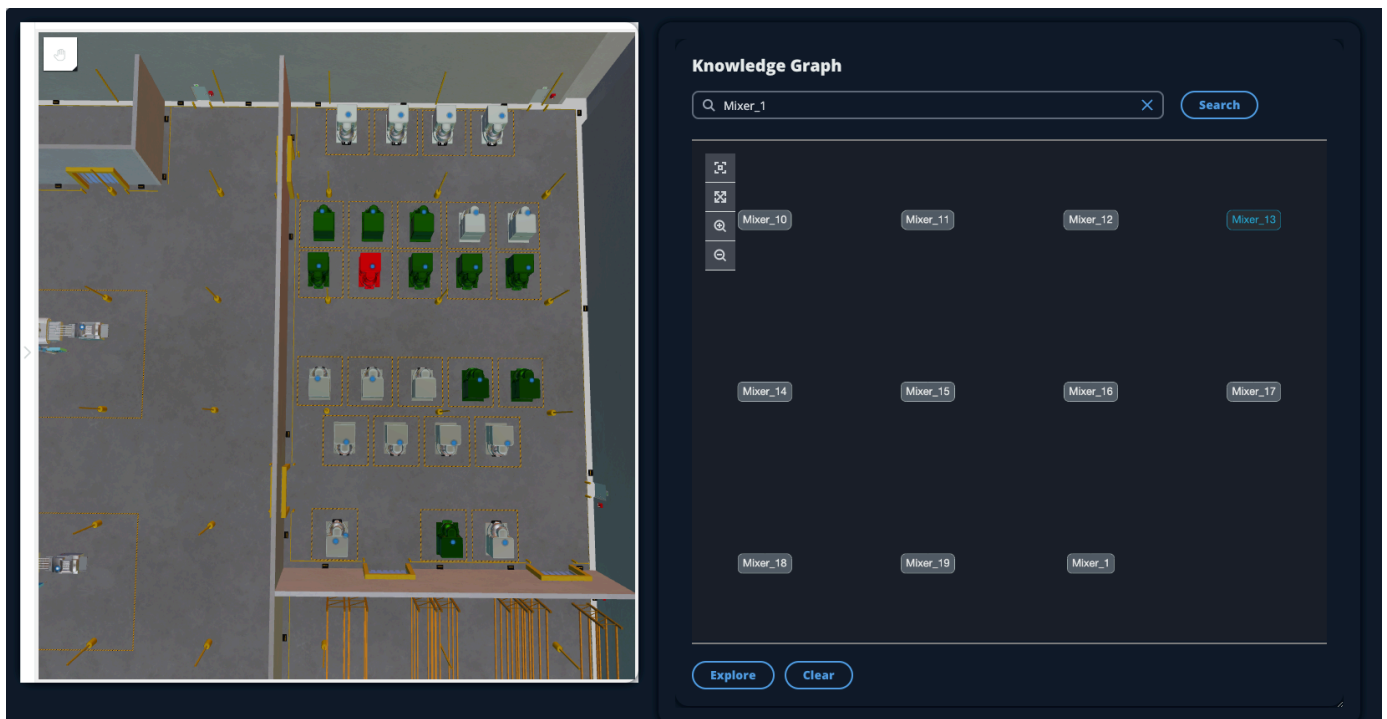
- 示 AWS IoT TwinMaker 例反應應用程式 github [自述](#)文檔。
- 該 AWS IoT TwinMaker 示例在 github 上反應應用程式[源代碼](#)。
- AWS IoT 應用程式套件[入門](#)文件。
- 該 AWS IoT 應用程式套件[視頻播放器組件](#)文檔。
- AWS IoT 應用程式套件[場景查看器組件](#)文檔。

下列程序示範 Web 應用程式中場景檢視器元件的功能。

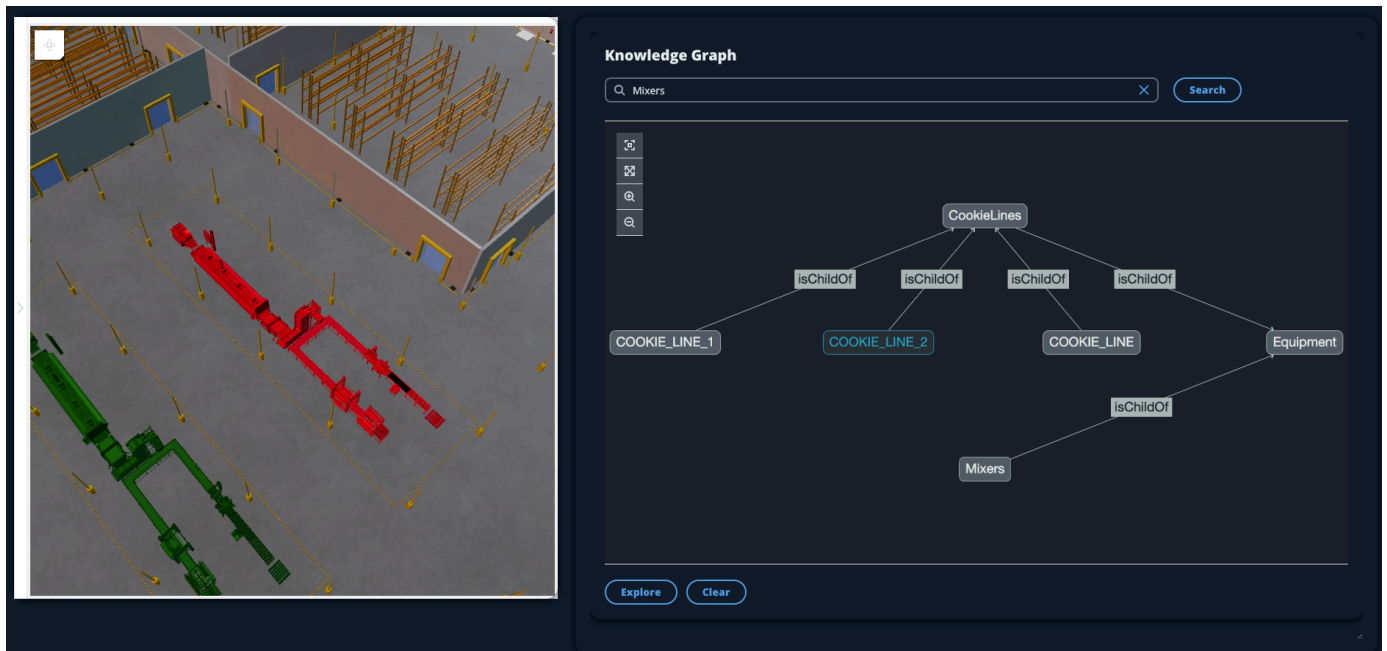
### Note

此過程基於 AWS IoT TwinMaker 示例反應應 AWS IoT 用程序中應用程式套件場景查看器組件的實現。

1. 打開示 AWS IoT TwinMaker 例反應應用程式的場景查看器組件。在搜尋欄位中輸入實體名稱或部分實體名稱 (區分大小寫的搜尋)，然後選取「搜尋」按鈕。如果一個模型綁定到實體 ID，那麼場景中的模型將被高亮顯示，實體的節點將顯示在場景查看器面板中。



2. 若要產生所有關係的圖形，請在場景檢視器 Widget 中選取一個節點，然後選取「瀏覽」按鈕。



3. 按「清除」(Clear) 按鈕清除目前的圖表選取並重新開始。

## 如何使用 AWS IoT TwinMaker 知識圖與 Grafana

本節說明如何將查詢編輯器面板新增至 AWS IoT TwinMaker Grafana 儀表板以執行和顯示查詢。

### AWS IoT TwinMaker 查詢編輯器必備

在 Grafana 中使用 AWS IoT TwinMaker 知識圖之前，請先完成下列先決條件：

- 建立 AWS IoT TwinMaker 工作區。您可以在[AWS IoT TwinMaker 主控台](#)中建立工作區。
- 配置 AWS IoT TwinMaker 為與 Grafana 一起使用。如需說明，請參閱[AWS IoT TwinMaker Grafana 板集成](#)。

#### **i** Note

若要使用 AWS IoT TwinMaker 知識圖表，您必須使用標準或分層組合定價模式。如需詳細資訊，請參閱 [切換 AWS IoT TwinMaker 定價模式](#)。

## AWS IoT TwinMaker 查詢編輯器權限

若要在 Grafana 中使用 AWS IoT TwinMaker 查詢編輯器，您必須擁有具有動作權限的 IAM 角色。iottwinmaker:ExecuteQuery 將該權限新增至您的工作區儀表板角色，如下列範例所示：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "{s3Arn}",
        "{s3Arn}/"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get",
        "iottwinmaker:List",
        "iottwinmaker:ExecuteQuery"
      ],
      "Resource": [
        "{workspaceArn}",
        "{workspaceArn}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

**Note**

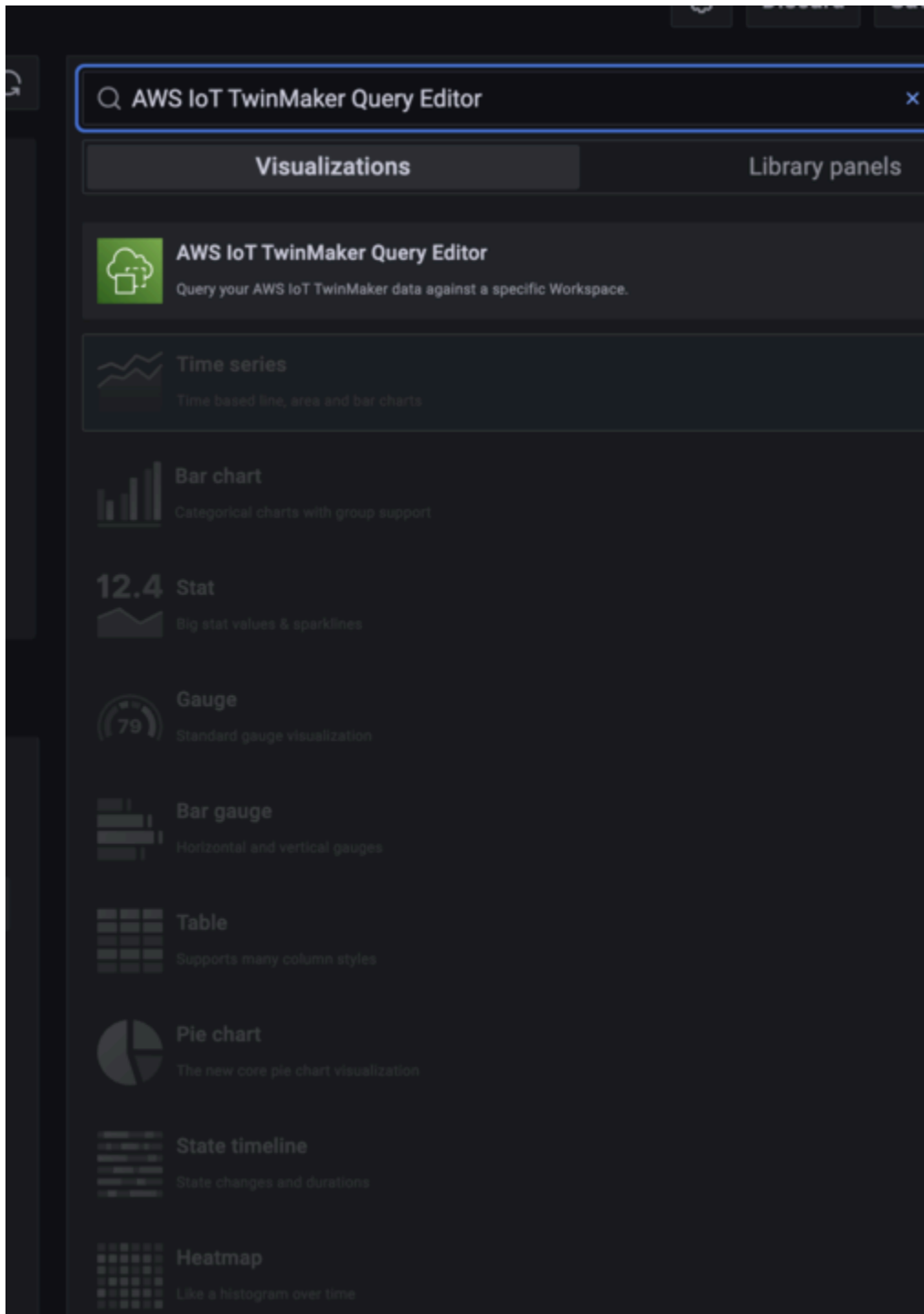
設定 AWS IoT TwinMaker Grafana 資料來源時，請務必在假設角色 ARN 欄位中使用具有此權限的角色。新增工作區後，您可以從「工作區」旁邊的下拉式清單中選取您的工作區。

如需詳細資訊，請參閱 [建立儀表板 IAM 角色](#)。

## 設定查 AWS IoT TwinMaker 詢編輯器面板

若要為知識圖表設定新的 Grafana 管控面板

1. 開啟您的 AWS IoT TwinMaker Grafana 儀表板。
2. 建立新管控面板。如需如何建立面板的詳細步驟，請參閱 Grafana 文件中的[建立控制面板](#)。
3. 從視覺效果清單中，選取AWS IoT TwinMaker 查詢編輯器。



4. 選取要執行查詢的資料來源。
5. (選擇性) 在提供的欄位中新增新面板的名稱。
6. 選取「套用」以儲存並確認您的新面板。

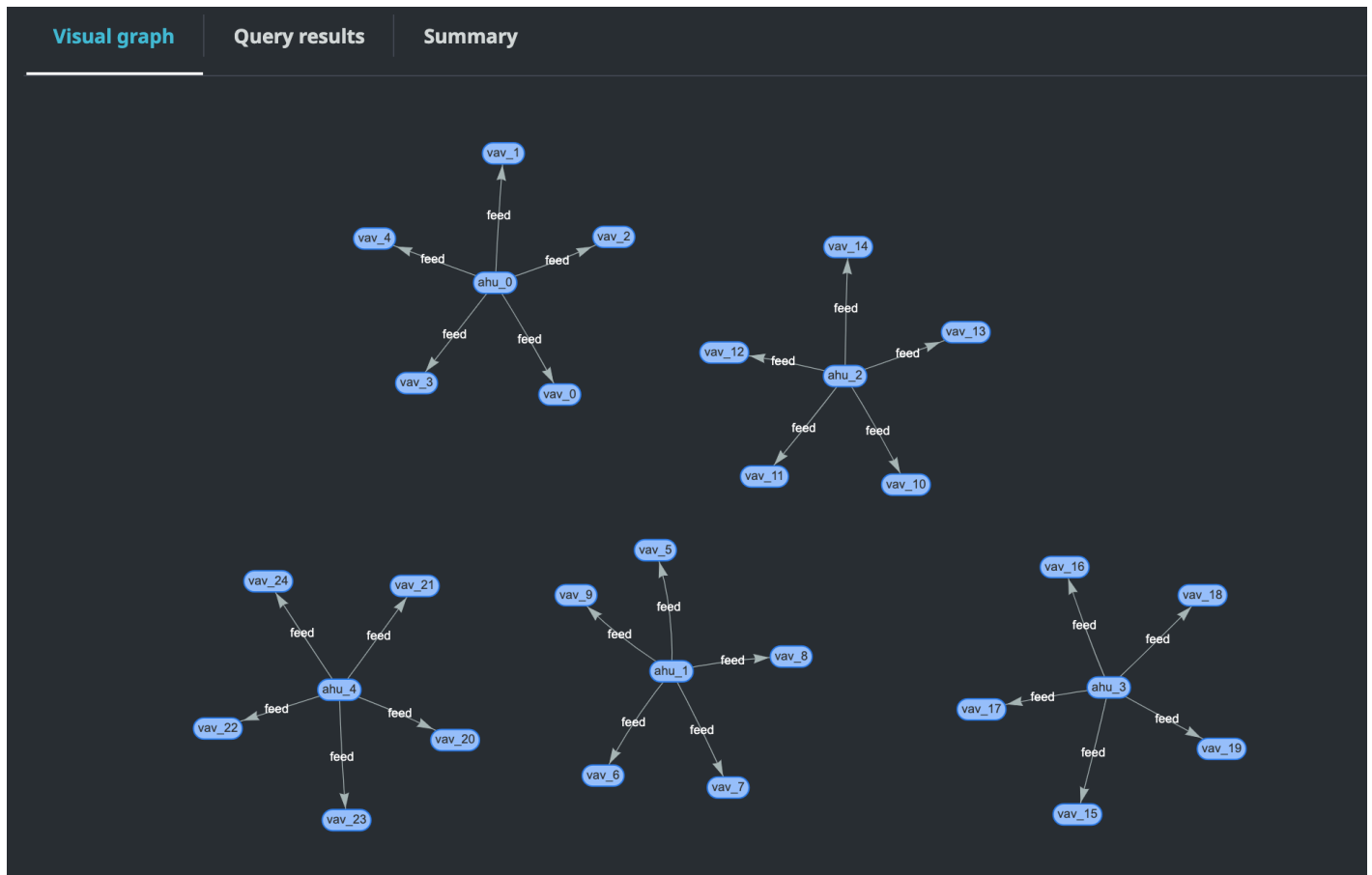


知識圖面板的運作方式與 AWS IoT TwinMaker 主控台中提供的查詢編輯器類似。您可以執行、撰寫和清除在面板中進行的查詢。如需如何撰寫查詢的詳細資訊，請參閱 [〈〉 AWS IoT TwinMaker 知識圖表 其他資源](#)。

### 如何使用 AWS IoT TwinMaker 查詢編輯器

查詢結果會以三種方式顯示，如下圖所示：以圖形視覺化方式顯示、列在表格中，或以執行摘要顯示。

- 圖形可視化：



視覺化圖形只會顯示結果中至少有一個關係的查詢資料。圖形會將圖元顯示為節點，並將關係顯示為圖形中的正向邊。

- 表格數據：

Visual graph | **Query results** | Summary

Results returned (25) Export as ▾

Q Search query results < 1 > ⚙

ahu ▲ | vav ▼ | r ▼

```
{
  "arn": "arn:aws:iottwinmaker:us-east-1:086801877023:workspace/SmartBuilding/entity/vav_66461816-02ab-355f-afd2-62a2cc92d336",
  "creationDate": 1667895664133,
  "entityId": "vav_66461816-02ab-355f-afd2-62a2cc92d336",
  "entityName": "vav_2",
  "lastUpdateDate": 1667895665269,
  "workspaceId": "SmartBuilding",
  "description": "",
  "components": [
    {
      "componentName": "VavComponent",
      "componentTypeId": "com.example.query.equipment.vav",
      "properties": [
        {
          "propertyName": "airTerminalUnitCertificates",
          "propertyValue": [ "AHRI", "UL" ] },
        {
          "propertyName": "airTerminalUnitBranchCount",
          "propertyValue": 2 },
        {
          "propertyName": "airTerminalUnitDimension",
          "propertyValue": {
            "width": 15, "length": 30, "height": 15
          }
        }
      ]
    }
  ]
  "relationshipName": "feed",
  "sourceEntityId": "ahu_23565bbb-3ec6-3ca0-9fce-e9b0cfeae7b1",
  "targetEntityId": "vav_66461816-02ab-355f-afd2-62a2cc92d336",
  "sourceComponentName": "AhuComponent",
  "sourceComponentTypeId": "com.example.query.equipment.ahu"
}
```

表格式資料格式會顯示所有查詢的資料。您可以在表格中搜尋特定結果或結果子集。資料可以以 JSON 或 CSV 格式匯出。

- 執行摘要

Visual graph	Query results	<b>Summary</b>		
Start	Status	Response	Statement	Duration
2022-11-15 11:36:08 UTC-0800	Success	25 returned	SELECT ahu, vav, r FROM EntityGraph MATCH (vav)<-[r:feed]->(ahu) WHERE vav.entityName LIKE 'vav_%'	0.833 sec

執行摘要會顯示有關查詢狀態的查詢和中繼資料。

## AWS IoT TwinMaker 知識圖表其他資源

本節提供用於在知識圖表中撰寫查詢的 PartiQL 語法的基本範例，以及提供知識圖資料模型相關資訊的 PartiQL 文件連結。

- [PartiQL 圖形資料模型文件](#)
- [PartiQL 圖形查詢文檔](#)

這組示例顯示了基本查詢及其響應。使用此作為編寫自己的查詢的參考。

### 基本查詢

- 使用過濾器獲取所有實體

```
SELECT entity
FROM EntityGraph MATCH (entity)
WHERE entity.entityName = 'room_0'
```

此查詢會傳回名為工作區中的所有實體room\_0。

FROM clause: EntityGraph 是包含工作區中所有實體及其關係的圖形集合。系統會 AWS IoT TwinMaker 根據工作區中的實體自動建立和管理此集合。

MATCH子句：指定與圖形的一部分匹配的模式。在這種情況下，模式(entity)匹配圖中的每個節點，並綁定到實體變量。該FROM條款之後必須接有MATCH條款。

WHERE clause：在節點的entityName字段上指定一個過濾器，其中的值必須匹配room\_0。

SELECT子句：指定entity變量，以便返回整個實體節點。

回應：

```
{
  "columnDescriptions": [
    {
      "name": "entity",
      "type": "NODE"
    }
  ],
  "rows": [
    {
      "rowData": [
        {
          "arn": "arn:aws:iottwinmaker:us-east-1: 577476956029: workspace /
SmartBuilding8292022 / entity / room_18f3ef90 - 7197 - 53 d1 - abab -
db9c9ad02781 ",
          "creationDate": 1661811123914,
```

```
"entityId": "room_18f3ef90-7197-53d1-abab-db9c9ad02781",
"entityName": "room_0",
"lastUpdateDate": 1661811125072,
"workspaceId": "SmartBuilding8292022",
"description": "",
"components": [
  {
    "componentName": "RoomComponent",
    "componentTypeId": "com.example.query.construction.room",
    "properties": [
      {
        "propertyName": "roomFunction",
        "propertyValue": "meeting"
      },
      {
        "propertyName": "roomNumber",
        "propertyValue": 0
      }
    ]
  }
]
```

會 `columnDescriptions` 傳回有關資料行的中繼資料，例如名稱和類型。返回的類型是 `NODE`。這表明整個節點已被返回。該類型的其他值可以是 `EDGE` 指示關係 `VALUE`，或者表示純量值，例如整數或字符串。

會 `rows` 傳回列清單。由於只有一個實體匹配，所以返回一個 `rowData` 包含實體中的所有字段。

#### Note

與 SQL 不同，您只能返回標量值，您可以使用 PartiQL 返回一個對象（作為 JSON）。

每個節點都包含所有實體層級欄位，例如 `entityId`、`arn` 和 `components`、元件層級欄位（例如 `componentTypeId` 和 `properties`）`componentName`，以及屬性層級欄位（例如 `propertyName` 和 `propertyValue`）都以巢狀 JSON 形式。

- 使用過濾器獲取所有關係：

```
SELECT relationship
FROM EntityGraph MATCH (e1)-[relationship]->(e2)
WHERE relationship.relationshipName = 'isLocationOf'
```

此查詢返回與關係名稱的工作區中的所有關係isLocationOf。

子MATCH句：指定匹配兩個節點（由表示()）的模式，這些節點由有向邊（表示由表示-[]->）連接並綁定到調用的變量relationship。

該WHERE子句：指定邊緣，其中的值是relationshipName字段的過濾器isLocationOf。

子SELECT句：指定關係變量，以便返回整個邊緣節點。

回應

```
{
  "columnDescriptions": [{
    "name": "relationship",
    "type": "EDGE"
  }],
  "rows": [{
    "rowData": [{
      "relationshipName": "isLocationOf",
      "sourceEntityId": "floor_83faea7a-ea3b-56b7-8e22-562f0cf90c5a",
      "targetEntityId": "building_4ec7f9e9-e67e-543f-9d1b-235df7e3f6a8",
      "sourceComponentName": "FloorComponent",
      "sourceComponentTypeId": "com.example.query.construction.floor"
    }]
  },
  ... //rest of the rows are omitted
]
}
```

中的欄類型columnDescriptions是一個EDGE。

每個rowData代表一個邊緣與字段類似relationshipName。這與實體上定義的關聯屬性名稱相同。sourceEntityId, sourceComponentName並sourceComponentTypeId提供有關在哪個實體和元件上定義關係屬性的資訊。targetEntityId指定此關係指向哪個實體。

- 取得與特定實體具有特定關係的所有實體

```
SELECT e2.entityName
FROM EntityGraph MATCH (e1)-[r]->(e2)
WHERE relationship.relationshipName = 'isLocationOf'
AND e1.entityName = 'room_0'
```

這個查詢會傳回與實體有isLocationOf關係的所有實體的所有room\_0實體名稱。

子MATCH句：指定匹配任何兩個節點 ( e1 , e2 ) 具有有向邊 ( r ) 的模式。

子WHERE句：指定關聯名稱和來源實體名稱的篩選器。

子SELECT句：返回e2節點中的entityName字段。

回應

```
{
  "columnDescriptions": [
    {
      "name": "entityName",
      "type": "VALUE"
    }
  ],
  "rows": [
    {
      "rowData": [
        "floor_0"
      ]
    }
  ]
}
```

在列說明中，列的類型entityName是VALUE一個字符串。

傳回一個實體floor\_0，。

比賽

MATCH子句支援下列模式：

- 匹配節點 'b' 指向節點 'a'：

```
FROM EntityGraph MATCH (a)-[rel]-(b)
```

- 匹配節點 'a' 指向節點 'b' :

```
FROM EntityGraph MATCH (a)-[]->(b)
```

假設不需要在關係上指定過濾器，則沒有綁定到關係的變量。

- 匹配節點 'a' 指向節點 'b' 和節點 'b' 指向節點 'a' :

```
FROM EntityGraph MATCH (a)-[rel]-(b)
```

這將返回兩個匹配：一個從 'a' 到 'b'，另一個從 'b' 到 'a'，所以建議是盡可能使用有向邊。

- 關係名稱也是屬性圖形的標籤 EntityGraph，因此您可以簡單地在冒號 (:) 之後指定關係名稱，而不是在 WHERE 子句 `rel.relationshipName` 中指定過濾器。

```
FROM EntityGraph MATCH (a)-[:isLocationOf]-(b)
```

- 鏈接：可以鏈接模式以匹配多個關係。

```
FROM EntityGraph MATCH (a)-[rel1]->(b)-[rel2]-(c)
```

- 可變躍點模式也可以跨越多個節點和邊緣：

```
FROM EntityGraph MATCH (a)-[]->{1,5}(b)
```

這個查詢與傳出邊緣從 1 到 5 跳內節點 'a' 的任何模式匹配。允許的數量詞如下：

`{m,n}`-在 m 和 n 重複之間

`{m,}`-m 或更多重複。

從：

實體節點可以包含巢狀資料，例如其本身包含更多巢狀資料 (例如屬性) 的元件。這些可以通過取消嵌套 MATCH 模式的結果來訪問。

```
SELECT e
FROM EntityGraph MATCH (e), e.components AS c, c.properties AS p
WHERE c.componentTypeId = 'com.example.query.construction.room',
AND p.propertyName = 'roomFunction'
AND p.propertyValue = 'meeting'
```

通過點.入一個變量訪問嵌套字段。逗號(,)用於取消嵌套(或連接)實體與內部的元件，然後取消這些元件內的屬性。AS用來將變數繫結至非巢狀變數，以便在WHERE或SELECT子句中使用它們。此查詢返回包含與組件類型 ID 的組件meeting中roomFunction的 value 命名屬性的所有實體  
com.example.query.construction.room

若要存取欄位的多個巢狀欄位(例如實體中的多個元件)，請使用逗號標記法執行聯結。

```
SELECT e
FROM EntityGraph MATCH (e), e.components AS c1, e.components AS c2
```

選擇：

- 返回一個節點：

```
SELECT e
FROM EntityGraph MATCH (e)
```

- 返回一個邊緣：

```
SELECT r
FROM EntityGraph MATCH (e1)-[r]->(e2)
```

- 返回一個標量值：

```
SELECT floor.entityName, room.description, p.propertyValue AS roomfunction
FROM EntityGraph MATCH (floor)-[:isLocationOf]-(room),
room.components AS c, c.properties AS p
```

使AS用別名來格式化輸出欄位的名稱。在這裡，而不是propertyValue作為響應中的列名roomfunction，返回。

- 返回別名：

```
SELECT floor.entityName AS floorName, luminaire.entityName as luminaireName
FROM EntityGraph MATCH (floor)-[:isLocationOf]-(room)-[:hasPart]-
(lightningZone)-[:feed]-(luminaire)
WHERE floor.entityName = 'floor_0'
AND luminaire.entityName like 'lumin%'
```

強烈建議使用別名是明確的，提高可讀性，並避免查詢中的任何歧義。



哪裡：

- 支援的邏輯運算子為ANDNOT、和OR。
- 支援的比較運算子為<<=>、=>、=、和!=。
- 如果您要在相同欄位上指定多個OR條件，請使用IN關鍵字。
- 在實體、元件或屬性欄位上篩選：

```
FROM EntityGraph MATCH (e), e.components AS c, c.properties AS p
WHERE e.entityName = 'room_0'
AND c.componentTypeId = 'com.example.query.construction.room',
AND p.propertyName = 'roomFunction'
AND NOT p.propertyValue = 'meeting'
OR p.propertyValue = 'office'
```

- 篩選屬configuration性。這unit是配置映射中的關鍵字，並且Celsius是值。

```
WHERE p.definition.configuration.unit = 'Celsius'
```

- 檢查映射屬性是否包含給定的鍵和值：

```
WHERE p.propertyValue.length = 20.0
```

- 檢查映射屬性是否包含給定的鍵：

```
WHERE NOT p.propertyValue.length IS MISSING
```

- 檢查列表屬性是否包含給定值：

```
WHERE 10.0 IN p.propertyValue
```

- 使用lower()函數進行不區分大小寫的比較。依預設，所有比較都區分大小寫。

```
WHERE lower(p.propertyValue) = 'meeting'
```

喜歡：

如果您不知道某個字段的確切值，並且可以在指定的字段上執行全文本搜索，則非常有用。%代表零個或更多。

```
WHERE e.entityName LIKE '%room%'
```

- 中綴搜索：%room%
- 前綴搜索：room%
- 後綴搜索：%room
- 如果您的值中有 '%'，請在中輸入逸出字元，LIKE並使用指定逸出字元ESCAPE。

```
WHERE e.entityName LIKE 'room\%' ESCAPE '\'
```

不同的：

```
SELECT DISTINCT c.componentTypeId
FROM EntityGraph MATCH (e), e.components AS c
```

- DISTINCT關鍵字會從最終結果中消除重複項目。

DISTINCT複雜資料類型不支援。

伯爵

```
SELECT COUNT(e), COUNT(c.componentTypeId)
FROM EntityGraph MATCH (e), e.components AS c
```

- 關COUNT鍵字會計算查詢結果中的項目數。
- COUNT巢狀複雜欄位和圖形模式欄位不支援。
- COUNTDISTINCT和巢狀查詢不支援彙總。

例如，不支援 COUNT(DISTINCT e.entityId)。

路徑

使用路徑投影進行查詢時支援下列陣列投影：

- 變數躍點查詢

```
SELECT p FROM EntityGraph MATCH p = (a)-[]->{1, 3}(b)
```

這個查詢匹配和投射任何模式的節點元數據與傳出邊緣從節點 a 內 1 到 3 跳。

- 固定跳查詢

```
SELECT p FROM EntityGraph MATCH p = (a)-[]->(b)<-[]-(c)
```

這個查詢匹配和項目實體的元數據和傳入邊緣到 b。

- 非定向查詢

```
SELECT p FROM EntityGraph MATCH p = (a)-[]-(b)-[]-(c)
```

此查詢匹配和投影節點的元數據 1 跳模式通過 b 連接 a 和 c。

```
{
  "columnDescriptions": [
    {
      "name": "path",
      "type": "PATH"
    }
  ],
  "rows": [
    {
      "rowData": [
        {
          "path": [
            {
              "entityId": "a",
              "entityName": "a"
            },
            {
              "relationshipName": "a-to-b-relation",
              "sourceEntityId": "a",
              "targetEntityId": "b"
            },
            {
              "entityId": "b",
              "entityName": "b"
            }
          ]
        }
      ]
    },
    {
      "rowData": [
        {
          "path": [
            {

```

```

    "entityId": "b",
    "entityName": "b"
  },
  {
    "relationshipName": "b-to-c-relation",
    "sourceEntityId": "b",
    "targetEntityId": "c"
  },
  {
    "entityId": "c",
    "entityName": "c"
  }
]
}
]
}
]
}
}

```

此 *PATH* 查詢響應包括僅元數據，該元數據通過 *b* 標識 *a* 和 *c* 之間的每個路徑/模式的所有節點和邊緣。

限制和偏移：

```

SELECT e.entityName
FROM EntityGraph MATCH (e)
WHERE e.entityName LIKE 'room_%'
LIMIT 10
OFFSET 5

```

*LIMIT* 指定要在查詢中傳回的結果數目，並 *OFFSET* 指定要略過的結果數目。

限制和 *maxResults*：

下列範例顯示總共傳回 500 個結果的查詢，但每次 API 呼叫一次只會顯示 50 個結果。您可以在需要限制顯示結果數量的地方使用此模式，例如，如果您只能在 UI 中顯示 50 個結果。

```

aws iottwinmaker execute-query \
--workspace-id exampleWorkspace \
--query-statement "SELECT e FROM EntityGraph MATCH (e) LIMIT 500"\
--max-results 50

```

- LIMIT關鍵字會影響查詢並限制產生的資料列。如果您需要控制每個 API 呼叫傳回的結果數，而不限制傳回結果的總數，請使用LIMIT。
- max-results是 [ExecuteQuery API 動作](#)的選用參數。max-results僅適用於 API 以及如何在上述查詢的範圍內讀取結果。

在查詢max-results中使用可減少顯示結果的數量，而不會限制傳回結果的實際數量。

下面的查詢遍歷結果的下一頁。此查詢使用 ExecuteQuery API 呼叫傳回列 51-100，其中下一頁的結果由指定 next-token — 在此情況下，權杖為：。"H7kyGmvK376L"

```
aws iottwinmaker execute-query \
--workspace-id exampleWorkspace \
--query-statement "SELECT e FROM EntityGraph MATCH (e) LIMIT 500"\
--max-results 50
--next-token "H7kyGmvK376L"
```

- 字next-token串會指定結果的下一頁。如需詳細資訊，請參閱 [ExecuteQueryAPI 動作](#)。

AWS IoT TwinMaker 知識圖查詢有以下限制：

限制名稱	配額	可調整
查詢執行逾時	10 秒	否
躍點數目上限	10	是
自我JOIN的最大數量	20	是
投影欄位的最大數目	20	是
條件運算式的最大數目 (ANDOR、 、 NOT)	10	是
LIKE運算式模式的最大長度 (包括萬用字元和逸出字元)	20	是
可在IN子句中指定的項目數目 上限	10	是
最大值 OFFSET	3000	是

限制名稱	配額	可調整
最大值 LIMIT	3000	是
遍歷的最大值 (+) OFFSET LIMIT	3000	是

# 資產同步處理 AWS IoT SiteWise

AWS IoT TwinMaker 支援資產和資產模型的資產同步處理 (資 AWS IoT SiteWise 產同步)。使用 AWS IoT SiteWise 元件類型，資產同步會取得現有資 AWS IoT SiteWise 產和資產模型，並將這些資源轉換為 AWS IoT TwinMaker 實體、元件和元件類型。以下各節將逐步說明如何設定資產同步，以及哪些資 AWS IoT SiteWise 產和資產模型可以同步至您的 AWS IoT TwinMaker 工作區。

## 主題

- [使用資產同步 AWS IoT SiteWise](#)
- [自訂和預設工作區之間的差異](#)
- [資源同步來源 AWS IoT SiteWise](#)
- [分析同步狀態和錯誤](#)
- [刪除同步工作](#)
- [資產同步限制](#)

## 使用資產同步 AWS IoT SiteWise

本主題說明如何開啟和設定 AWS IoT SiteWise 資產同步。根據您使用的工作區類型，遵循適當的程序。

### Important

[the section called “自訂和預設工作區之間的差異”](#)如需有關自訂工作區和預設工作區之間差異的資訊，請參閱。

## 主題

- [使用自訂工作區](#)
- [使用 IoT SiteWiseDefaultWorkspace](#)

## 使用自訂工作區

開啟資產同步之前，請檢閱這些先決條件。

## 必要條件

使用前 AWS IoT SiteWise，請確保您已完成以下操作：

- 您有 AWS IoT TwinMaker 工作區。
- 您已在中建立了一些資產和資產模型 AWS IoT SiteWise。如需詳細資訊，請參閱[建立資產模型](#)。
- 您已建立 IAM 角色，該角色具有下列 AWS IoT SiteWise 動作的讀取權限：ListAssetsListAssetModelsDescribeAsset、和DescribeAssetModel。

IAM 角色也必須具有下列的寫入權限 AWS IoT

TwinMaker：CreateEntityUpdateEntityDeleteEntity、CreateComponentType、UpdateComponentType和DeleteComponentType。

您可以使用下列 IAM 角色做為必要角色的範本：

```
// trust relationships
{
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": [
            "iottwinmaker.amazonaws.com",
            "iotsitewise.amazonaws.com"
          ]
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
}

// permissions
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:*",
      "Resource": "*"
    },
  ],
}
```



```
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:Describe*",
    "iotsitewise:List*"
  ],
  "Resource": "*"
}
```

使用下列程序來開啟和設定 AWS IoT SiteWise 資產同步。

1. 在[AWS IoT TwinMaker 主控台](#)中，導覽至「設定」頁面。
2. 開啟「模型來源」頁籤。

**Settings**  
Settings for your TwinMaker account

Pricing options | **Model sources**

**AWS IoT SiteWise connector** Info Open AWS IoT SiteWise Connect workspace

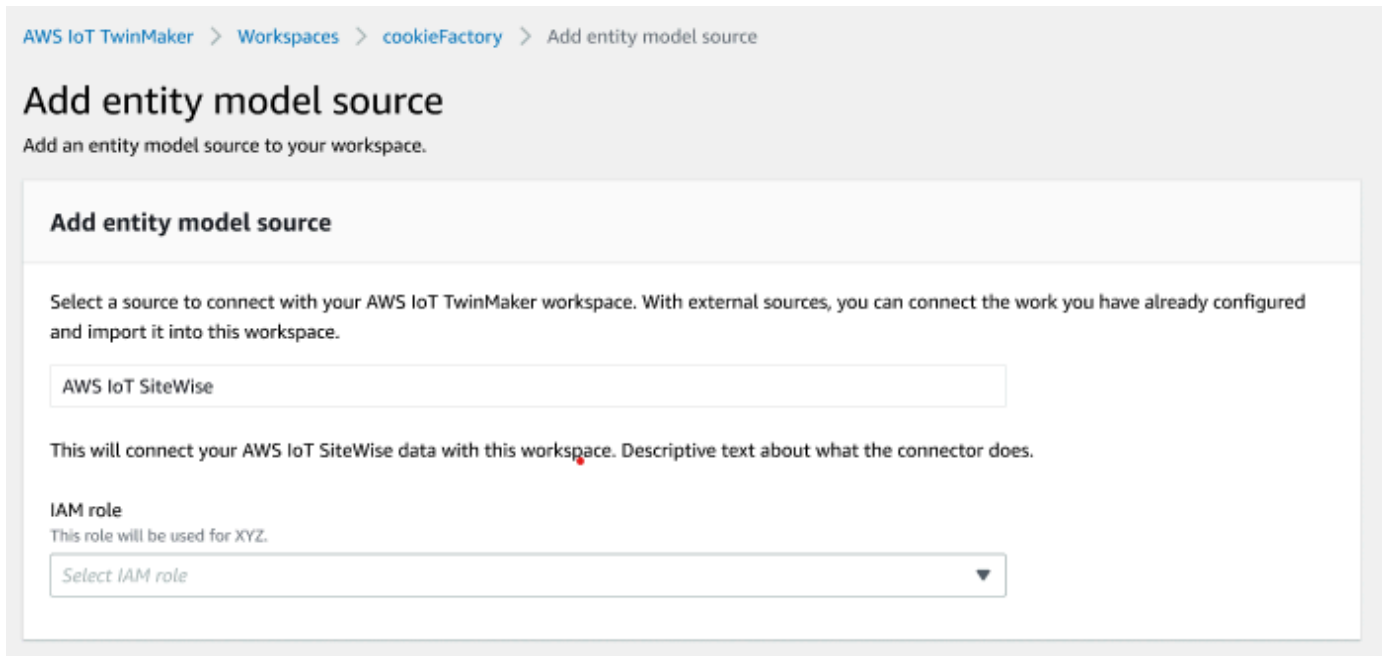
Connector status	Workspace	Last updated
⊖ Disabled	-	-

3. 選擇「Connect 工作區」，將 AWS IoT TwinMaker 工作區連結至資 AWS IoT SiteWise 產。

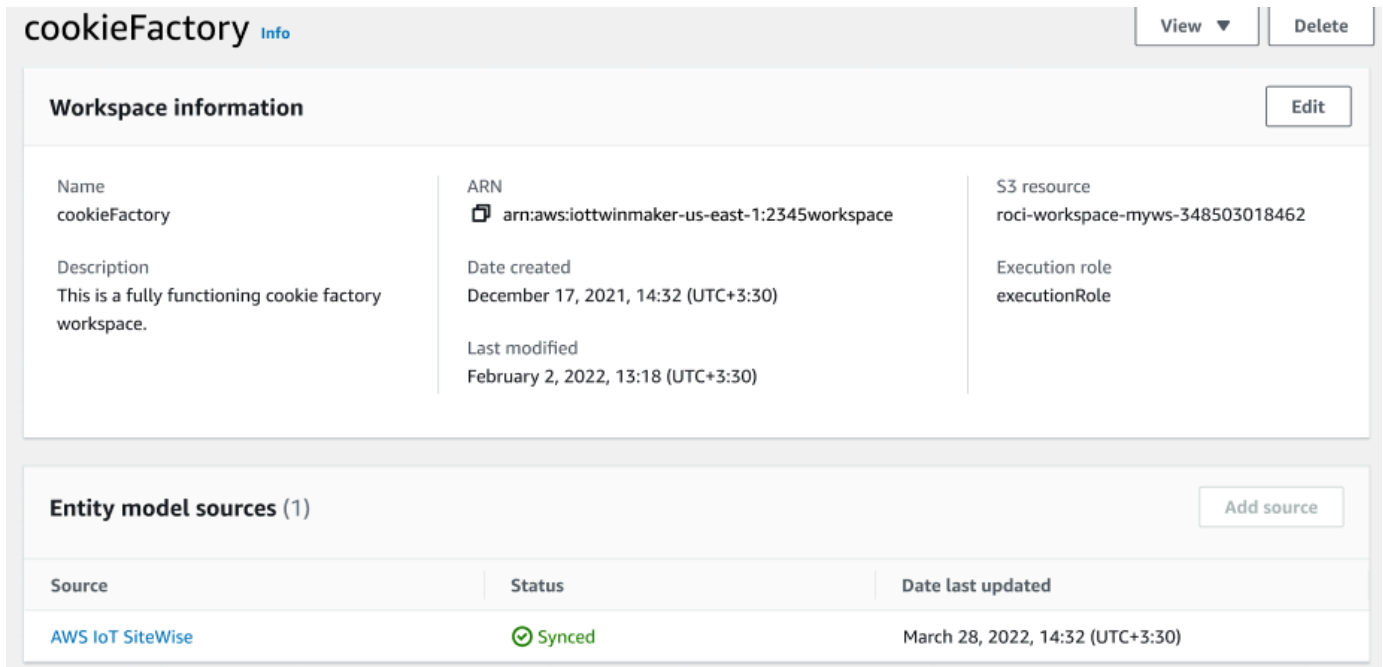
#### **Note**

您只能將資產同步與單一 AWS IoT TwinMaker 工作區搭配使用。如果您希望在不同的工作區中進行同步，則必須中斷與一個工作區的同步連接到另一個工作區。

4. 接下來，導覽至您要在其中使用資產同步的工作區。
5. 選擇 Add sources (新增來源)。這會開啟 [新增實體模型來源] 頁面。



- 在 [新增實體模型來源] 頁面上，確認來源欄位已顯示AWS IoT SiteWise。選取您建立的 IAM 角色做為 IAM 角色的先決條件。
- 您現在已開啟資 AWS IoT SiteWise 產同步。您應該會在選取的「工作區」頁面頂端看到一個構象橫幅，確認資產同步處於作用中狀態。您現在也應該會看到「實體模型來源」區段中列出的同步來源。



## 使用 IoT SiteWiseDefaultWorkspace

當您選擇加入[AWS IoT SiteWiseAWS IoT TwinMaker 整合](#)時，會建立名為IoTSiteWiseDefaultWorkspace為的預設工作區，並自動與同步 AWS IoT SiteWise。

您也可以使用 AWS IoT TwinMaker CreateWorkspace API 建立名為的工作區IoTSiteWiseDefaultWorkspace。

### 必要條件

在創建之前IoTSiteWiseDefaultWorkspace，請確保您已完成以下操作：

- 建立 AWS IoT TwinMaker 服務連結角色。如需詳細資訊，請參閱[使用 AWS IoT TwinMaker的服務連結角色](#)。
- 前往 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。

檢閱角色或使用者，並確認該角色或使用者具有的權限*iotsitewise:EnableSiteWiseIntegration*。

如有需要，請新增角色或使用者的權限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:EnableSiteWiseIntegration",
      "Resource": "*"
    }
  ]
}
```

## 自訂和預設工作區之間的差異

### Important

新 AWS IoT SiteWise 功能，例如 [CompositionModel](#)，僅在中提供IoTSiteWiseDefaultWorkspace。我們建議您使用預設工作區而非自訂工作區。

使用時IoTSiteWiseDefaultWorkspace，與使用具有資產同步的自訂工作區有一些明顯的差異。

- 建立預設工作區時，Amazon S3 位置和 IAM 角色是選擇性的。

#### Note

您可以使用UpdateWorkspace來提供 Amazon S3 位置和 IAM 角色。

- IoTSiteWiseDefaultWorkspace沒有可將資源同步到的資 AWS IoT SiteWise 源計數限制 AWS IoT TwinMaker。
- 當您從同步資源時 AWS IoT SiteWise，它們SyncSource將是SITEWISE\_MANAGED。這包括Entities和ComponentTypes。
- 新 AWS IoT SiteWise 功能，例CompositionModel如僅在中提供IoTSiteWiseDefaultWorkspace。

有一些特定的限制IoTSiteWiseDefaultWorkspace，它們是：

- 無法刪除預設工作區。
- 若要刪除資源，您必須先刪除 AWS IoT SiteWise 資源，然後刪除中 AWS IoT TwinMaker 的對應資源。

## 資源同步來源 AWS IoT SiteWise

本主題列出您可以從哪些資產同步 AWS IoT SiteWise 到 AWS IoT TwinMaker 工作區。

#### Important

[自訂和預設工作區之間的差異](#)如需有關自訂工作區和預設工作區之間差異的資訊，請參閱。

## 自訂和預設工作區

下列資源已同步，且可在自訂和預設工作區中使用：

### 資產模型

AWS IoT TwinMaker 會為中的每個資產模型建立新元件類型 AWS IoT SiteWise。

- 資產模型TypeId的元件將使用下列其中一種樣式：
  - 自訂工作區- `iotsitewise.assetmodel:assetModelId`
  - 預設工作區- `assetModelId`
- 資產模型中的每個性質都是元件類型中的新性質，具有以下命名樣式之一：
  - 自訂工作區- `Property_propertyId`
  - 預設工作區- `propertyId`

中的屬性名稱 AWS IoT SiteWise 會儲存為屬性定義displayName中的。

- 資產模型中的每個階層都是類型的新性質，LIST且nestedType位RELATIONSHIP於元件類型中。階層會對應至具有以下其中一個字首的名稱的屬性：
  - 自訂工作區- `Hierarchy_hierarchyId`
  - 預設工作區- `hierarchyId`

## 資產

AWS IoT TwinMaker 會為中的每個資產建立新實體 AWS IoT SiteWise。

- 與中的相assetId同 AWS IoT SiteWise。entityId
- 這些實體具有稱為的單一元件sitewiseBase，其中具有與此資產的資產模型相對應的元件類型。
- 任何資產層級取代 (例如設定屬性別名或度量單位) 都會反映在中的實體中 AWS IoT TwinMaker。

## 僅預設工作區

下列資產會同步，並且僅可在預設工作區中使用IoTSiteWiseDefaultWorkspace。

### AssetModelComponents

AWS IoT TwinMaker 為AssetModelComponents中的每個項目建立一個新元件類型 AWS IoT SiteWise。

- 資產模型TypeId的元件使用下列樣式：assetModelId
- 資產模型中的每個性質都是元件類型中的新性質，性質名稱為propertyId。中的屬性名稱 AWS IoT SiteWise 會儲存為屬性定義displayName中的。
- 資產模型中的每個階層都是類型的新性質，LIST且nestedType位RELATIONSHIP於元件類型中。階層會對應至名稱前置為的屬性。hierarchyId

## AssetModelCompositeModel

AWS IoT TwinMaker 為AssetModelCompositeModel中的每個項目建立一個新元件類型 AWS IoT SiteWise。

- 資產模型TypeId的元件使用下列樣式：assetModelId\_assetModelCompositeModelId
- 資產模型中的每個性質都是元件類型中的新性質，性質名稱為propertyId。中的屬性名稱 AWS IoT SiteWise 會儲存為屬性定義displayName中的。

## AssetCompositeModels

AWS IoT TwinMaker 為AssetCompositeModel中的每個項目建立新複合元件 AWS IoT SiteWise。

- 與中的相assetModelCompositeModelId同 AWS IoT SiteWise。componentName

## 資源未同步

下列資源未同步：

### 未同步的資產和資產模型

- 警報模型將同步為複合模型，但與警報相關的資產中的對應資料不會同步。
- [AWS IoT SiteWise 資料串流](#)不會同步。只有在資產模型中建立模型的屬性才會同步。
- 屬性、測量值、轉換、彙總和中繼資料計算 (例如公式和視窗) 的屬性值不會同步。只會同步有關屬性的中繼資料，例如別名、測量單位和資料類型。可以使用一般 AWS IoT TwinMaker 資料連接器 API 來查詢這些值。 [GetPropertyValueHistory](#)

## 在中使用已同步的實體和元件類型 AWS IoT TwinMaker

同步資產之後 AWS IoT SiteWise，同步的元件類型就會在中讀取 AWS IoT TwinMaker。任何更新或刪除動作都必須在中完成 AWS IoT SiteWise，如果 SyncJob 仍處於作用中狀態，AWS IoT TwinMaker 則這些變更會同步至。

同步處理的實體和 AWS IoT SiteWise 基礎元件也是唯讀的 AWS IoT TwinMaker。您可以將其他未同步的元件新增至已同步的實體，只要沒有實體層級屬性 (例如說明) 或entityName已更新。

某些限制適用於您如何與已同步的實體互動。您無法在已同步實體階層中的已同步實體下建立子系實體。此外，您無法建立從同步元件類型延伸的非同步元件類型。

**Note**

如果在中刪除資產 AWS IoT SiteWise 或刪除同步工作，則會與實體一起刪除其他元件。

您可以在 Grafana 儀表板中使用這些同步的實體，並將它們新增為場景撰寫器中的標籤，就像一般實體一樣。您也可以針對這些已同步的實體發出知識圖查詢。

**Note**

未經修改的已同步實體不會收取費用，但如果在中進行變更，則會向您收取這些實體的費用 AWS IoT TwinMaker。例如，如果您將未同步的元件新增至已同步的實體，該實體現在會收取費用。AWS IoT TwinMaker 如需詳細資訊，請參閱 [AWS IoT TwinMaker 定價](#)。

## 分析同步狀態和錯誤

本主題提供有關如何分析同步錯誤和狀態的指引。

**Important**

[the section called “自訂和預設工作區之間的差異”](#) 如需有關自訂工作區和預設工作區之間差異的資訊，請參閱。

## 同步工作狀態

根據同步工作的狀態，同步工作會有下列其中一種狀態。

- 同步工作 CREATING 狀態表示工作正在檢查權限並從中載入資料 AWS IoT SiteWise 以準備同步。
- 同步工作 INITIALIZING 狀態表示中的所有現有資源 AWS IoT SiteWise 都已同步到 AWS IoT TwinMaker。如果使用者中有大量資產和資產模型，則此步驟可能需要更長的時間才能完成 AWS IoT SiteWise。您可以透過檢查 [AWS IoT TwinMaker 主控台](#) 中的同步工作或呼叫 ListSyncResources API 來監視已同步的資源數量。
- 同步工作 ACTIVE 狀態表示初始化步驟已完成。工作現在已準備就緒，可以從中同步任何新的更新 AWS IoT SiteWise。
- 同步工作 ERROR 狀態會指出前述任何狀態的錯誤。檢閱錯誤訊息。IAM 角色設定可能有問題。如果您想要使用新的 IAM 角色，請刪除發生錯誤的同步工作，然後使用新角色建立新的同步工作。

同步錯誤會出現在模型來源頁面中，該頁面可從工作區中的「實體」模型來源表格存取。模型來源頁面會顯示無法同步的資源清單。同步工作會自動重試大多數錯誤，但如果資源需要動作，則會維持狀ERROR態。您也可以使用 [ListSyncResources](#) API 取得錯誤清單。

若要查看目前來源的所有列出錯誤，請使用下列程序。

1. 導覽至 [AWS IoT TwinMaker 主控台](#) 中的工作區。
2. 選取「實體」模型來 AWS IoT SiteWise 源模式中列出的來源，以開啟資產同步詳細資訊頁面。

The screenshot displays the 'AWS IoT SiteWise source' configuration page. The 'Overview' section shows the source is 'ACTIVE' and was created on January 20, 1970. It lists 8 total resources, with 6 in sync and 2 in error. The 'Errors (2)' section contains a search bar and a table with the following data:

Resource name	External id	Status	Status reason
e8a7fff4-289c-4b28-8814-6dc3e5a13612	e8a7fff4-289c-4b28-8814-6dc3e5a13612	ERROR	{code:"SYNC_INITIALIZING_ERROR","message":"SYNC INITIALIZING ERROR"}
18fd0d54-a268-4558-b40a-34c3f7af9228	18fd0d54-a268-4558-b40a-34c3f7af9228	ERROR	{code:"SYNC_INITIALIZING_ERROR","message":"SYNC INITIALIZING ERROR"}

3. 如前面的螢幕擷取畫面所示，任何具有持續性錯誤的資源都會列在「錯誤」表格中。您可以使用此表格追蹤並修正與特定資源相關的錯誤。

可能的錯誤包括：

- 雖然 AWS IoT SiteWise 支援重複的資產名稱，但 AWS IoT TwinMaker 僅在 ROOT 層級支援它們，而不支援相同的父實體下。如果在中的父實體下有兩個名稱相同的資產 AWS IoT SiteWise，其中一個資產無法同步。若要修正此錯誤，請先刪除其中一個資產，或在同步 AWS IoT SiteWise 之前將資產移至其他父資產下方。
- 如果您已經擁有與 AWS IoT SiteWise 資產 ID 相同 ID 的實體，則該資產無法同步，直到您刪除現有實體為止。



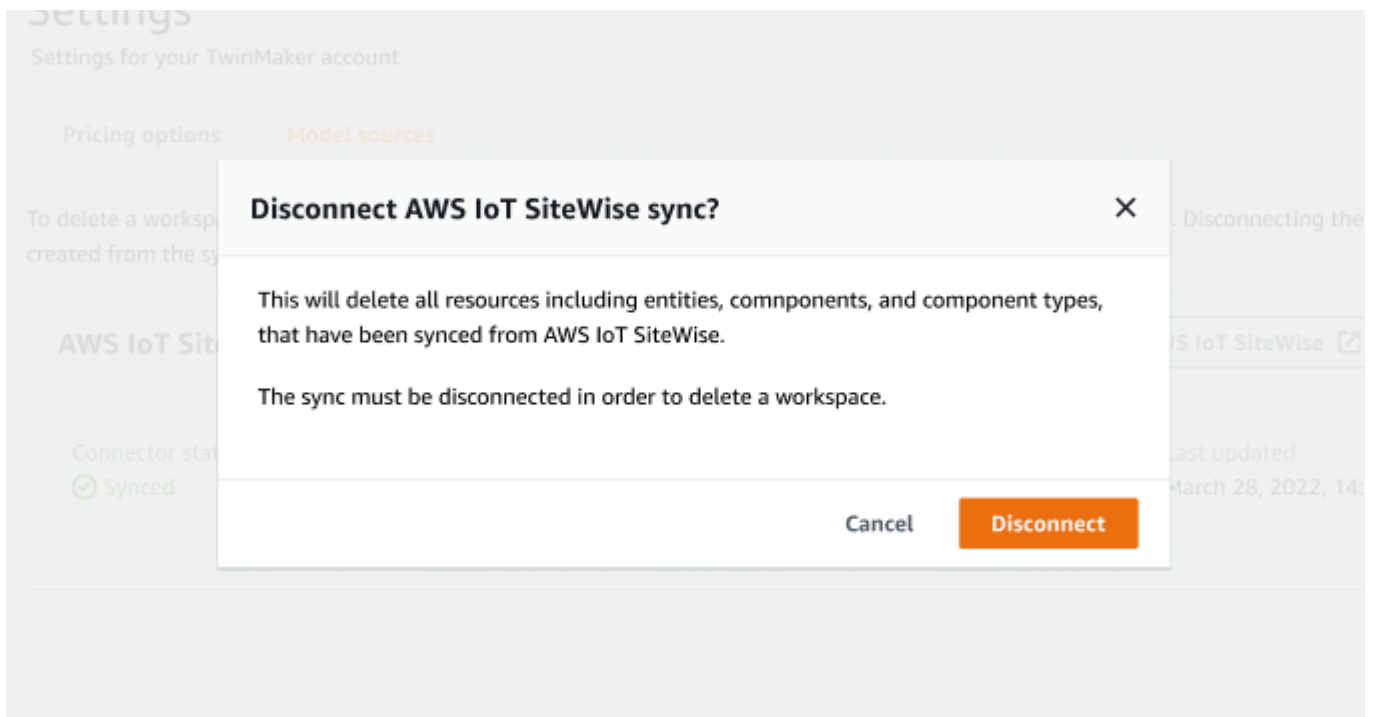
## 刪除同步工作

請使用下列程序刪除同步工作。

### **⚠ Important**

[the section called “自訂和預設工作區之間的差異”](#)如需有關自訂工作區和預設工作區之間差異的資訊，請參閱。

1. 導覽至 [AWS IoT TwinMaker 主控台](#)。
2. 開啟您要從中刪除同步工作的工作區。
3. 在「實體模型來源」下，選取要開啟來 AWS IoT SiteWise 源詳細資訊頁面的來源。
4. 若要停止同步工作，請選擇「中斷連線」。確認您的選擇以完全刪除同步工作。



刪除同步工作後，您可以在相同或不同的工作區中再次建立同步工作。

如果工作區中有任何同步工作，則無法刪除該工作區。刪除工作區之前，請先刪除同步工作。

如果刪除同步工作期間發生任何錯誤，同步工作會保持DELETING狀態並自動重試。如果發生與刪除資源相關的任何錯誤，您現在可以手動刪除任何已同步的實體或元件類型。

**Note**

同步處理的所有資源 AWS IoT SiteWise 都會先刪除，然後刪除同步工作本身。

## 資產同步限制

**Important**

[the section called “自訂和預設工作區之間的差異”](#)如需有關自訂工作區和預設工作區之間差異的資訊，請參閱。

由於 [AWS IoT SiteWise 配額](#) 高於預設配 [AWS IoT TwinMaker 額](#)，因此我們正在增加以下同步處理的實體和元件類型的限制 AWS IoT SiteWise。

- 工作區中有 1000 個同步元件類型，因為它只能從中同步 1000 個資產模型 AWS IoT SiteWise。
- 工作區中有 100,000 個已同步的實體，因為它只能從中同步 100,000 個資產 AWS IoT SiteWise。
- 每個父系實體最多 2000 個子系實體。它可以同步每個單一母資產 2000 個子資產。

**Note**

[GetEntityAPI](#) 只會傳回階層屬性的前 50 個子實體，但您可以使用 [GetPropertyValueAPI](#) 來分頁和擷取所有子系實體的清單。

- 每個已同步元件的 600 個屬性 AWS IoT SiteWise，可將資產模型與總共 600 個屬性和階層同步。

**Note**

這些限制僅適用於已同步的實體。如果您需要增加未同步資源的這些限制，請求提高配額。

# AWS IoT TwinMakerGrafana 板集成

AWS IoT TwinMaker通過應用程式插件支持 Grafana 集成。您可以使用 Grafana 8.2.0 及更新版本與您的數位孿生應用程式互動。該AWS IoT TwinMaker插件提供了自定義面板，儀表板模板以及用於連接到數字孿生數據的數據源。

如需如何使用 Grafana 加入並設定儀表板權限的詳細資訊，請參閱下列主題：

## 主題

- [用於 Grafana 場景查看器的 CORS 配置](#)
- [設定您的 Grafana 環境](#)
- [建立儀表板 IAM 角色](#)
- [建立AWS IoT TwinMaker視訊播放程式政策](#)

### Note

您需要修改 Amazon S3 儲存貯體的 CORS (跨來源資源共用) 組態，以允許 Grafana 使用者界面從儲存貯體載入資源。如需指示，請參閱[用於 Grafana 場景查看器的 CORS 配置](#)。

如需 AWS IoT TwinMaker Grafana 外掛程式的詳細資訊，請參閱[AWS IoT TwinMaker應用程式文件](#)。

如需 Grafana 外掛程式關鍵元件的詳細資訊，請參閱下列內容：

- [AWS IoT TwinMaker資料來源](#)
- [儀表板範本](#)
- [場景檢視器面板](#)
- [視訊播放器面板](#)

## 用於 Grafana 場景查看器的 CORS 配置

AWS IoT TwinMakerGrafana 外掛程式需要 CORS (跨來源資源共用) 組態，這可讓 Grafana 使用者界面從 Amazon S3 儲存貯體載入資源。如果沒有 CORS 組態，您會在場景檢視器上收到「載入 3D 場景失敗且網路失敗」的錯誤訊息，因為 Grafana 網域無法存取 Amazon S3 儲存貯體中的資源。

若要使用 CORS 設定您的 Amazon S3 儲存貯體，請使用下列步驟：

1. 登入 IAM 主控台並開啟 [Amazon S3 主控台](#)。
2. 在「值區」清單中，選擇您用作工作區資源值AWS IoT TwinMaker區的值區名稱。
3. 選擇許可。
4. 在「跨來源資源共用」區段中，選取「編輯」以開啟 CORS 編輯器。
5. 在 CORS 組態編輯器文字方塊中，輸入或複製並貼上下列 JSON CORS 設定，方法是以您的網域取代 Grafana 工作區網域 *GRAFANA-WORKSPACE-DOMAIN*。

**Note**

您需要在 "AllowedOrigins": JSON 元素的開頭保持星號\* 字符。

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "PUT",
      "POST",
      "DELETE",
      "HEAD"
    ],
    "AllowedOrigins": [
      "*GRAFANA-WORKSPACE-DOMAIN"
    ],
    "ExposeHeaders": [
      "ETag"
    ]
  }
]
```

6. 選擇保存更改以完成 CORS 配置。

如需有關搭配 Amazon S3 儲存貯體的 CORS 的詳細資訊，請參閱[使用跨來源資源共用 \(CORS\)](#)。

## 設定您的 Grafana 環境

您可以使用 Amazon 受管 Grafana 進行全受管服務，或設定自己管理的 Grafana 環境。使用 Amazon 受管 Grafana，您可以根據自己的需求快速部署、操作和擴展開放原始碼 Grafana。或者，您也可以設定自己的基礎架構來管理 Grafana 伺服器。

如需兩個 Grafana 環境選項的相關資訊，請參閱下列主題：

- [Amazon Managed Grafana](#)
- [自我管理 Grafana](#)

### Amazon Managed Grafana

Amazon 託管 Grafana 提供了一個 AWS IoT TwinMaker 插件，以便您可以快速將 AWS IoT TwinMaker 與 Grafana 集成。由於 Amazon 受管的 Grafana 會為您管理 Grafana 伺服器，因此您無需建置、封裝或部署任何硬體或任何其他 Grafana 基礎設施，就能將資料視覺化。有關 Amazon 託管 Grafana 的更多信息，請參閱 [什麼是 Amazon 託管 Grafana](#) ？。

#### Note

Amazon 託管的 Grafana 目前支持 Grafana 插件 AWS IoT TwinMaker 的 1.3.1 版本。

### Amazon 託管 Grafana 先決條件

若要 AWS IoT TwinMaker 在 Amazon 受管的 Grafana 儀表板中使用，請先完成下列先決條件：

- 建立 AWS IoT TwinMaker 工作區。如需 [有關建立工作區的詳細資訊](#)，請參閱 [入門 AWS IoT TwinMaker](#)。

#### Note

當您第一次在 AWS 管理主控台中建立 Amazon 受管的 Grafana 工作區時，AWS IoT TwinMaker 並未列出。但是，該插件已經安裝在所有工作區上。您可以在開源 Grafana AWS IoT TwinMaker 插件列表中找到該插件。您可以在「AWS IoT TwinMaker 資料來源」頁面上選擇新增資料來源來尋找資料來源。

當您建立 Amazon 受管的 Grafana 工作區時，系統會自動建立 IAM 角色來管理 Grafana 執行個體的許可。這稱為「工作區 IAM 角色」。這是您將用來配置 Grafana 所有AWS IoT TwinMaker數據源的身份驗證提供程序選項。Amazon 受管的 Grafana 不支援自動新增許可AWS IoT TwinMaker，因此您必須手動設定這些許可。如需設定手動權限的詳細資訊，請參閱[建立儀表板 IAM 角色](#)。

## 自我管理 Grafana

您可以選擇託管自己的基礎架構來執行 Grafana。如需在電腦上本機執行 Grafana 的相關資訊，請參閱[安裝 Grafana](#)。該AWS IoT TwinMaker插件可在公共 Grafana 書目錄。[如需在 Grafana 環境中安裝此外掛程式的相關資訊，請參閱AWS IoT TwinMaker應用程式。](#)

當您在本機執行 Grafana 時，您無法輕易共用儀表板或提供多個使用者存取權。[如需使用本機 Grafana 共用儀表板的指令碼快速入門指南，請參閱AWS IoT TwinMaker範例儲存庫](#)。此資源會引導您完成在公有端點上託管 Grafana 環境的 Cloud9 和 Amazon EC2。

您必須決定要使用哪個驗證提供者來設定 TwinMaker 資料來源。您可以根據預設認證鏈結設定環境的認證(請參閱[使用預設認證提供者鏈結](#))。預設認證可以是任何使用者或角色的永久認證。例如，如果您在 Amazon EC2 上執行 Grafana，則預設登入資料鏈可以存取 [Amazon EC2 執行角色](#)，然後該角色將成為您的身份驗證提供者。在步驟中需要身份驗證提供者的 IAM Amazon 資源名稱 (ARN)。[建立儀表板 IAM 角色](#)

## 建立儀表板 IAM 角色

使用此功能AWS IoT TwinMaker，您可以在 Grafana 儀表板上控制資料存取。Grafana 儀表板使用者應該有不同的權限範圍才能檢視資料，而在某些情況下，則需要寫入資料。例如，警報操作員可能沒有觀看視頻的權限，而管理員則具有所有資源的權限。Grafana 透過資料來源定義許可，其中提供了登入資料和 IAM 角色。AWS IoT TwinMaker資料來源會擷取具有該角色權限的AWS認證。如果未提供 IAM 角色，Grafana 會使用登入資料的範圍，此範圍無法縮減。AWS IoT TwinMaker

若要在 Grafana 中使用AWS IoT TwinMaker儀表板，您需要建立 IAM 角色並附加政策。您可以使用下列範本來協助您建立這些原則。

## 建立 IAM 政策

建立在 IAM 主控台 `YourWorkspaceIdDashboardPolicy` 中呼叫的 IAM 政策。此政策可讓您的工作區存取 Amazon S3 儲存貯體和AWS IoT TwinMaker資源。您也可以決定[針對 Amazon Kinesis Video Streams 使用AWS IoT Greengrass邊緣連接器](#)，這需要 Kinesis 影片串流的許可，以及為元件設定的AWS IoT SiteWise資產。若要符合您的使用案例，請選擇下列其中一個原則範本。

## 1. 沒有視頻權限政策

如果您不想使用 Grafana [視訊播放器面板](#)，請使用下列範本建立原則。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3::bucketName/*",
        "arn:aws:s3:::bucketName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}
```

系統會為每個工作區建立一個 Amazon S3 儲存貯體。它包含要在儀表板上檢視的 3D 模型和場景。面 [SceneViewer](#) 板會從此值區載入項目。

## 2. 限定範圍的視訊權限原則

若要限制 Grafana 中「視訊播放器」面板上的存取，請依標籤將 Amazon Kinesis Video Streams 的 AWS IoT Greengrass 邊緣連接器分組。如需關閉影片資源的限定範圍權限的詳細資訊，請參閱。[建立 AWS IoT TwinMaker 視訊播放程式政策](#)

## 3. 所有影片權限

如果您不想將視頻分組，則可以使所有視頻都可以從 Grafana 視頻播放器訪問。任何擁有 Grafana 工作區存取權的人都可以播放您帳戶中任何串流的影片，而且可以唯讀存取任何 AWS IoT SiteWise 資產。這包括 future 建立的任何資源。

使用下列範本建立策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ],
}
```



```

{
  "Effect": "Allow",
  "Action": [
    "kinesisvideo:GetDataEndpoint",
    "kinesisvideo:GetHLSStreamingSessionURL"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:GetAssetPropertyValue",
    "iotsitewise:GetInterpolatedAssetPropertyValues"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    }
  }
}
]
}

```

此原則範本提供下列權限：

- S3 儲存貯體的唯讀存取權以載入場景。
- 工作區中所有實體和元件的唯讀存取權。AWS IoT TwinMaker
- 以唯讀存取權來串流您帳戶中所有 Kinesis 影片串流影片。
- 只讀訪問您帳戶中所有AWS IoT SiteWise資產的財產價值歷史記錄。
- 將資料擷取至以金鑰EdgeConnectorForKVS和值workspaceId標記的AWS IoT SiteWise資產的任何屬性。

## 標記您的相機AWS IoT SiteWise資產請求從邊緣上傳視頻

使用 Grafana 中的視訊播放器，使用者可以手動要求將視訊從邊緣快取上傳到 Kinesis Video Streams。您可以針對與 Amazon Kinesis 影片串流的AWS IoT Greengrass邊緣連接器相關聯且已標記金鑰EdgeConnectorForKVS的任何AWS IoT SiteWise資產開啟此功能。

標籤值可以是以下列任何字元分隔的工作區域清單：. : + = @ \_ / -例如，如果您想要跨AWS IoT TwinMaker工作區使用與 Amazon Kinesis Video Streams 的AWS IoT Greengrass邊緣連接器相關聯的AWS IoT SiteWise資產，則可以使用遵循以下模式的標籤：WorkspaceA/WorkspaceB/WorkspaceC。Grafana 外掛程式會強制使用AWS IoT TwinMaker工作區 ID 來分組資產資料擷取。AWS IoT SiteWise

## 將更多權限新增至儀表板原則

AWS IoT TwinMakerGrafana 外掛程式會使用您的驗證提供者來呼叫您建立 AssumeRole 的儀表板角色。在內部，插件通過在調用中使用會 AssumeRole 話策略來限制您可以訪問的最高權限範圍。如需階段作業原則的詳細資訊，請參閱[工作階段原則](#)

這是您在工作區的儀表板角色上可以擁有的最大寬容政策：AWS IoT TwinMaker

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
        "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "Effect": "Allow",
    "Action": "iottwinmaker:ListWorkspaces",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:GetHLSStreamingSessionURL"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:GetAssetPropertyValue",
      "iotsitewise:GetInterpolatedAssetPropertyValues"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:BatchPutAssetPropertyValue"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId"
      }
    }
  }
]
}

```

如果您添加了Allow更多權限的語句，則它們將無法在AWS IoT TwinMaker插件上使用。這是通過設計來確保插件使用的最低必要權限。

但是，您可以進一步縮小權限的範圍。如需相關資訊，請參閱 [建立AWS IoT TwinMaker視訊播放程式政策](#)。

## 建立 Grafana 儀表板 IAM 角色

在 IAM 主控台中，建立名為的 IAM 角色 `YourWorkspaceIdDashboardRole`。附加 `YourWorkspaceIdDashboardPolicy` 至角色。

若要編輯儀表板角色的信任原則，您必須授與 Grafana 驗證提供者呼叫儀表板角色 `AssumeRole` 的權限。使用下列範本更新信任原則：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "ARN of Grafana authentication provider"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需建立 Grafana 環境及尋找驗證提供者的詳細資訊，請參閱。[設定您的 Grafana 環境](#)

## 建立AWS IoT TwinMaker視訊播放程式政策

以下是包含 Grafana AWS IoT TwinMaker 插件所需的所有視頻權限的政策模板：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/*",
        "arn:aws:s3:::bucketName"
      ]
    },
    {
      "Effect": "Allow",
```

```

    "Action": [
      "iottwinmaker:Get*",
      "iottwinmaker:List*"
    ],
    "Resource": [
      "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId",
      "arn:aws:iottwinmaker:region:accountId:workspace/workspaceId/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iottwinmaker:ListWorkspaces",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:GetHLSStreamingSessionURL"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:GetAssetPropertyValue",
      "iotsitewise:GetInterpolatedAssetPropertyValues"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:BatchPutAssetPropertyValue"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
      }
    }
  }
]

```

```
}
```

如需完整原則的詳細資訊，請參閱主[建立 IAM 政策](#)題中的所有視訊權限原則範本。

## 縮減對資源的存取範圍

Grafana 中的視訊播放器面板可直接呼叫 Kinesis Video Streams 和 IoT，SiteWise 以提供完整的視訊播放體驗。若要避免未經授權存取與您的AWS IoT TwinMaker工作區無關聯的資源，請為您的工作區儀表板角色的 IAM 政策新增條件。

## 縮減 GET 權限的範圍

您可以標記資源來縮小 Amazon Kinesis Video Streams 和資AWS IoT SiteWise產的存取範圍。您可能已根據 AWS IoT TwinMaker WorkSpaceID 標記AWS IoT SiteWise相機資產，以啟用視訊上傳請求功能，請參閱[從邊緣上傳視訊](#)主題。您可以使用相同的標籤索引鍵值配對來限制對AWS IoT SiteWise資產的 GET 存取，也可以用相同的方式標記 Kinesis Video Streams。

然後，您可以將此條件添加到運動視頻和異位語句中：*YourWorkspaceId*DashboardPolicy

```
"Condition": {
  "StringLike": {
    "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId"
  }
}
```

## 真實使用案例：分組攝影機

在這種情況下，你有一個大陣列的攝影機監控烘烤餅乾的過程在一個工廠。在麵糊室製作一批餅乾麵糊，麵糊在冷凍室冷凍，並在烘烤室烘烤餅乾。這些房間中的每個房間都有攝影頭，由不同的操作員團隊分別監控每個過程。您希望每組運營商都可以為其各自的房間進行授權。當為 cookie 工廠構建數字孿生時，使用單個工作區，但攝影機權限需要按房間限定。

您可以根據相機的群組 ID 標記相機群組，以達到此權限分離。在這個案例中，群組 ID 為 BatterRoom FreezerRoom、和。 BakingRoom每個房間的攝影機都連接到 Kinesis Video Streams，並且應該有一個標籤，其中包含：鍵 = EdgeConnectorForKVS、值 = BatterRoom。該值可以是由下列任何字元分隔的群組清單：. : + = @ \_ / -

要修改*YourWorkspaceId*DashboardPolicy，請使用以下政策聲明：

```
....
```

```

{
  "Effect": "Allow",
  "Action": [
    "kinesisvideo:GetDataEndpoint",
    "kinesisvideo:GetHLSStreamingSessionURL"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*groupingId*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:GetAssetPropertyValue",
    "iotsitewise:GetInterpolatedAssetPropertyValues"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*groupingId*"
    }
  }
},
...

```

這些陳述式會限制串流視訊播放和AWS IoT SiteWise屬性歷程記錄存取群組中的特定資源。由您的使用案例定義。*groupingId*在我們的場景中，這將是 RooMid。

## 範圍關閉AWS IoT SiteWise BatchPutAssetPropertyValue 權限

提供此權限可開啟「[視訊播放器](#)」中的[視訊上傳要求功能](#)。上傳影片時，您可以指定時間範圍，並透過選擇 Grafana 控制面板上的「送出」來提交要求。

要授予物聯網站：BatchPutAssetPropertyValue 權限，請使用默認策略：

```

...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ]
}

```

```

],
"Resource": "*",
"Condition": {
  "StringLike": {
    "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
  }
}
},
...

```

透過使用此原則，使用者可以呼叫 BatchPutAssetPropertyValue AWS IoT SiteWise攝影機資產上的任何屬性。您可以在陳述式的條件中指定特定 propertyId，以限制該特定 PropertyID 的授權。

```

{
  ...
  "Condition": {
    "StringEquals": {
      "iotsitewise:propertyId": "propertyId"
    }
  }
  ...
}

```

Grafana 中的「視訊播放器」面板會將資料擷取至測量屬性 (名為) VideoUploadRequest，以啟動將視訊從邊緣快取上傳至 Kinesis Video Streams。在控制台中找到此屬性的屬性 propertyId。AWS IoT SiteWise要修改*YourWorkspaceId*DashboardPolicy，請使用以下政策聲明：

```

...,
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/EdgeConnectorForKVS": "*workspaceId*"
    },
    "StringEquals": {
      "iotsitewise:propertyId": "VideoUploadRequestPropertyId"
    }
  }
}

```



```
},  
...
```

此聲明會限制將資料擷取到已標記AWS IoT SiteWise相機資產的特定性質。如需詳細資訊，請參閱[如AWS IoT SiteWise何使用 IAM](#)。

# 將 AWS IoT SiteWise 警報 Connect 到儀表 AWS IoT TwinMaker 板 Grafana

## Note

此功能處於公開預覽狀態，可能會有所變更。

AWS IoT TwinMaker 能夠將 AWS IoT SiteWise 警報導入 AWS IoT TwinMaker 組件。這可讓您查詢警示狀態並設定警示閾值，而無需實作資料移轉的自訂 AWS IoT SiteWise 資料連接器。您可以使用 AWS IoT TwinMaker Grafana 外掛程式將警報狀態視覺化，並在 Grafana 中設定警報閾值，而無需對警報進行 API 呼叫 AWS IoT TwinMaker 或直接與警報互動。AWS IoT SiteWise

## AWS IoT SiteWise 警示組態先決條件

在建立警報並將其整合到 Grafana 儀表板之前，請確定您已檢閱下列先決條件：

- 熟悉 AWS IoT SiteWise 模型和資產系統。如需詳細資訊，請參閱《使用指南》中的 [〈建立資產模型〉](#) 和 [〈建立資產〉](#)
- 熟悉 IoT Events 警示模型，以及如何將它們附加至 AWS IoT SiteWise 模型。如需詳細資訊，請參閱使用指南中的 [定義 AWS IoT 事件警示](#)。
- AWS IoT TwinMaker 與 Grafana 整合，讓您可以在 Grafana 中存取您的 AWS IoT TwinMaker 資源。若要取得更多資訊，請參閱 [AWS IoT TwinMaker Grafana 板集成](#)。

## 定義 AWS IoT SiteWise 警示元件 IAM 角色

AWS IoT TwinMaker 使用工作區 IAM 角色來查詢和設定 Grafana 中的警示閾值。AWS IoT TwinMaker 工作區角色需要下列權限，才能與 Grafana 中的 AWS IoT SiteWise 警示互動：

```
{
  "Effect": "Allow",
  "Action": [
    "iotevents:DescribeAlarmModel",
  ],
  "Resource": ["{IoTEventsAlarmModelArn}"]
},{
```

```
"Effect": "Allow",
  "Action": [
    "iotsitewise:BatchPutAssetPropertyValue"
  ],
  "Resource": ["{IoTSitewiseAssetArn}"]
}
```

在主 [AWS IoT TwinMaker 控制台](#) 中，建立代表您 AWS IoT SiteWise 資產的實體。請務必使用 `com.amazon.iotsitewise.alarm` 作為元件類型來新增該實體的元件，並挑選對應的資產和警示模型。

**Add component**

**Component information**

Name  
DustAlarm

Type  
Types of components include documents, time-series data, structured data, and unstructured data.  
com.amazon.iotsitewise.alarm

Asset Model  
Choose an asset model.  
ConstructionSpot

Asset  
Choose an asset.  
Spot1

Alarm Model  
Choose an alarm model.  
ConstructionSpotDustAlarm

上面的屏幕截圖是使用該類型創建此實體的示例 `com.amazon.iotsitewise.alarm`。

當您建立此元件時，AWS IoT TwinMaker 會自動從 AWS IoT SiteWise 和匯入相關的警示屬性 AWS IoT Events。您可以重複此警報元件類型模式，為工作區中所需的所有資產建立警報元件。

## 透過 AWS IoT TwinMaker API 進行查詢和更新

建立警示元件後，您可以透過 AWS IoT TwinMaker API 查詢警示狀態、閾值以及更新警示閾值。

以下是查詢警報狀態的請求範例：

```
aws iottwinmaker get-property-value-history --cli-input-json \
'{
  "workspaceId": "{workspaceId}",
  "entityId": "{entityId}",
  "componentName": "{componentName}",
  "selectedProperties": ["alarm_status"],
  "startTime": "{startTimeIsoString}",
  "endTime": "{endTimeIsoString}"
}'
```

以下是查詢警示臨界值的範例請求。

```
aws iottwinmaker get-property-value-history --cli-input-json \
'{
  "workspaceId": "{workspaceId}",
  "entityId": "{entityId}",
  "componentName": "{componentName}",
  "selectedProperties": ["alarm_threshold"],
  "startTime": "{startTimeIsoString}",
  "endTime": "{endTimeIsoString}"
}'
```

以下是更新警示臨界值的範例請求：

```
aws iottwinmaker batch-put-property-values --cli-input-json \
'{
  "workspaceId": "{workspaceId}",
  "entries": [
    {
      "entityPropertyReference": {
        "entityId": "{entityId}",
        "componentName": "{componentName}",
        "propertyName": "alarm_threshold"
      },
      "propertyValues": [
        {
          "value": {
            "doubleValue": "{newThreshold}"
          },
          "time": "{effectiveTimeIsoString}"
        }
      ]
    }
  ]
}'
```

```

    }
  ]
}'

```

## 為您的 Grafana 儀表板設定鬧鐘

需要創建第二個啟用寫入的儀表板 IAM 角色，這是一個普通角色，但有權將動作添加 `iottwinmaker:BatchPutPropertyValues` 到 TwinMaker 工作區 arn，如下例所示。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:Get*",
        "iottwinmaker:List*",
        "iottwinmaker:BatchPutPropertyValues"
      ],
      "Resource": [
        "{workspaceArn}",
        "{workspaceArn}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iottwinmaker:ListWorkspaces",
      "Resource": "*"
    }
  ]
}

```

或者，您可以在 IAM 角色的末尾新增此陳述式：

```

{
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:BatchPutPropertyValues"
  ],
  "Resource": [
    "{workspaceArn}",

```

```

    "{workspaceArn}/*"
  ]
}

```

數據源需要使用您創建的儀表板寫入角色設置寫入 arn。

修改 IAM 角色後，請登入您的 Grafana 儀表板以擔任更新的角色 arn。選中「定義警報配置面板的寫入權限」核取方塊，然後在 arn 中複製「寫入」角色。

The screenshot shows the Grafana Settings page for a data source named "AWS IoT TwinMaker-4". The "Connection Details" section includes fields for Authentication Provider (AWS SDK Default), Assume Role ARN (arn:aws:iam:\*), External ID, Endpoint (https://{service}.{region}.amazonaws.com), and Default Region (us-east-1). Below this is a warning box titled "Assume Role ARN" with instructions to specify an IAM role. The "TwinMaker settings" section includes a "Workspace" field (enter workspace ID), a checked checkbox for "Define write permissions for Alarm Configuration Panel", and an "Assume Role ARN Write" field containing "arn:aws:iam:\*", which is circled in red. At the bottom are buttons for "Back", "Explore", "Delete", and "Save & test".

## 使用 Grafana 儀表板進行警報可視化

使用以下步驟將警報配置面板新增至儀表板並對其進行配置：

1. 在面板選項中選取工作區。
2. 在查詢配置中設置數據源。
3. 請使用下列查詢類型：Get Property Value History by Entity。
4. 選取要新增警示的實體或實體變數。
5. 選取圖元後，請選取要套用屬性的元件或元件變數。
6. 對於屬性，請選擇：alarm\_status和alarm\_threshold。

連接後，您應該會看到警報 ID 的 ID 和當前閾值。

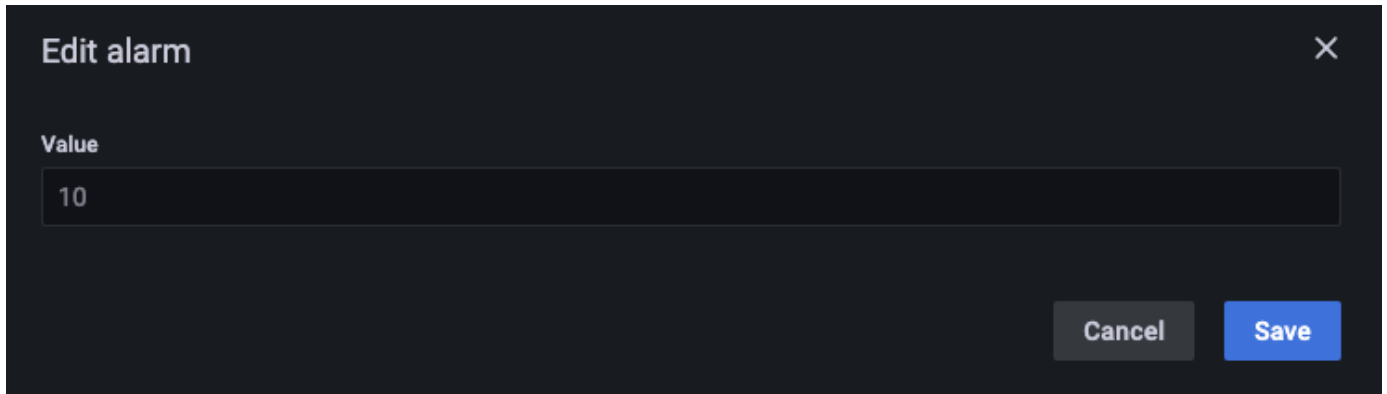
### Note

對於公開預覽，不會顯示任何通知。您應該檢閱警示狀態和臨界值，以確定已正確套用屬性。

7. 應使用預設「遞增」的「查詢順序」，以便顯示最新值。
8. 查詢的過濾器部分可以保留空白。完整的配置如下圖所示：

9. 透過使用 [編輯鬧鐘] 按鈕，您可以開啟對話方塊來變更目前的警示臨界值。

## 10. 選取「儲存」以設定新的臨界值。



The screenshot shows a dark-themed dialog box titled "Edit alarm" with a close button in the top right corner. Below the title, there is a label "Value" and a text input field containing the number "10". At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

### Note

此面板只能與包含目前的即時時間範圍搭配使用。將其與過去結束和開始的時間範圍一起使用時，將警報閾值編輯為始終是當前閾值時，可能會顯示未預期的值。



# AWS IoT TwinMaker 馬特港集成

Matterport 提供多種擷取選項來掃描真實世界的環境，並建立身歷其境的 3D 模型，也稱為 Matterport 數位雙胞胎。這些模型被稱為馬特波特空間。AWS IoT TwinMaker 支持 Matterport 集成，允許您將 Matterport 數字雙胞胎導入到場景中。AWS IoT TwinMaker 透過將 Matterport 數位雙胞胎與配對 AWS IoT TwinMaker，您可以在虛擬環境中視覺化並監控您的數位雙胞胎系統。



[如需使用 Matterport 的詳細資訊，請參閱 Matterport 的說明文件和 Matterport 頁面AWS IoT TwinMaker。](#)

## 整合主題

- [整合概述](#)
- [物流端整合先決條件](#)
- [產生並記錄您的 Matterport 認證](#)
- [將您的馬特港認證存儲在 AWS Secrets Manager](#)
- [將物件連接埠空間匯入場景 AWS IoT TwinMaker](#)
- [在 Grafana 儀表板中使用馬特港空間 AWS IoT TwinMaker](#)
- [在 Web 應用程式中使用 Matterport 空格 AWS IoT TwinMaker](#)

## 整合概述

此整合可讓您執行下列作業：

- 在 AWS IoT TwinMaker 應用程式套件中使用您的 Matterport 標籤和空格。
- 在 AWS IoT TwinMaker Grafana 儀表板中檢視匯入的資料。有關使用 AWS IoT TwinMaker 和 Grafana 的更多信息，請閱讀 Grafana 儀表板[集成文檔](#)。
- 將 Matterport 空間匯入您的 AWS IoT TwinMaker 場景。
- 選取並匯入您想要繫結至場景中資料的 AWS IoT TwinMaker Matterport 標籤。
- 自動顯示您的 Matterport 空間並標記 AWS IoT TwinMaker 場景中的變更，並核准要同步的內容。

整合程序包含 3 個關鍵步驟。

1. [產生並記錄您的 Matterport 認證](#)
2. [將您的馬特港認證存儲在 AWS Secrets Manager](#)
3. [將物件連接埠空間匯入場景 AWS IoT TwinMaker](#)

您可以在[AWS IoT TwinMaker 主控台](#)中開始整合。在主控台的「設定」頁面的第三方資源下，開啟 Matterport 整合，在整合所需的資源之間瀏覽。

The screenshot shows the AWS IoT TwinMaker Settings page for Matterport integration. The page is titled 'Settings' and includes a navigation menu on the left with options like 'How it works', 'Workspaces', 'Component types', 'Entities', 'Resource library', 'Scenes', and 'Query editor'. The main content area is divided into sections: 'Pricing options', 'Model sources', and '3rd party resources'. Under '3rd party resources', there is a section for 'Matterport integration (1) Info'. This section contains a 'How it works' guide with four steps: 1. Contact Matterport, 2. Record your Matterport SDK credentials, 3. Add your Matterport credentials into AWS secrets manager, and 4. Select your Matterport account in a scene composer scene. Below the steps is a 'Connected accounts' table, which currently shows 'No connections' and a button to 'AWS Secret Manager'.

## 物流端整合先決條件

在整合 Matterport 與之前，AWS IoT TwinMaker 請確定您符合下列先決條件：

- 您已購買企業級 [Matterport](#) 帳戶，以及整合所需的 Matterport 產品。AWS IoT TwinMaker
- 您有 AWS IoT TwinMaker 工作區。如需詳細資訊，請參閱[開始使用 AWS IoT TwinMaker](#)。
- 您已更新 AWS IoT TwinMaker 工作區角色。如需有關建立工作區角色的詳細資訊，請參閱[建立和管理的服務角色 AWS IoT TwinMaker](#)。

將以下內容新增至您的工作區角色：

```
{
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": [
    "AWS Secrets Manager secret ARN"
  ]
}
```

- 您必須聯絡 Matterport 以配置啟用整合的必要授權。Matterport 也會為整合啟用專用模型內嵌 (PME)。

如果您已經有 Matterport 客戶經理，請直接與他們聯絡。

如果您沒有 Matterport 的聯絡點，請使用下列程序聯絡 Matterport 並要求整合：

1. 開啟[馬特港和 AWS IoT TwinMaker](#)頁面。
2. 按「聯絡我們」按鈕，開啟聯絡表單。
3. 在表格上填寫必要的信息。
4. 當您準備好時，請選擇「打招呼」，將您的請求傳送至 Matterport。

請求整合之後，您可以產生繼續整合程序所需的必要 Matterport SDK 和私有模型內嵌 (PME) 認證。

#### Note

這可能涉及您為購買新產品或服務而產生費用。

## 產生並記錄您的 Matterport 認證

若要與 Matterport 整合 AWS IoT TwinMaker，您必須提供 AWS Secrets Manager Matterport 認證。請使用下列程序來產生 Matterport SDK 認證。

1. 登入您的[物業港](#)帳戶。
2. 前往您的帳戶設定頁面。
3. 進入設置頁面後，選擇開發人員工具選項。
4. 在 [開發人員工具] 頁面上，前往 SDK 金鑰管理區段。
5. 進入 SDK 金鑰管理區段後，選取新增 SDK 金鑰的選項。

- 取得 Matterport SDK 金鑰後，請將網域新增至您的 Grafana 伺服器 AWS IoT TwinMaker 的金鑰。如果您使用的是 AWS IoT TwinMaker 應用程式套件，請務必新增您的自訂網域。
- 接下來，找到「應用程式集成管理」部分，您應該會看到列出您的 PME 應用程式。記錄下列資訊：
  - 用戶端識別碼
  - 用戶端密碼

#### Note

由於用戶端密碼只會顯示一次，因此我們強烈建議您記錄您的用戶端密碼。您必須在 AWS Secrets Manager 主控台中出示您的用戶端密碼，才能繼續進行 Matterport 整合。

當您購買必要元件且 Matterport 已啟用您帳戶的 PME 時，系統會自動建立這些認證。如果未顯示這些認證，請聯絡 Matterport。若要請求聯絡，請參閱 [Matterport 和 AWS IoT TwinMaker 聯絡表單](#)。

[如需更多有關 Matterport SDK 認證的資訊，請參閱 Matterport 官方 SDK 文件概觀。](#)

## 將您的馬特港認證存儲在 AWS Secrets Manager

請使用下列程序將您的 Matterport 認證儲存在中。AWS Secrets Manager

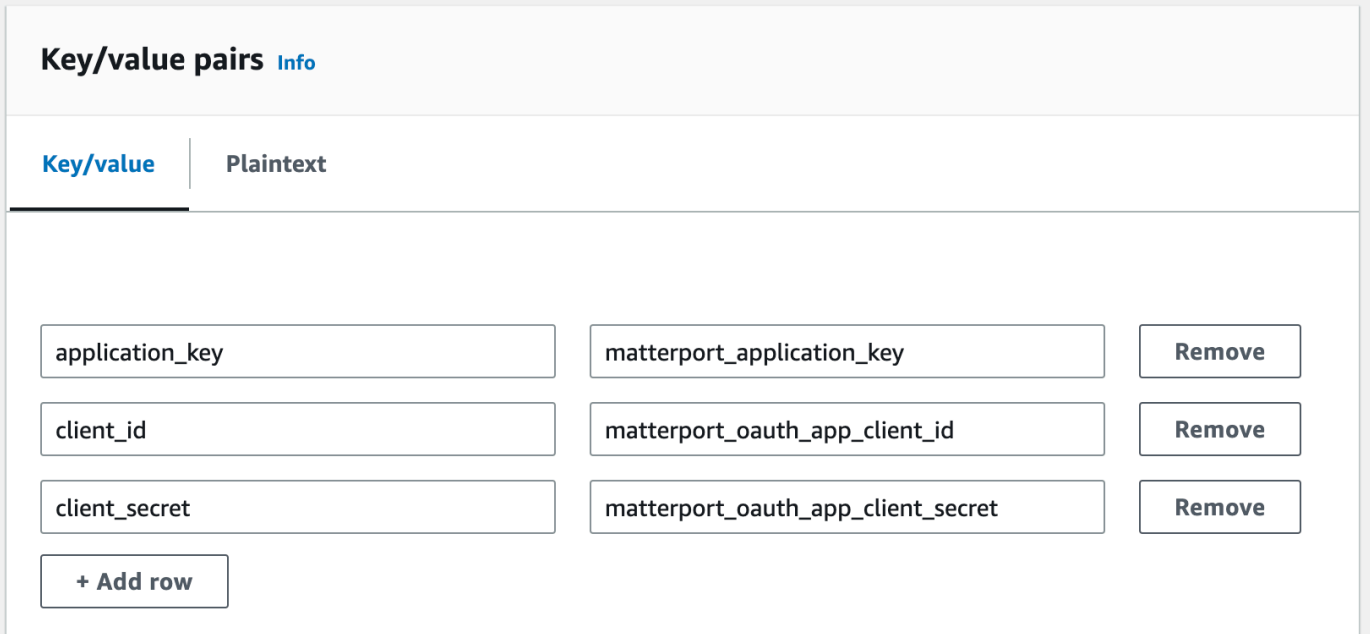
#### Note

您需要從[產生並記錄您的 Matterport 認證](#)主題中的程序建立的用戶端密碼，才能繼續進行 Matterport 整合。

- 登入 AWS Secrets Manager 主控台。
- 導覽至「密碼」頁面，然後選取「儲存新密碼」。
- 對於「密碼」類型，選取「其他類型的密碼」。
- 在 [索引鍵/值配對] 區段中，新增下列索引鍵值配對，並將您的 Matterport 認證做為值：
  - <your Matterport credentials> 創建一個鍵-值對，鍵：application\_key 和值：。

- <your Matterport credentials> 創建一個鍵-值對，鍵：client\_id 和值：。
- <your Matterport credentials> 創建一個鍵-值對，鍵：client\_secret 和值：。

完成後，您應該擁有類似下列範例的組態：



The screenshot shows the 'Key/value pairs' section of the AWS Secrets Manager console. It features a table with two columns: 'Key/value' and 'Plaintext'. The table contains three rows of key-value pairs, each with a 'Remove' button to its right. Below the table is a '+ Add row' button.

Key/value	Plaintext	
application_key	matterport_application_key	Remove
client_id	matterport_oauth_app_client_id	Remove
client_secret	matterport_oauth_app_client_secret	Remove

+ Add row

5. 對於加密金鑰，您可以保持aws/secretsmanager選取的預設加密金鑰。
6. 選擇「下一步」以前往「設定密碼」頁面。
7. 填寫「秘密名稱」和「描述」欄位。
8. 在「標籤」區段中為此機密新增標籤。

創建標籤時，如下面的屏幕截圖所示分配密鑰：AWSIoTTwinMaker\_Matterport

AWS Secrets Manager > Secrets > Store a new secret

Step 1  
Choose secret type

Step 2  
**Configure secret**

Step 3  
Configure rotation - optional

Step 4  
Review

## Configure secret

### Secret name and description [Info](#)

**Secret name**  
A descriptive name that helps you find your secret later.

Secret name must contain only alphanumeric characters and the characters /\_+=.@-

**Description - optional**

Maximum 250 characters.

### Tags - optional

Key	Value - optional	
<input type="text" value="AWSIoTwinMaker_Matterport"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add"/>		

### Note

您必須新增標籤。添加第三方秘密時需要標籤 AWS Secrets Manager，儘管標籤被列為可選。

值欄位為選填。提供金鑰後，您可以選取 [新增] 以繼續下一個步驟。

- 選擇 [下一步] 以前往 [設定輪替] 頁面。設定秘密輪換是選擇性的。如果您想要完成新增密碼而不需要輪換，請再次選擇 [下一步]。如需秘密輪換的詳細資訊，請參閱[旋轉 AWS Secrets Manager 密碼](#)。
- 在 [檢閱] 頁面上確認您的密碼組態。準備好新增秘密後，請選擇 [商店]。

如需使用的詳細資訊 AWS Secrets Manager，請參閱下列 AWS Secrets Manager 文件：

- [使用建立和管理機密 AWS Secrets Manager](#)
- [什麼是 AWS Secrets Manager ?](#)

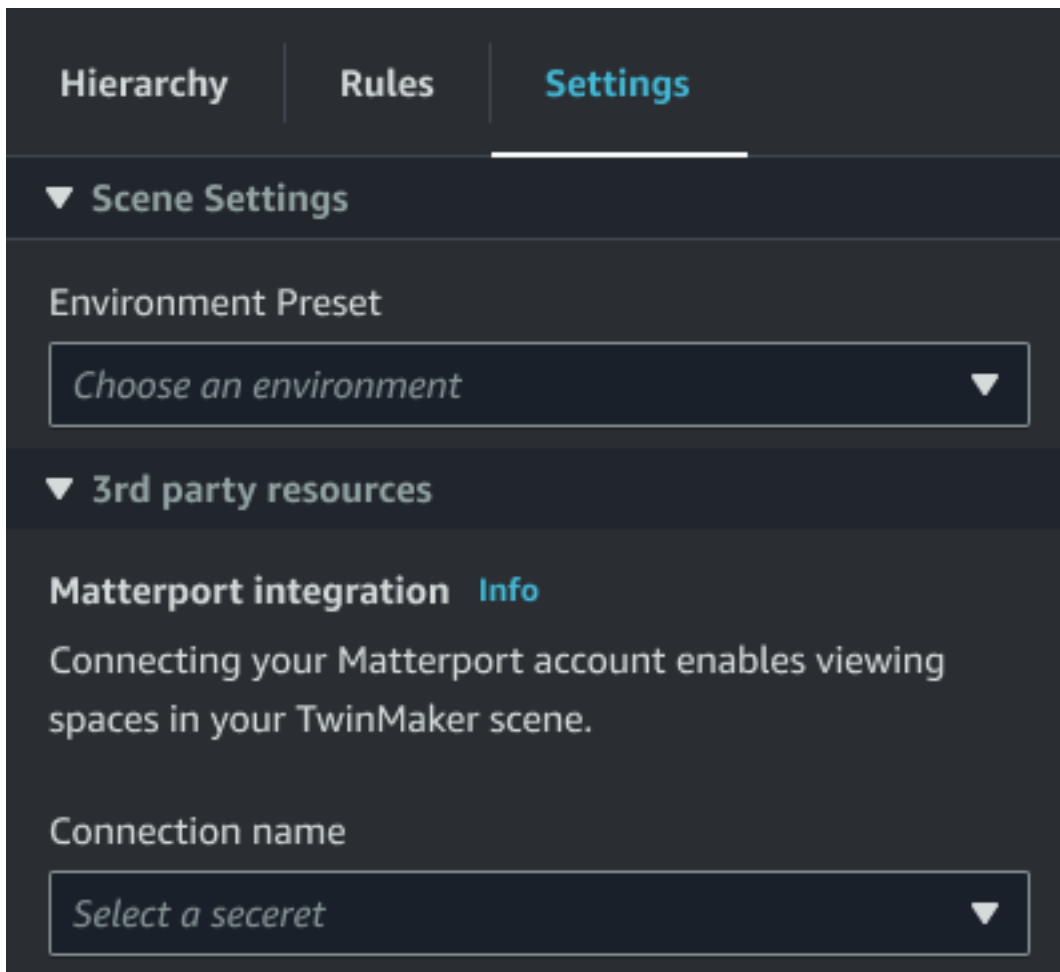
- [旋轉 AWS Secrets Manager 秘密](#)

現在您可以將 AWS IoT TwinMaker Matterport 資產匯入場景了。請參閱下一節中的程序，[將物件連接埠空間匯入場景 AWS IoT TwinMaker](#)

## 將物件連接埠空間匯入場景 AWS IoT TwinMaker

從場景設定頁面中選擇已連接的 Matterport 帳戶，將 Matterport 掃描檔新增至您的場景。請遵循下列程序來匯入您的 Matterport 掃描和標籤：

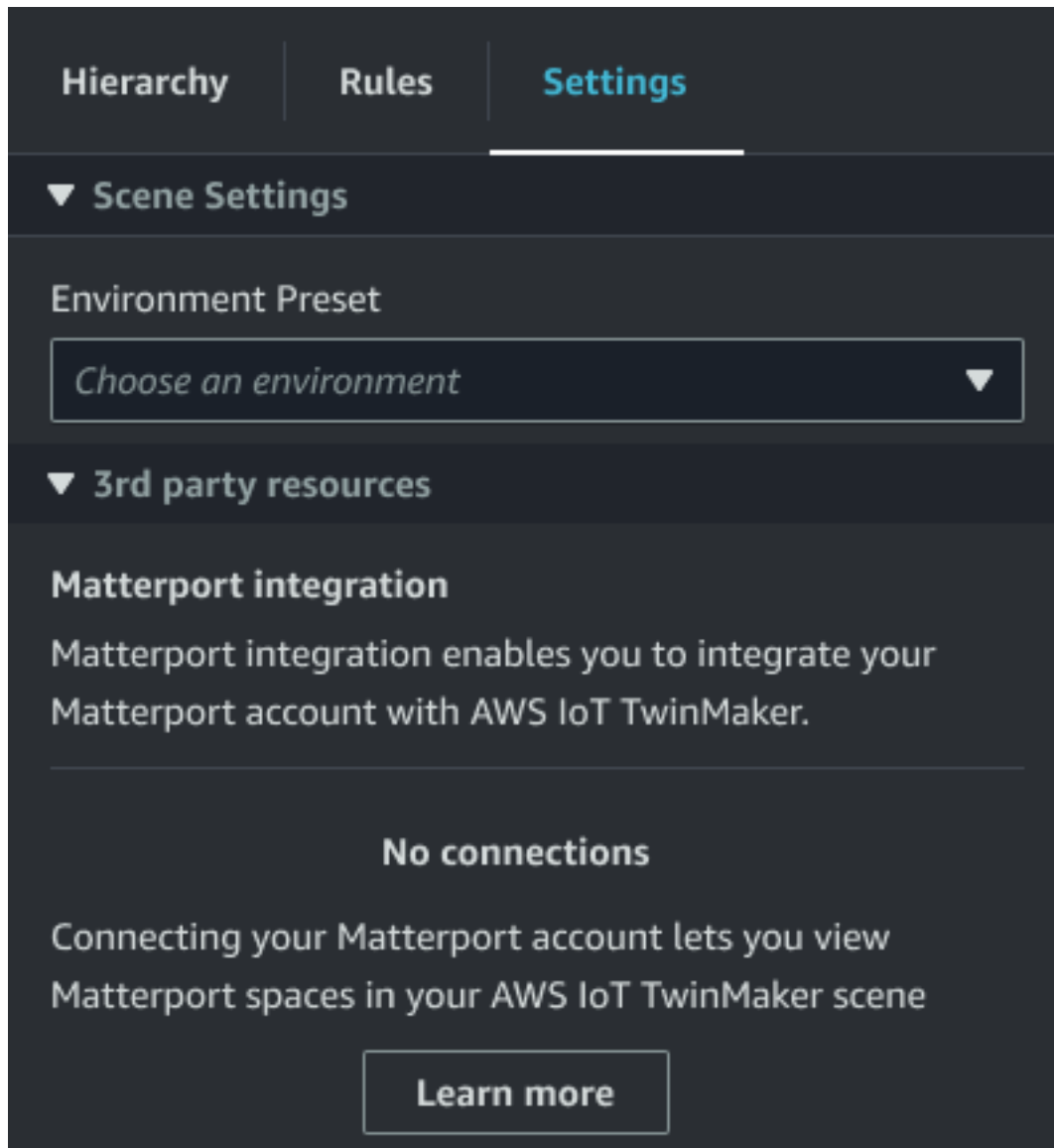
1. 登入 [AWS IoT TwinMaker 主控台](#)。
2. 建立或開啟您要在其中使用 Matterport 空間的現有 AWS IoT TwinMaker 場景。
3. 場景開啟後，導覽至「設定」標籤。
4. 在 [設定] 的 [協力廠商資源] 下，找到 [連線名稱]，然後輸入您在程序中建立的密碼[將您的馬特港認證存儲在 AWS Secrets Manager](#)。



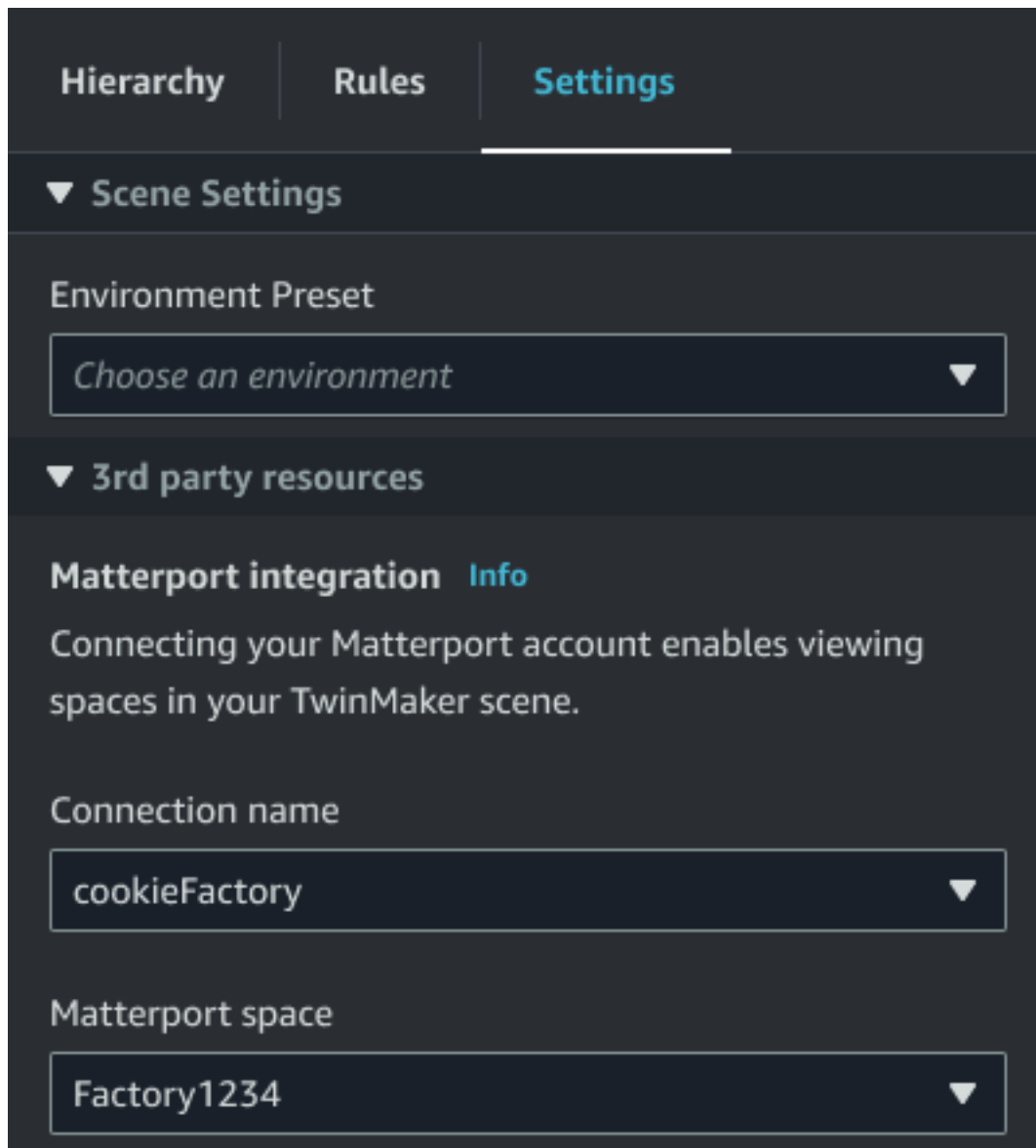


**Note**

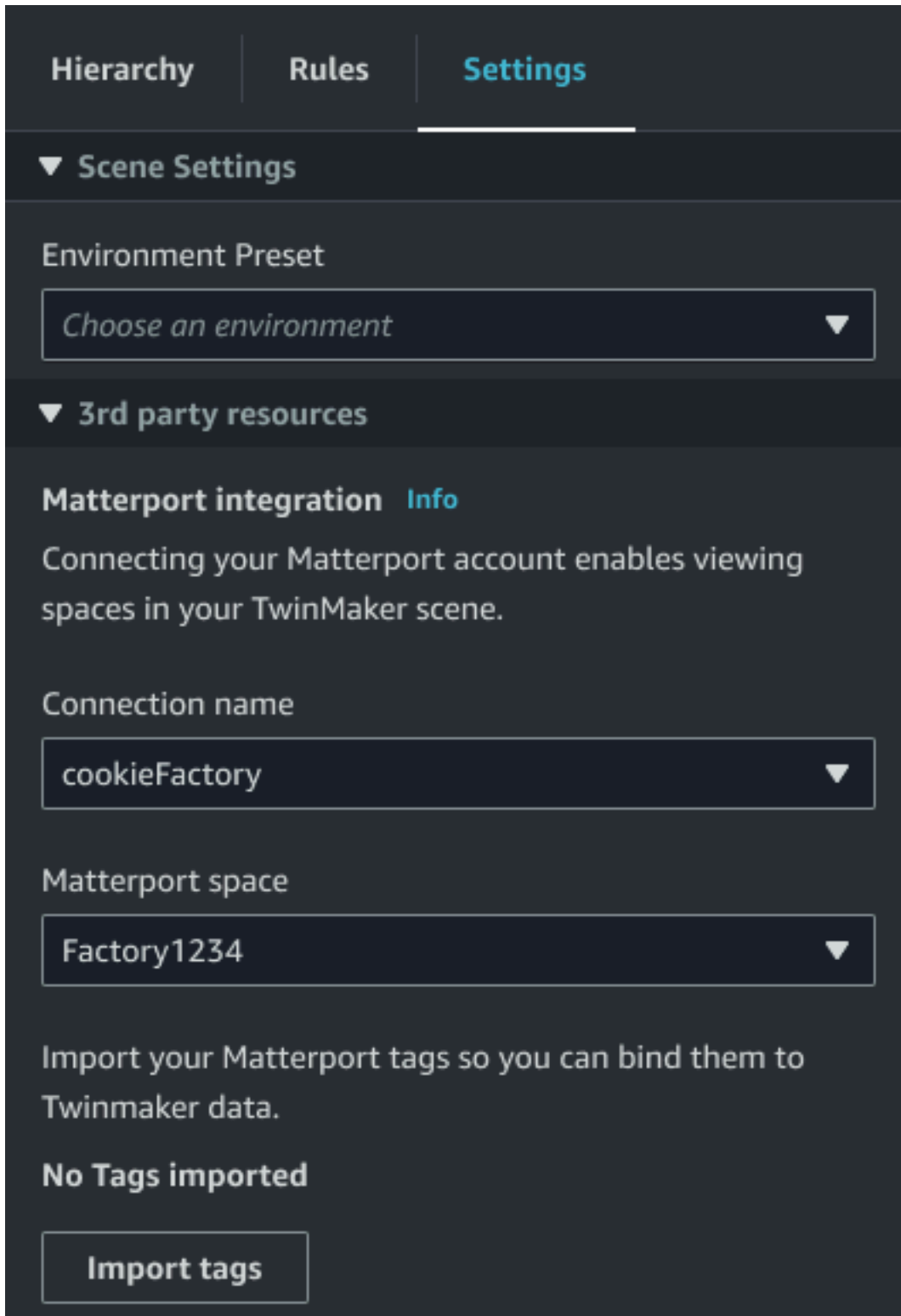
如果您看到指出「沒有連線」的訊息，請瀏覽至[AWS IoT TwinMaker 主控台設定頁面](#)以開始 Matterport 整合程序。



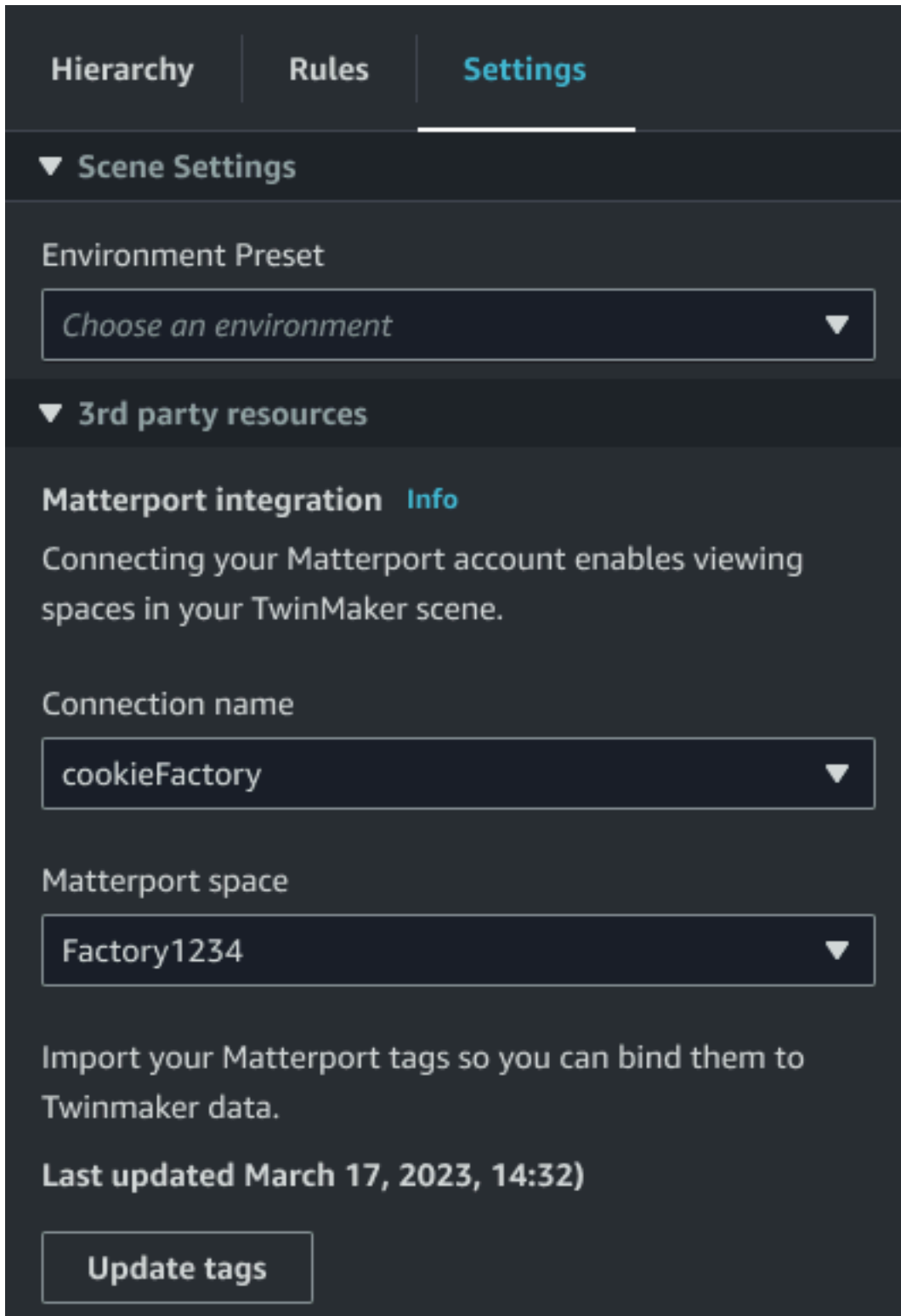
5. 接下來，在 Matterport 空間下拉式清單中選擇您想要在場景中使用的 Matter port 空間。



6. 選取空間後，您可以匯入 Matterport 標籤，然後按「匯入標籤」按鈕將其轉換為 AWS IoT TwinMaker 場景標籤。



匯入 Matterport 標籤之後，該按鈕會被更新標籤按鈕取代。您可以在中持續更新您的 Matterport 標籤，以 AWS IoT TwinMaker 便它們始終反映您 Matterport 帳戶中的最新變更。



7. 您已成功 AWS IoT TwinMaker 與 Matterport 整合，現在您的 AWS IoT TwinMaker 場景同時擁有匯入的 Matterport 空間和標籤。您可以像處理任何其他場景一樣在此 AWS IoT TwinMaker 場景中工作。

如需使用全 AWS IoT TwinMaker 景的詳細資訊，請參閱[建立和編輯 AWS IoT TwinMaker 場景](#)。

## 在 Grafana 儀表板中使用馬特港空間 AWS IoT TwinMaker

將 Matterport 空間匯入 AWS IoT TwinMaker 場景後，您可以使用 Grafana 儀表板中的 Matterport 空間檢視該場景。如果您已經使用 Grafana 進行配置 AWS IoT TwinMaker，則只需打開 Grafana 儀表板，即可使用匯入的 Matterport 空間檢視您的場景。

如果您尚未 AWS IoT TwinMaker 使用 Grafana 進行設定，請先完成 Grafana 整合程序。AWS IoT TwinMaker 與 Grafana 整合時，您有兩種選擇。您可以使用自我管理的 Grafana 執行個體，也可以使用 Amazon 受管的 Grafana。

請參閱下列文件，進一步瞭解 Grafana 選項與整合程序：

- [AWS IoT TwinMaker Grafana 儀表板集成](#)。
- [Amazon 管理 Grafana](#)。
- [自我管理的 Grafana](#)。

## 在 Web 應用程式中使用 Matterport 空格 AWS IoT TwinMaker

將 Matterport 空間匯入 AWS IoT TwinMaker 場景後，您可以使用應用程式套件 Web AWS IoT 應用程式中的 Matterport 空間檢視該場景。

請參閱下列文件，進一步了解如何使用 AWS IoT 應用程式套件：

- 若要進一步瞭解如何 AWS IoT TwinMaker 搭配應用 AWS IoT 程式套件使用，請參閱[使用 AWS IoT TwinMaker UI 組件創建自定義 Web 應用程式](#)。
- 要了解有關使用 AWS IoT 應用程式包的更多信息，請訪問[AWS IoT 應用程式包 Github](#) 頁面。
- 如需如何使用應用程式套件啟動新 Web AWS IoT 應用程式的指示，請造訪官方 [IoT App Kit](#) 文件頁面。

# AWS IoT TwinMaker視訊整合

攝像機為數字雙胞胎模擬提供了一個很好的機會。您可以使用AWS IoT TwinMaker來模擬相機的位置和身體狀況。在中AWS IoT TwinMaker為您的現場攝影機建立實體，並使用視訊元件將即時視訊和中繼資料從您的網站串流到AWS IoT TwinMaker場景或 Grafana 儀表板。

AWS IoT TwinMaker可以通過兩種方式從邊緣設備捕獲視頻。您可以使用 Kinesis 視訊串流的邊緣連接器從邊緣裝置串流視訊，也可以將視訊儲存在邊緣裝置上，並透過 MQTT 訊息開始視訊上傳。使用此元件從您的裝置串流視訊資料，以便與AWS IoT服務搭配使用。若要為 Kinesis Video [Streams 產生必要的資源並部署邊緣連接器](#)，請參閱 [Kinesis 視訊串流邊緣連接器入門](#)。GitHub如需有關AWS IoT Greengrass元件的詳細資訊，請參閱 [Kinesis Video Streams 邊緣連接器](#)的AWS IoT Greengrass文件。

建立所需的AWS IoT SiteWise模型並設定 Kinesis Video Streams Greengrass 元件之後，您可以在主控台中將邊緣上的視訊串流或錄製到數位孿生應用程式。AWS IoT TwinMaker您也可以在 Grafana 儀表板中檢視裝置上的即時串流和中繼資料。如需有關整合 Grafana 的更多資訊AWS IoT TwinMaker，請參閱 [AWS IoT TwinMakerGrafana 板集成](#)

## 使用 Kinesis 視訊串流的邊緣連接器來串流影片 AWS IoT TwinMaker

使用 Kinesis 視訊串流的邊緣連接器，您可以將視訊和資料串流到AWS IoT TwinMaker場景中的實體。您可以使用視訊元件來執行此動作。若要建立要在場景中使用的視訊組件，請完成以下程序。

### 必要條件


在AWS IoT TwinMaker場景中建立視訊組件之前，請確定您已完成下列先決條件。

- 為 Kinesis 視訊串流的邊緣連接器建立所需的AWS IoT SiteWise模型和資產。如需為連接器建立AWS IoT SiteWise資產的詳細資訊，請參閱 [Kinesis 視訊串流的邊緣連接器入門](#)。
- 在您的AWS IoT Greengrass裝置上部署了 Kinesis 視訊串流邊緣連接器。如需部署 Kinesis 視訊串流邊緣連接器元件的詳細資訊，請參閱部署 [讀我檔案](#)。

### 建立AWS IoT TwinMaker場景的視訊元件

完成下列步驟，為您的場景建立 Kinesis 視訊串流元件的邊緣連接器。

1. 在主AWS IoT TwinMaker控台中，開啟您要新增視訊元件的場景。
2. 場景開啟後，選擇現有的圖元或建立要加入元件的圖元，然後選擇「新增元件」(Add component)。
3. 在 [新增元件] 窗格中，輸入元件的名稱，然後針對 [類型] 選擇 [類型]。
4. 透過選取您建立的AWS IoT SiteWise相機模型名稱來選擇資產模型。此名稱應具有以下格式：EdgeConnectorForKVSCameraModel-0abc，其中末尾的字母和數字字符串與您自己的資產名稱匹配。
5. 在「資產」中，選擇您要從中串流視訊的AWS IoT SiteWise攝影機資產。會出現一個小窗口，向您顯示當前視頻流的預覽。


 Note

若要測試您的視訊串流，請選擇 [測試]。此測試會傳送 MQTT 事件以啟動視訊即時串流。等待片刻，看到視頻顯示在播放器中。

6. 若要將視訊元件新增至實體，請選擇 [新增元件]。

## 將 Kinesis 視訊串流中的視訊和中繼資料新增至 Grafana 儀表板


在AWS IoT TwinMaker場景中為實體建立視訊元件後，您可以在 Grafana 中設定視訊面板以查看即時串流。請確定您已AWS IoT TwinMaker與 Grafana 正確整合。如需詳細資訊，請參閱 [AWS IoT TwinMakerGrafana 板集成](#)。

 Important

若要在 Grafana 儀表板中檢視視訊，您必須確定 Grafana 資料來源具有適當的 IAM 許可。若要建立必要的角色和策略，請參閱[建立儀表板 IAM 角色](#)。

請完成下列步驟，即可在 Grafana 儀表板中查看 Kinesis Video Streams 和中繼資料。

1. 開啟 AWS IoT TwinMaker 儀表板。
2. 選擇「新增面板」，然後選擇「新增空白面板」。
3. 從面板列表中，選擇AWS IoT TwinMaker視頻播放器面板。
4. 在AWS IoT TwinMaker視訊播放程式面板中，輸入的串流名稱 KinesisVideoStreamName，以及您要從中串流視訊的 Kinesis 視訊串流名稱。

 Note

若要將中繼資料串流至 Grafana 視訊面板，您必須先建立具有視訊串流元件的實體。

5. 選用性：若要將中繼資料從AWS IoT SiteWise資產串流至視訊播放器，對於 Enti AWS IoT TwinMaker ty，請選擇您在AWS IoT TwinMaker場景中建立的實體。針對「組件」名稱，選擇您為AWS IoT TwinMaker場景中實體建立的視訊組件。



# 使用快速AWS IoT TwinMaker連結資源庫

AWS IoT TwinMaker提供 Flink 程式庫，您可以使用該程式庫將資料讀取和寫入至數位雙胞胎中使用的外部資料存放區。

您可以使用 AWS IoT TwinMaker Flink 程式庫，方法是在 Apache Flink 的受管理服務中將其安裝為自訂連接器，並在 Apache Flink 的受管理服務中的 Zeppelin 筆記本中執行 Flink SQL 查詢。筆記本可以提升為持續執行的串流處理應用程式。程式庫會利用AWS IoT TwinMaker元件從您的工作區擷取資料。

AWS IoT TwinMakerFlink 程式庫需要下列項目。

## 必要條件

1. 完全填入的工作區，其中包含場景和元件。針對來自AWS服務 (AWS IoT SiteWise和 Kinesis Video Streams) 的資料使用內建元件類型。為來自第三方來源的資料建立自訂元件類型。如需詳細資訊，請參閱 [???](#)。
2. 工作室筆記本與 Apache Flink 的管理服務的阿帕奇 Flink 的理解。這些筆記本電腦由[阿帕奇柏林飛艇](#)供電，並使用 [Apache Flink](#) 框架。如需詳細資訊，請參閱將[工作室筆記本搭配 Apache Flink 適用於 Apache Flink 的受管理服務搭配使用](#)。

如需有關使用資源庫的指示，請參閱 [AWS IoT TwinMakerFlink 資源庫使用者指南](#)。

如需在AWS IoT TwinMaker範例中AWS IoT TwinMaker使用快速入門進行設定的指示，請參閱 [README 檔案](#)以取得範例見解應用程式。

# AWS IoT TwinMaker 中的記錄和監控

監控對於維護 AWS IoT TwinMaker 及其他 AWS 解決方案的可靠性、可用性和效能至關重要。AWS IoT TwinMaker 支援下列監控工具，可讓您監看服務、在發現錯誤時回報，並適時採取自動動作。

- Amazon Amazon Amazon Amazon SNS 會即時 CloudWatch 監控您的 AWS 資源，以及您在上執行的應用程式 AWS。您可以收集和追蹤指標、建立自訂儀表板，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體指標。如需詳細資訊，請參閱 [Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon CloudWatch](#)
- Amazon CloudWatch Logs 可監控、存放並讓您從 AWS IoT TwinMaker 管道和其他來源存取日誌檔案。CloudTrail CloudWatch 日誌可監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 [Amazon Amazon Amazon Amazon Amazon CloudWatch Logs 使用者指南](#)。
- AWS CloudTrail 擷取您 AWS 帳戶發出或代表發出的 API 呼叫和相關事件，並傳送記錄檔案至您指定的 Simple Storage Service (Amazon S3) 儲存貯體。您可以找出哪些使用者和帳戶呼叫 AWS、發出呼叫的來源 IP 地址，以及呼叫的發生時間。如需詳細資訊，請參閱 [AWS CloudTrail 使用者指南](#)。

## 主題

- [AWS IoT TwinMaker 使用亞馬遜 CloudWatch 指標監控](#)
- [使用 AWS CloudTrail 記錄 AWS IoT TwinMaker API 呼叫](#)

## AWS IoT TwinMaker 使用亞馬遜 CloudWatch 指標監控

您可以使 AWS IoT TwinMaker 用監控 CloudWatch，這會收集原始資料並將其處理成可讀且近乎即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 [Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon Amazon CloudWatch](#)

AWS IoT TwinMaker 將下列各節中列出的量度和維度發佈至 AWS/IoT TwinMaker 命名空間。

**Tip**

AWS IoT TwinMaker 會以一分鐘間隔發佈指標。當您在 CloudWatch 主控台中以圖形檢視這些指標時，建議您選擇 1 分鐘的「期間」，以查看指標資料可用解析度的最高解析度。

## 內容

- [指標](#)

## 指標

AWS IoT TwinMaker 發佈下列指標。

## 指標

指標	描述
ComponentTypeCreationFailure	<p>此測量結果報告建立元件類型是否成功。</p> <p>當元件類型處於CREATING狀態時，就會發佈量度。當使用結構描述初始設定式中所需的屬性建立元件類型，並且這些屬性會以預設值實例化時，就會發生這種情況。</p> <p>指標值可以是0成功或1失敗。</p> <p>尺寸：ComponentTypeId、WorkspaceId。</p> <p>單位：計數</p>
ComponentTypeUpdateFailure	<p>此測量結果報告元件類型更新是否成功。</p> <p>當元件類型處於UPDATING狀態時，就會發佈量度。當組件類型使用結構描述初始設定項中的必要屬性更新，並且這些屬性以默認值實例化時，就會發生這種情況。</p> <p>指標值可以是0成功或1失敗。</p> <p>尺寸：ComponentTypeId、WorkspaceId。</p>

指標	描述
EntityCreationFailure	<p>單位：計數</p> <p>此測量結果會報告實體建立是否成功。當實體處於CREATING狀態時，即會發佈量度。當使用元件建立實體時，就會發生這種情況。</p> <p>指標值可以是0成功或1失敗。</p> <p>尺寸：EntityName、EntityId、WorkspaceId。</p> <p>單位：計數</p>
EntityUpdateFailure	<p>此測量結果會報告實體更新是否成功。當實體處於UPDATING狀態時，即會發佈量度。更新實體時會發生這種情況。</p> <p>指標值可以是0成功或1失敗。</p> <p>尺寸：EntityName、EntityId、WorkspaceId。</p> <p>單位：計數</p>
EntityDeletionFailure	<p>此測量結果會報告刪除實體是否成功。當實體處於DELETING狀態時，即會發佈量度。刪除實體時會發生這種情況。</p> <p>指標值可以是0成功或1失敗。</p> <p>尺寸：EntityName、EntityId、WorkspaceId。</p> <p>單位：計數</p>

 Tip

所有指標都會發佈至命AWS/IoTTwinMaker名空間。

## 使用 AWS CloudTrail 記錄 AWS IoT TwinMaker API 呼叫

AWS IoT TwinMaker與整合AWS CloudTrail，這項服務可提供由使用者、角色或中AWS服務所採取之動作的記錄AWS IoT TwinMaker。CloudTrail 會擷取的 API 呼叫AWS IoT TwinMaker當作事件。擷取的呼叫包括從 AWS IoT TwinMaker 主控台進行的呼叫，以及針對 AWS IoT TwinMaker API 操作的程式碼呼叫。如果您建立追蹤記錄，就可以將 CloudTrail 事件持續交付到 Amazon S3 儲存貯體，包括的事件AWS IoT TwinMaker。即使您未設定追蹤記錄，依然可以透過 CloudTrail 主控台內的 Event history (事件歷史記錄) 檢視最新的事件。您可以使用收集的資訊來 CloudTrail判斷提交給和的請求 AWS IoT TwinMaker、提出請求的 IP 地址、提出請求的對象、提出請求的時間，以及其他詳細資訊。

若要取得有關的更多資訊 CloudTrail，請參閱[AWS CloudTrail使用者指南](#)。

### AWS IoT TwinMaker中的資訊 CloudTrail

當您建立AWS帳戶時，CloudTrail 會自動啟用。CloudTrail 記錄支援中發生的事件活動AWS IoT TwinMaker，以及事件歷史記錄中的其他AWS服務事件。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱[使用 CloudTrail 事件歷程記錄檢視事件](#)。

如需您 AWS 帳戶中正在進行事件的記錄 (包含 AWS IoT TwinMaker 的事件)，請建立線索。線索能 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台建立追蹤記錄時，該追蹤記錄會套用到所有AWS區域。CloudTrail記錄來自AWS分割區中所有區域的事件，然後將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他AWS服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務與整合](#)
- [設定 Amazon SNS 知 SNS 知知 SNS 知知 CloudTrail](#)
- [接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail日誌檔案](#)

大多數AWS IoT TwinMaker操作都由 API 參考記錄 CloudTrail 並記錄在 [AWS IoT TwinMakerAPI 參考](#)中。

下列資料平面操作不會被記錄 CloudTrail：

- [GetPropertyValue](#)
- [GetPropertyValueHistory](#)
- [BatchPutPropertyValues](#)

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

# 中的安全性 AWS IoT TwinMaker

雲端安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，這些架構是為了滿足對安全性最敏感的組織的需求而建置的。

安全是 AWS 與您之間共同承擔的責任。[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：

- 雲端安全性 — AWS 負責保護中執行 AWS 服務的基礎架構 AWS 雲端。AWS 還為您提供可以安全使用的服務。若要瞭解適用於的規範遵循計劃 AWS IoT TwinMaker，請參閱合規計劃的[AWS 服務範圍內的合規計劃](#) AWS 的服務。
- 雲端中的安全性 — 您的責任取決於您使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您的公司的要求和適用法律和法規。

本文件可協助您瞭解如何在使用時套用共同責任模型 AWS IoT TwinMaker。下列主題說明如何設定 AWS IoT TwinMaker 以符合安全性與合規性目標。您也會學到如何使用其他可協助您監控和保護 AWS IoT TwinMaker 資源的 AWS 服務。

## 主題

- [資料保護 AWS IoT TwinMaker](#)
- [的 Identity and Access Management AWS IoT TwinMaker](#)
- [AWS IoT TwinMaker 和介面 VPC 端端點 \( \)AWS PrivateLink](#)
- [符合性驗證 AWS IoT TwinMaker](#)
- [韌性 AWS IoT TwinMaker](#)
- [基礎架構安全 AWS IoT TwinMaker](#)

## 資料保護 AWS IoT TwinMaker

AWS [共用責任模型](#)適用於中的資料保護 AWS IoT TwinMaker。如此模型所述，AWS 負責保護執行所有 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，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。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 資源進行通訊。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。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 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。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，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 \(FIPS\) 140-2 概觀](#)。

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。這包括當您使用主控台、API AWS IoT TwinMaker 或 AWS SDK 時 AWS 服務使用或其他使用時。AWS CLI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。

## 靜態加密

AWS IoT TwinMaker 如果您選擇，可將您的工作區資訊存放在服務為您建立的 Amazon S3 儲存貯體中。服務為您建立的值區已啟用預設伺服器端加密。如果您在建立新工作區時選擇使用自己的 Amazon S3 儲存貯體，建議您啟用預設伺服器端加密。如需 Amazon S3 中預設加密的詳細資訊，請參閱[設定 Amazon S3 儲存貯體的預設伺服器端加密行為](#)。

## 傳輸中加密

所有傳送至的資料都會使用 HTTPS 通訊協定透過 TLS 連線傳送，因此在傳輸過程中預設 AWS IoT TwinMaker 是安全的。

### Note

我們建議您在 Amazon S3 儲存貯體地址上使用 HTTPS 作為控制項，以便在與 Amazon S3 儲存貯體 AWS IoT TwinMaker 互動時強制執行傳輸中加密。如需 Amazon S3 儲存貯體的詳細資訊，請參閱[建立、設定和使用 Amazon S3 儲存貯體](#)。

## 的 Identity and Access Management AWS IoT TwinMaker



AWS Identity and Access Management (IAM) 可協助系統管理員安全地控制 AWS 資源存取權。AWS 服務 IAM 管理員控制哪些人可以驗證 (登入) 和授權 (具有權限) 以使用 AWS IoT TwinMaker 資源。您可以使用 IAM AWS 服務，無需額外付費。

## 主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [如何與 IAM AWS IoT TwinMaker 搭配使用](#)
- [以身分識別為基礎的原則範例 AWS IoT TwinMaker](#)
- [疑難排解 AWS IoT TwinMaker 身分和存取](#)
- [使用 AWS IoT TwinMaker 的服務連結角色](#)
- [AWS 受管理的政策 AWS IoT TwinMaker](#)

## 物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在進行的工作 AWS IoT TwinMaker。

**服務使用者** — 如果您使用 AWS IoT TwinMaker 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 AWS IoT TwinMaker 功能來完成工作時，您可能需要其他權限。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 AWS IoT TwinMaker 中的某項功能，請參閱 [疑難排解 AWS IoT TwinMaker 身分和存取](#)。

**服務管理員** — 如果您負責公司的 AWS IoT TwinMaker 資源，您可能擁有完整的存取權 AWS IoT TwinMaker。決定您的服務使用者應該存取哪些 AWS IoT TwinMaker 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步瞭解貴公司如何搭配使用 IAM AWS IoT TwinMaker，請參閱 [如何與 IAM AWS IoT TwinMaker 搭配使用](#)。

**IAM 管理員**：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 AWS IoT TwinMaker 存取權的詳細資訊。若要檢視可在 IAM 中使用的 AWS IoT TwinMaker 基於身分的政策範例，請參閱 [以身分識別為基礎的原則範例 AWS IoT TwinMaker](#)

## 使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。AWS IAM Identity Center (IAM 身分中心) 使用者、貴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識別的範例。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 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》AWS 帳戶中的[如何登入](#)您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。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[簽署 AWS API 請求](#)。

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) 來增加帳戶的安全性。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[在 AWS 中使用多重要素驗證 \(MFA\)](#)。

### 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，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 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

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identity Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。AWS Directory Service 同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

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，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用

程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，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](#)是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，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，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的 [建立 IAM 使用者 \(而非角色\) 的時機](#)。

## IAM 角色

[IAM 角色](#)是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以 [切換角色，在中暫時擔任 IAM 角色](#)。AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的 [使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要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 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源 (而不是使用角色作為代理)。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的 [IAM 角色與資源類型政策的差異](#)。

- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。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。
- 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。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 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 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。如需更多資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務](#)。
- 服務連結角色 — 服務連結角色是連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。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 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。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 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。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暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的 [利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的 [建立 IAM 角色 \(而非使用者\) 的時機](#)。

## 使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS，當與身分識別或資源相關聯時，會定義其權限。AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，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哪些內容。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。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 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。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的[建立 IAM 政策](#)。

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。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶。受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，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

資源型政策是連接到資源的 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](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

## 存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (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 和 Amazon VPC 是支援 ACL 的服務範例。AWS WAF若要進一步了解 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原則類型。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。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 實體許可範圍](#)。

- 服務控制策略 ( SCP ) — SCP 是 JSON 策略，用於指定中組織或組織單位 ( OU ) 的最大權限。AWS Organizations 是一種用於分組和集中管理您企業擁有的多個 AWS 帳戶的服務。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 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 [SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。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。如需更多資訊，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。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的 [政策評估邏輯](#)。

## 如何與 IAM AWS IoT TwinMaker 搭配使用

在您使用 IAM 管理存取權限之前 AWS IoT TwinMaker，請先了解哪些 IAM 功能可搭配使用 AWS IoT TwinMaker。

您可以搭配使用的 IAM 功能 AWS IoT TwinMaker

IAM 功能	AWS IoT TwinMaker 支持
<a href="#">身分型政策</a>	是
<a href="#">資源型政策</a>	否
<a href="#">政策動作</a>	是
<a href="#">政策資源</a>	是
<a href="#">政策條件索引鍵</a>	是
<a href="#">ACL</a>	否

IAM 功能	AWS IoT TwinMaker 支持
<a href="#">ABAC(政策中的標籤)</a>	部分
<a href="#">臨時憑證</a>	是
<a href="#">主體許可</a>	是
<a href="#">服務角色</a>	是
<a href="#">服務連結角色</a>	否

若要深入瞭解其他 AWS 服務如何 AWS IoT TwinMaker 與大多數 IAM 功能搭配運作，請參閱AWS IAM Identity Center 使用者指南中的搭配 IAM 使用的[AWS 服務](#)。

## 以身分識別為基礎的原則 AWS IoT TwinMaker

支援身分型政策	是
---------	---

身分型政策是可以連接到身分 (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。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

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](#)。

## 以身分識別為基礎的原則範例 AWS IoT TwinMaker

若要檢視以 AWS IoT TwinMaker 身為基礎的原則範例，請參閱。[以身分識別為基礎的原則範例 AWS IoT TwinMaker](#)

## 以資源為基礎的政策 AWS IoT TwinMaker

支援以資源基礎的政策	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](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

若要啟用跨帳戶存取，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作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。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。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 角色與資源型政策有何差異](#)。

## 的政策動作 AWS IoT TwinMaker

支援政策動作 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。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。

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。原則動作通常與關聯的 AWS API 作業具有相同的名稱。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 AWS IoT TwinMaker 動作清單，請參閱服務授權參考 AWS IoT TwinMaker 中[所定義的動作](#)。

中的策略動作在動作之前 AWS IoT TwinMaker 使用下列前置詞：

```
iottwinmaker
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [  
  "iottwinmaker:action1",  
  "iottwinmaker:action2"  
]
```



若要檢視以 AWS IoT TwinMaker 身為基礎的原則範例，請參閱。[以身分識別為基礎的原則範例 AWS IoT TwinMaker](#)

## 的政策資源 AWS IoT TwinMaker

支援政策資源 **是**

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。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 AWS IoT TwinMaker 資源類型及其 ARN 的清單，請參閱服務授權參考 AWS IoT TwinMaker 中[所定義的資源](#)。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 IoT TwinMaker 定義的動作](#)。

若要檢視以 AWS IoT TwinMaker 身為基礎的原則範例，請參閱。[以身分識別為基礎的原則範例 AWS IoT TwinMaker](#)

## 的政策條件索引鍵 AWS IoT TwinMaker

支援服務特定政策條件金鑰 **是**

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。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，請使用邏輯 OR 運算來 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 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容金鑰](#)。

若要查看 AWS IoT TwinMaker 條件索引鍵清單，請參閱服務授權參考 AWS IoT TwinMaker 中的 [條件金鑰](#)。若要瞭解您可以使用條件索引鍵的動作和資源，請參閱 [定義的動作 AWS IoT TwinMaker](#)。

若要檢視以 AWS IoT TwinMaker 身為基礎的原則範例，請參閱 [以身分識別為基礎的原則範例 AWS IoT TwinMaker](#)

## AWS IoT TwinMaker 中的存取控制清單 (ACL)

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。

## 以屬性為基礎的存取控制 (ABAC) 搭配 AWS IoT TwinMaker

支援 ABAC (政策中的標籤)	部分
------------------	----

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。您可以將標籤附加到 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。

若要根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件金鑰，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[什麼是 ABAC?](#)。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 \(ABAC\)](#)。

## 使用臨時登入資料 AWS IoT TwinMaker

支援臨時憑證 是

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料[搭配AWS 服務 使用](#)，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。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[切換至角色 \(主控台\)](#)。

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](#)。

## 的跨服務主體權限 AWS IoT TwinMaker

支援轉寄存取工作階段 (FAS) 是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。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 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 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》](#)。

## AWS IoT TwinMaker的服務角色

支援服務角色 是

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。如需更多資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務](#)。

#### Warning

變更服務角色的權限可能會中斷 AWS IoT TwinMaker 功能。只有在 AWS IoT TwinMaker 提供指引時才編輯服務角色。

## 服務連結角色 AWS IoT TwinMaker

支援服務連結角色。 否

服務連結角色是一種連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服务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。

## 以身分識別為基礎的原則範例 AWS IoT TwinMaker

根據預設，使用者和角色不具備建立或修改 AWS IoT TwinMaker 資源的權限。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。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。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的 [建立 IAM 政策](#)。

如需有關由所定義之動作和資源類型的詳細資訊 AWS IoT TwinMaker，包括每個資源類型的 ARN 格式，請參閱服務授權參考 AWS IoT TwinMaker 中的動作、資源和條件索引 [鍵](#)。

### 主題

- [政策最佳實務](#)
- [使用 AWS IoT TwinMaker 主控台](#)
- [允許使用者檢視他們自己的許可](#)

## 政策最佳實務

以身分識別為基礎的政策會決定某人是否可以建立、存取或刪除您帳戶中的 AWS IoT TwinMaker 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始將權限授與使用者和工作負載，請使用可授與許多常見使用案例權限的 AWS 受管理原則。它們可用在您的 AWS 帳戶。建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定 AWS 服務) 使用 AWS CloudFormation。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫 API 作業時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 [IAM 安全最佳實務](#)。

## 使用 AWS IoT TwinMaker 主控台

若要存取 AWS IoT TwinMaker 主控台，您必須擁有最少一組權限。這些權限必須允許您列出和檢視有關 AWS 帳戶。AWS IoT TwinMaker 如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

若要確保使用者和角色仍可使用 AWS IoT TwinMaker 主控台，請同時將 AWS IoT TwinMaker ConsoleAccess 或受 ReadOnly AWS 管理的原則附加至實體。如需詳細資訊，請參閱 [《使用指南》](#) 中的 [〈將權限新增至 AWS IAM Identity Center 使用者〉](#)。

## 允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

## 疑難排解 AWS IoT TwinMaker 身分和存取

使用下列資訊可協助您診斷和修正使用和 IAM 時可能會遇到的 AWS IoT TwinMaker 常見問題。

### 主題

- [我沒有執行操作的授權 AWS IoT TwinMaker](#)
- [我沒有授權執行 iam : PassRole](#)
- [我想允許我以外的人訪 AWS 帳戶 問我的 AWS IoT TwinMaker 資源](#)

### 我沒有執行操作的授權 AWS IoT TwinMaker

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在 mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 *iottwinmaker:GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iottwinmaker:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 *iottwinmaker:GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

### 我沒有授權執行 iam : PassRole

如果您收到錯誤，告知您未獲授權執行 *iam:PassRole* 動作，您的政策必須更新，允許您將角色傳遞給 AWS IoT TwinMaker。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 marymajor 的 IAM 使用者嘗試使用主控台在 AWS IoT TwinMaker 中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

## 我想允許我以外的人訪 AWS 帳戶 問我的 AWS IoT TwinMaker 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解是否 AWS IoT TwinMaker 支援這些功能，請參閱[如何與 IAM AWS IoT TwinMaker 搭配使用](#)。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱 [IAM 使用者指南中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限](#)。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的[提供第三方 AWS 帳戶 擁有的存取權](#)。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的[將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 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 角色與資源型政策的差異](#)。

## 使用 AWS IoT TwinMaker 的服務連結角色

AWS IoT TwinMaker 使用 AWS Identity and Access Management (IAM) [服務連結角色](#)。服務連結角色是直接連結到 AWS IoT TwinMaker 的唯一 IAM 角色類型。服務連結角色由預先定義，AWS IoT TwinMaker 並包含服務代表您呼叫其他服 AWS 務所需的所有權限。

服務連結角色可讓您 AWS IoT TwinMaker 更輕鬆地設定，因為您不必手動新增必要的權限。AWS IoT TwinMaker 定義其服務連結角色的權限，除非另有定義，否則只 AWS IoT TwinMaker 能擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。這樣可以保護您的 AWS IoT TwinMaker 資源，因為您無法不小心移除存取資源的權限。

如需關於支援服務連結角色的其他服務資訊，請參閱 [《可搭配 IAM 運作的 AWS 服務》](#)，尋找服務連結角色欄中顯示為是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。



## 服務連結角色權限 AWS IoT TwinMaker

AWS IoT TwinMaker 使用名為的服務連結角色 `AWSServiceRoleForIoT`— AWS IoT TwinMaker 允許呼叫其他 AWS 服務，並代表您同步其資源。

服 `AWSServiceRoleForIoT` 務連結角色會信任下列服務擔任該角色：

- `iottwinmaker.amazonaws.com`

名為的角色權限原則 `AWSIoTServiceRolePolicy` AWS IoT TwinMaker 允許對指定的資源完成下列動作：

- 動作：`all your iotsitewise asset and asset-model resources` 上的 `iotsitewise:DescribeAsset`, `iotsitewise:ListAssets`, `iotsitewise:DescribeAssetModel`, and `iotsitewise:ListAssetModels`, `iottwinmaker:GetEntity`, `iottwinmaker>CreateEntity`, `iottwinmaker:UpdateEntity`, `iottwinmaker>DeleteEntity`, `iottwinmaker:ListEntities`, `iottwinmaker:GetComponentType`, `iottwinmaker>CreateComponentType`, `iottwinmaker:UpdateComponentType`, `iottwinmaker>DeleteComponentType`, `iottwinmaker:ListComponentTypes`

您必須設定許可，以允許您的使用者、群組或角色建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的 [服務連結角色許可](#)。

## 建立服務連結角色 AWS IoT TwinMaker

您不需要手動建立一個服務連結角色。當您在 AWS Management Console、或 AWS API 中同步資 AWS IoT SiteWise 產和資產模型 (資產同步) 時 AWS CLI，會為您 AWS IoT TwinMaker 建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您 AWS IoT SiteWise 同步資產和資產模型 (資產同步) 時，AWS IoT TwinMaker 會再次為您建立服務連結角色。

您也可以使用 IAM 主控台建立具有「IoT TwinMaker -受管角色」使用案例的服務連結角色。在 AWS CLI 或 AWS API 中，使用 `iottwinmaker.amazonaws.com` 服務名稱建立服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的「[建立服務連結角色](#)」。如果您刪除此服務連結角色，您可以使用此相同的程序以再次建立該角色。

## 編輯下列項目的服務連結角色 AWS IoT TwinMaker

AWS IoT TwinMaker 不允許您編輯 `AWSServiceRoleForIoT` TwinMaker 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱 IAM 使用者指南中的[編輯服務連結角色](#)。

## 刪除下列項目的服務連結角色 AWS IoT TwinMaker

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。不過，您必須先清除仍在使用服務連結角色的任何服務連結工作區，才能手動刪除角色。

### Note

當您嘗試刪除資源時，如果 AWS IoT TwinMaker 服務正在使用此角色，則刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

## 使用 IAM 手動刪除服務連結角色

使用 IAM 主控台或 AWS API 刪除 `AWSServiceRoleForIoT` TwinMaker 服務連結角色。AWS CLI 如需詳細資訊，請參閱《IAM 使用者指南》中的[刪除服務連結角色](#)。

## 支援 AWS IoT TwinMaker 服務連結角色的區域

AWS IoT TwinMaker 支援在所有提供服務的區域中使用服務連結角色。如需詳細資訊，請參閱[AWS 區域與端點](#)。

## AWS 受管理的政策 AWS IoT TwinMaker

若要新增使用者、群組和角色的權限，使用 AWS 受管理的原則比自己撰寫原則更容易。建立 [IAM 客戶受管政策](#) 需要時間和專業知識，而受管政策可為您的團隊提供其所需的許可。若要快速開始使用，您可以使用我們的 AWS 受管政策。這些政策涵蓋常見的使用案例，並可在您的 AWS 帳戶中使用。如需 AWS 受管政策的詳細資訊，請參閱 IAM 使用者指南中的[AWS 受管政策](#)。

AWS 服務會維護和更新 AWS 受管理的策略。您無法變更 AWS 受管理原則中的權限。服務偶爾會在 AWS 受管政策中新增其他許可以支援新功能。此類型的更新會影響已連接政策的所有身分識別 (使用者、群組和角色)。當新功能啟動或新操作可用時，服務很可能會更新 AWS 受管政策。服務不會從 AWS 受管理的政策移除權限，因此政策更新不會破壞您現有的權限。

此外，還 AWS 支援跨多個服務之工作職能的受管理原則。例如，ReadOnly存取 AWS 管理原則會提供所有 AWS 服務和資源的唯讀存取權。當服務啟動新功能時，會為新作業和資源新 AWS 增唯讀權限。如需任務職能政策的清單和說明，請參閱 IAM 使用者指南中[有關任務職能的AWS 受管政策](#)。

## AWS 受管理的策略：AWSIoTtwinMakerServiceRolePolicy

您無法附加 AWSIoTtwinMakerServiceRolePolicy 到 IAM 實體。此政策會連接到服務連結角色，而此角色可讓 代表您執行動作。如需詳細資訊，請參閱 [服務連結角色權限 AWS IoT TwinMaker](#)。

名為的角色權限原則 AWSIoTtwinMakerServiceRolePolicy AWS IoT TwinMaker 允許對指定的資源完成下列動作：

- 動作：all your iotsitewise asset and asset-model resources 上的 iotsitewise:DescribeAsset, iotsitewise:ListAssets, iotsitewise:DescribeAssetModel, and iotsitewise:ListAssetModels, iottwinmaker:GetEntity, iottwinmaker:CreateEntity, iottwinmaker:UpdateEntity, iottwinmaker>DeleteEntity, iottwinmaker:ListEntities, iottwinmaker:GetComponentType, iottwinmaker:CreateComponentType, iottwinmaker:UpdateComponentType, iottwinmaker>DeleteComponentType, iottwinmaker:ListComponentTypes

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "SiteWiseAssetReadAccess",
    "Effect": "Allow",
    "Action": [
      "iotsitewise:DescribeAsset"
    ],
    "Resource": [
      "arn:aws:iotsitewise:*:*:asset/*"
    ]
  ]
}
```

```

    },
    {
      "Sid": "SiteWiseAssetModelReadAccess",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel"
      ],
      "Resource": [
        "arn:aws:iotsitewise:*:*:asset-model/*"
      ]
    },
    {
      "Sid": "SiteWiseAssetModelAndAssetListAccess",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssetModels"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "TwinMakerAccess",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:GetEntity",
        "iottwinmaker:CreateEntity",
        "iottwinmaker:UpdateEntity",
        "iottwinmaker>DeleteEntity",
        "iottwinmaker:ListEntities",
        "iottwinmaker:GetComponentType",
        "iottwinmaker:CreateComponentType",
        "iottwinmaker:UpdateComponentType",
        "iottwinmaker>DeleteComponentType",
        "iottwinmaker:ListComponentTypes"
      ],
      "Resource": [
        "arn:aws:iottwinmaker:*:*:workspace/*"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "iottwinmaker:linkedServices": [
            "IOTSITEWISE"
          ]
        }
      }
    }
  ]
}

```



變更	描述	日期
	<p>iottwinmaker:UpdateEntity, iottwinmaker&gt;DeleteEntity, iottwinmaker&gt;ListEntities, iottwinmaker:GetComponentType, iottwinmaker&gt;CreateComponentType, iottwinmaker:UpdateComponentType, iottwinmaker&gt;DeleteComponentType, iottwinmaker&gt;ListComponentTypes</p> <p>如需詳細資訊，請參閱 <a href="#">服務連結角色權限 AWS IoT TwinMaker</a>。</p>	
開始追蹤變更	開始追蹤其 AWS 受管理策略的變更。	2022 年 5 月 11 日

## AWS IoT TwinMaker 和介面 VPC 端端點 (AWS PrivateLink)

您可以在虛擬私有雲 (VPC) 和 AWS IoT TwinMaker 建立介面 VPC 端點之間建立私有連線。介面端點由其提供支援 [AWS PrivateLink](#)，您可以在沒有網際網路閘道、網路位址轉譯 (NAT) 裝置、VPN 連線或 AWS 直 Connect 連線的情況下私有存取 AWS IoT TwinMaker API。VPC 中的執行個體不需要公有 IP 位址即可與 AWS IoT TwinMaker API 通訊。您的 VPC 和 AWS IoT TwinMaker 不會離開 Amazon 網絡之間的流量。

每個介面端點都是由您子網路中的一或多個[彈性網路介面](#)表示。

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的介面虛擬私人雲端端點 ([AWS PrivateLink](#))。

## AWS IoT TwinMaker VPC 端點的考量

在為其設定介面 VPC 端點之前 AWS IoT TwinMaker，請先參閱 Amazon VPC 使用者指南中的[界面端點屬性和限制](#)。

AWS IoT TwinMaker 支援從您的 VPC 呼叫其所有 API 動作。

- 對於資料平面 API 作業，請使用下列端點：

```
data.iottwinmaker.region.amazonaws.com
```

資料平面 API 作業包括下列項目：

- [GetProperty價值](#)
  - [GetPropertyValueHistory](#)
  - [BatchPutPropertyValues](#)
- 對於控制平面 API 操作，請使用以下端點：

```
api.iottwinmaker.region.amazonaws.com
```

支援的控制平面 API 作業包括：

- [CreateComponent类型](#)
- [CreateEntity](#)
- [CreateScene](#)
- [CreateWorkspace](#)
- [DeleteComponent类型](#)
- [DeleteEntity](#)
- [DeleteScene](#)
- [DeleteWorkspace](#)
- [GetComponent类型](#)
- [GetEntity](#)
- [GetScene](#)
- [GetWorkspace](#)
- [ListComponent类型](#)
- [ListComponent类型](#)

- [ListEntities](#)
- [ListScenes](#)
- [ListTagsForResource](#)
- [ListWorkspaces](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateComponent](#)类型
- [UpdateEntity](#)
- [UpdateScene](#)
- [UpdateWorkspace](#)

## 為 AWS IoT TwinMaker 建立介面 VPC 端點

您可以使用 Amazon VPC 主控台或 AWS Command Line Interface (AWS CLI) 來為 AWS IoT TwinMaker 服務建立 VPC 端點。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[建立介面端點](#)。

建立使用下列服務名稱 AWS IoT TwinMaker 的 VPC 端點。

- 對於資料平面 API 作業，請使用下列服務名稱：

```
com.amazonaws.region.iottwinmaker.data
```

- 對於控制平面 API 操作，請使用以下服務名稱：

```
com.amazonaws.region.iottwinmaker.api
```

如果您為端點啟用私有 DNS，則可以使用該區域 AWS IoT TwinMaker 的預設 DNS 名稱向其發出 API 要求，例如 `iottwinmaker.us-east-1.amazonaws.com`。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[透過介面端點存取服務](#)。

AWS IoT TwinMaker PrivateLink 在下列地區支援：

- us-east-1

下列可用區域支援此 ControlPlane 服務：use1-az1、use1-az2、和 use1-az6。



下列可用區域支援此 DataPlane 服務：use1-az1、use1-az2、和 use1-az4。

- us-west-2

以下可用區域支援和 DataPlane 服務：usw2-az1、usw2-az2、和 usw2-az3。ControlPlane

- eu-west-1
- eu-central-1
- ap-southeast-1
- ap-southeast-2

如需可用區域的詳細資訊，請參閱[AWS 源的可用區域 ID- AWS Resource Access Manager](#)。

## 透 AWS IoT TwinMaker 過介面 VPC 端點存取

建立介面端點時，AWS IoT TwinMaker 會產生端點特定的 DNS 主機名稱，供您用來與之通訊。AWS IoT TwinMaker 私人 DNS 選項預設為啟用。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[使用私有託管區域](#)。

如果您為端點啟用私有 DNS，則可以 AWS IoT TwinMaker 透過下列其中一個 VPC 端點向其中一個發出 API 要求。

- 對於資料平面 API 作業，請使用下列端點。將##取代為您的 AWS 地區。

```
data.iottwinmaker.region.amazonaws.com
```

- 對於控制平面 API 操作，請使用以下端點。將##取代為您的 AWS 地區。

```
api.iottwinmaker.region.amazonaws.com
```

如果停用端點的私人 DNS，則必須執行下列動作才能 AWS IoT TwinMaker 透過端點存取：

- 在 API 要求中指定 VPC 人雲端端點 URL。
  - 對於資料平面 API 作業，請使用下列端點 URL。將 *vpc ## ID ##### VPC ## ID ##*。

```
vpc-endpoint-id.data.iottwinmaker.region.vpce.amazonaws.com
```

- 針對控制平面 API 作業，請使用下列端點 URL。將 *vpc ## ID ##### VPC ## ID ##*。

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- 停用主機前置詞插入。當您呼叫每個 API 作業時，AWS CLI 和 AWS SDK 會在服務端點前面加上各種主機前置詞。這會導致 AWS CLI 和 AWS SDK 在您指定 VPC 端點 AWS IoT TwinMaker 時產生無效的 URL。

#### Important

您無法在 AWS CLI 或中停用主機前置詞插入 AWS Tools for PowerShell。這表示如果您已停用私有 DNS，您將無法 AWS IoT TwinMaker 透過 VPC 端點使用 AWS CLI 或 AWS Tools for PowerShell 存取。如果您想要使用這些工具 AWS IoT TwinMaker 透過端點存取，請啟用私有 DNS。

有關如何在 SDK 中禁用主機前綴插入的 AWS 詳細信息，請參閱以下每個 SDK 的文檔部分：

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto3\)](#)
- [AWS SDK for Ruby](#)

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[透過介面端點存取服務](#)。

## 建立 VPC 端端點原則 AWS IoT TwinMaker

您可以將端點政策連接至控制 AWS IoT TwinMaker 存取權限的 VPC 端點。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。

- 可供執行動作的資源。

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[使用 VPC 端點控制對服務的存取](#)。

範例：用於動作的 VPC 端點原則 AWS IoT TwinMaker

以下是的端點策略範例 AWS IoT TwinMaker。連接到端點時，此政策會授予對所有資源 AWS 帳戶 iottwinmakeradmin 中 IAM 使用者列出 AWS IoT TwinMaker 動作 123456789012 的存取權。

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/role"
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:CreateEntity",
        "iottwinmaker:GetScene",
        "iottwinmaker:ListEntities"
      ]
    }
  ]
}
```

## 符合性驗證 AWS IoT TwinMaker

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱[AWS 服務 遵循規範計劃](#)方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱[AWS 規範計劃](#)。

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[下載中的報告中的 AWS Artifact](#)。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、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遵循法規：

- [安全性與合規性快速入門指南](#) — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- [在 Amazon Web Services 上架構 HIPAA 安全性與合規性](#) — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 應用程式。

**Note**

並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 [HIPAA 資格服務參照](#)。

- [AWS 合規資源AWS](#) — 此工作簿和指南集合可能適用於您的產業和所在地。
- [AWS 客戶合規指南](#) — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- [使用AWS Config 開發人員指南中的規則評估資源](#) — 此 AWS Config 服務會評估您的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 提供了內部安全狀態的全面視圖 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 controls reference](#)。
- [Amazon GuardDuty](#) — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。GuardDuty 可協助您滿足特定合規性架構所要求的入侵偵測需求，如 PCI DSS 等各種合規性需求。
- [AWS Audit Manager](#)— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

## 韌性 AWS IoT TwinMaker

AWS 全球基礎架構是圍繞 AWS 區域 和可用區域建立的。AWS 區域 提供多個實體分離和隔離的可用區域，這些區域與低延遲、高輸送量和高冗餘網路相連。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 全域基礎結構](#)。

除了 AWS 全球基礎架構之外，還 AWS IoT TwinMaker 提供多種功能，協助支援您的資料恢復能力和備份需求。

## 基礎架構安全 AWS IoT TwinMaker

作為受管服務，AWS IoT TwinMaker 受 [Amazon Web Services : 安 AWS 全流程概觀白皮書中所述的全球網路安全](#)程序保護。

您可以使用 AWS 已發佈的 API 呼叫透 AWS IoT TwinMaker 過網路進行存取。用戶端必須支援 Transport Layer Security (TLS) 1.2 或更新版本。建議使用 TLS 1.3 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 以產生暫時安全憑證以簽署請求。

# 端點和配額

## AWS IoT TwinMaker 端點和配額

您可以在[AWS 一般參考](#)中找到有關 AWS IoT TwinMaker 端點和配額的資訊。

- 如需服務端點的相關資訊，請參閱[AWS IoT TwinMaker 服務端點](#)。
- 如需配額的相關資訊，請參閱[AWS IoT TwinMaker 服務配額](#)。
- 如需 API 節流限制的相關資訊，請參閱 [AWS IoT TwinMaker API 節流限制](#)。

## 有關 AWS IoT TwinMaker 端點的其他資訊

若要以程式設計方式連線到 AWS IoT TwinMaker，請使用端點。如果您使用 HTTP 用戶端，則需要前置控制平面和資料平面 API，如下所示。但是，不必向 AWS SDK 和 AWS Command Line Interface 命令添加前綴，因為它們會自動添加必要的前綴。

- 使用控制平面 API 的 `api` 前置詞。例如 `api.iottwinmaker.us-west-1.amazonaws.com`。
- 使用資料平面 API 的 `data` 前置詞。例如 `data.iottwinmaker.us-west-1.amazonaws.com`。

# AWS IoT TwinMaker 使用者指南的文件歷程記錄

下表說明的文件發行版本AWS IoT TwinMaker。

變更	描述	日期
<a href="#">新服務連結角色和新的 IAM 政策</a>	AWS IoT TwinMaker新增服務連結角色，稱為 <a href="#">AWSServiceRoleForIoT</a> TwinMaker。AWS IoT TwinMaker添加了這個新的服務鏈接角色，AWS IoT TwinMaker以允許調用其他AWS服務並代表您同步其資源。新的 <code>AWSIoTServiceRolePolicy</code> IAM 政策已附加至此角色，而且該政策授AWS IoT TwinMaker予呼叫其他AWS服務並代表您同步其資源的權限。	2023 年 11 月 17 日
<a href="#">初始版本</a>	AWS IoT TwinMaker使用者指南的初始版本	2021 年 11 月 30 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。