aws

使用者指南

AWS IoT TwinMaker

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

AWS IoT TwinMaker: 使用者指南

Copyright © 2025 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
实体组件模型 3
視覺效果 4
入門 AWS IoT TwinMaker 6
建立和管理 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 Amazon S3 儲存貯體 8
內建連接器的許可 10
連接器到外部資料來源的許可 13
修改工作區 IAM 角色以使用 Athena 資料連接器 14
建立工作區 16
建立您的第一個實體 18
設定 AWS 帳戶 21
註冊 AWS 帳戶 21
建立具有管理存取權的使用者 22
使用和建立元件類型 24
內建元件類型 24
AWS IoT TwinMaker 元件類型的核心功能 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
大量操作 34
重要概念和術語 34
AWS IoT TwinMaker metadataTransferJob 功能 35
執行大量匯入和匯出操作 36
metadataTransferJob 先決條件 36

IAM 許可 36
執行大量操作 40
錯誤處理 43
匯入中繼資料範本 44
AWS IoT TwinMaker metadataTransferJob 範例 47
AWS IoT TwinMaker 中繼資料傳輸任務結構描述 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
在將資源匯入 之前最佳化資源 AWS IoT TwinMaker 109
中效能的最佳實務 AWS IoT TwinMaker 110
進一步了解 110

在 中上傳資源 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 AWS IoT TwinMaker 場景中使用 3D 導覽 112
新增固定攝影機 113
增強型編輯 114
場景物件的目標放置 114
子模型選擇 115
在場景階層中編輯實體 115
將註釋新增至實體 116
將浮水印新增至標籤 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 3D 模型 133
在場景中使用 3D 圖磚 133
動態場景 135
靜態與動態場景 135
場景元件類型和實體 136
動態場景概念 137
AWS IoT TwinMaker 應用程序包集成 138
切換 AWS IoT TwinMaker 定價模式 139
知識圖表 141
AWS IoT TwinMaker 知識圖表核心概念 141
使用知識圖表 142
產生場景圖表 144
AWS IoT TwinMaker 場景圖表先決條件 145
在場景中繫結 3D 節點 146
建立 Web 應用程式 147
知識圖表 Grafana 面板 149
AWS IoT TwinMaker 查詢編輯器先決條件 149
知識圖表 Grafana 許可 150
知識圖表其他資源 154
與 的資產同步 AWS IoT SiteWise 167
搭配 使用資產同步 AWS IoT SiteWise 167

使用自訂工作區 167
使用 IoTSiteWiseDefaultWorkspace 173
自訂和預設工作區之間的差異 173
從 同步的資源 AWS IoT SiteWise 174
自訂和預設工作區 174
僅限預設工作區 175
資源未同步 176
在 中使用同步實體和元件類型 AWS IoT TwinMaker 176
分析同步狀態和錯誤 177
同步任務狀態 177
刪除同步任務 178
資產同步限制 180
設定 Grafana 儀表板 181
CORS 組態 181
設定您的 Grafana 環境 183
Amazon Managed Grafana 183
自我管理的 Grafana 184
建立儀表板角色 184
建立 IAM 政策 185
從邊緣上傳影片 188
新增更多許可 188
建立 Grafana Dashboard IAM 角色 190
建立 AWS IoT TwinMaker 影片播放器政策 190
縮小對 資源的存取範圍 192
縮小 GET 許可的範圍 192
Scope down AWS IoT SiteWise BatchPutAssetPropertyValue 許可 194
將警示連接至 Grafana 儀表板 196
AWS IoT SiteWise 警示組態先決條件 196
定義 AWS IoT SiteWise 警示元件 IAM 角色 196
透過 AWS IoT TwinMaker API 查詢和更新 198
設定警示的 Grafana 儀表板 199
使用 Grafana 儀表板進行警示視覺化 201
馬特港集成 204
整合概述 205
物流端整合先決條件 206
事業端口 SDK 憑證 207

將馬特港認證存儲在 AWS Secrets Manager 208
場景中的馬特港掃描 AWS IoT TwinMaker 211
Grafana 儀表板中的 AWS IoT TwinMaker 馬特波特 216
與應用程序套件 AWS IoT 集成 216
將影片串流至 AWS IoT TwinMaker 217
使用 Kinesis 影片串流的邊緣連接器在 中串流影片 AWS IoT TwinMaker 217
先決條件 217
建立 AWS IoT TwinMaker 場景的影片元件 217
將影片和中繼資料從 Kinesis 影片串流新增至 Grafana 儀表板 218
使用快速AWS IoT TwinMaker連結資源庫 220
記錄和監控 221
使用亞馬遜 CloudWatch 指標監控 221
指標 222
使用 AWS CloudTrail 記錄 API 呼叫 224
AWS IoT TwinMaker中的資訊 CloudTrail 224
安全 226
資料保護 226
靜態加密 227
傳輸中加密 227
身分和存取權管理 228
目標對象 228
使用身分驗證 229
使用政策管理存取權 231
AWS IoT TwinMaker 如何使用 IAM 233
身分型政策範例 239
故障診斷 241
使用服務連結角色 243
AWS 受管政策 245
VPC 端點 (AWS PrivateLink) 249
AWS IoT TwinMaker VPC 端點的考量事項 249
為 AWS IoT TwinMaker建立介面 VPC 端點 251
AWS IoT TwinMaker 透過界面 VPC 端點存取 252
為 建立 VPC 端點政策 AWS IoT TwinMaker 253
合規驗證 254
恢復能力 255
基礎設施安全性 255

端點和配額 256
AWS IoT TwinMaker 端點和配額 256
其他端點資訊 256
文件歷史紀錄 257
..... cclviii

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。

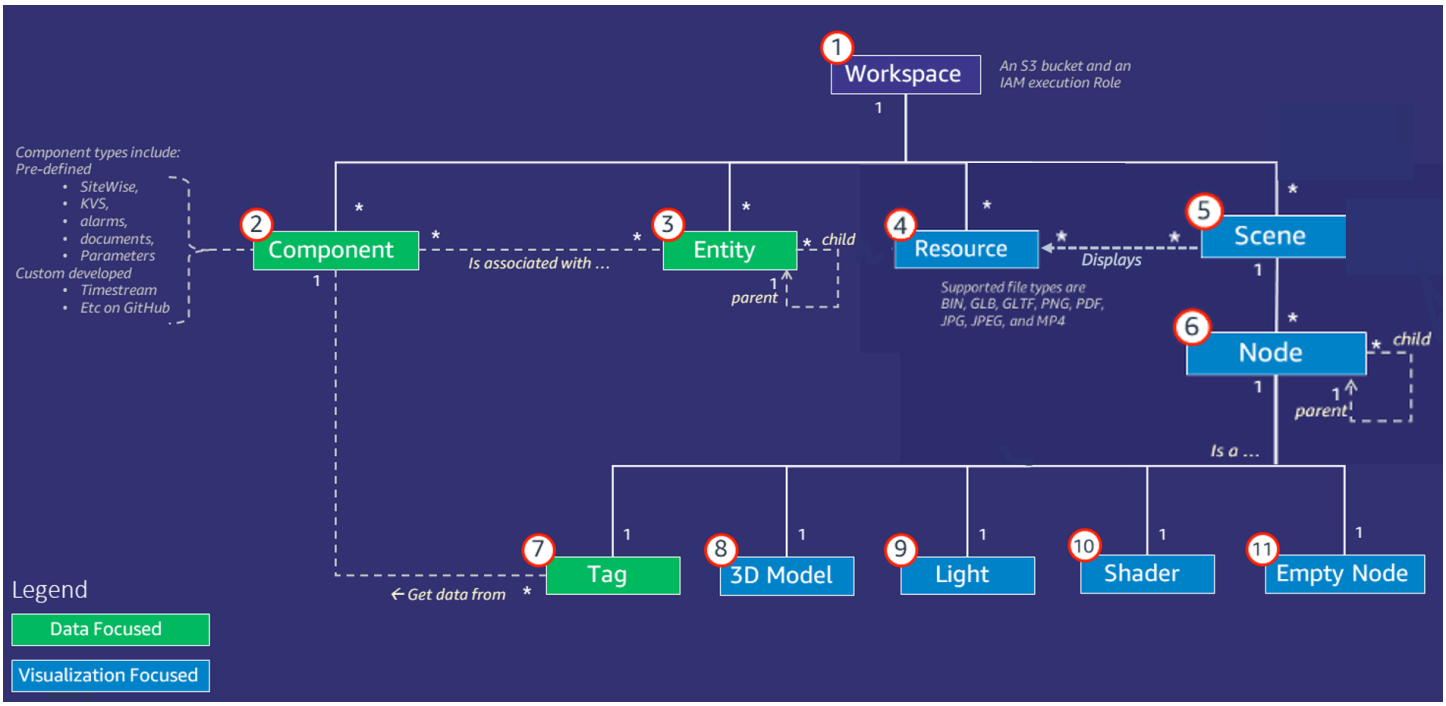
  知識圖和連接器共同提供單一介面，用於查詢不同位置的資料。
- 場景作曲家： AWS IoT TwinMaker 控制台提供場景合成工具，用於在 3D 中創建場景。您可以上傳先前建立的 3D/CAD 模型，並針對網頁顯示進行最佳化，並轉換為 .gltf 或 .glb 格式。然後使用場景管理員在單一場景中放置多個模型，建立其作業的視覺表現法。

  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。

实体组件模型

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。

入門 AWS IoT TwinMaker

本節中的主題說明如何執行下列動作。

- 建立和設定新的工作區。
- 建立實體並將其新增元件。

事前準備：

若要建立第一個工作區和場景，您需要下列 AWS 資源。

- AWS 帳戶。
- 的 IAM 服務角色 AWS IoT TwinMaker。當您在AWS IoT TwinMaker 主控台中建立新的 AWS IoT TwinMaker 工作區時，預設會自動產生此角色。

  如果您不選擇讓 AWS IoT TwinMaker 自動建立新的 IAM 服務角色，則必須指定您已建立的服務角色。

  如需建立和管理此服務角色的說明，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連接政策，授予 服務讀取和寫入 Amazon S3 儲存貯體的許可。 AWS IoT TwinMaker 使用此角色代表您存取其他 服務。您也需要在此角色與 之間指派信任關係，AWS IoT TwinMaker 以便服務可以擔任該角色。如果您的雙身與其他 AWS 服務互動，也請為這些服務新增必要的許可。

主題

- 建立和管理 的服務角色 AWS IoT TwinMaker
- 建立工作區
- 建立您的第一個實體
- 設定 AWS 帳戶

建立和管理 的服務角色 AWS IoT TwinMaker

AWS IoT TwinMaker 要求您使用服務角色，以允許它代表您存取其他服務中的資源。此角色必須與 建立信任關係 AWS IoT TwinMaker。建立工作區時，您必須將此角色指派給工作區。本主題包含範例政策，說明如何設定常見案例的許可。

指派信任

下列政策會在您的角色與 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。

JSON

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

Amazon S3 許可

下列政策可讓您的角色從 Amazon S3 儲存貯體讀取和刪除和寫入。Workspaces 會將資源存放在 Amazon S3 中，因此所有工作區都需要 Amazon S3 許可。

JSON

```
{
  "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**
>
> 當您建立工作區時， 會在 Amazon S3 儲存貯體中 AWS IoT TwinMaker 建立檔案，指出工作區正在使用該檔案。當您刪除工作區時，此政策會提供刪除該檔案的 AWS IoT TwinMaker 許可。
>
> AWS IoT TwinMaker 會放置與您工作區相關的其他物件。刪除工作區時，您有責任刪除這些物件。

將許可指派給特定 Amazon S3 儲存貯體

在 AWS IoT TwinMaker 主控台中建立工作區時，您可以選擇讓 AWS IoT TwinMaker 為您建立 Amazon S3 儲存貯體。您可以使用下列 AWS CLI 命令找到此儲存貯體的相關資訊。

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

下列範例顯示此命令輸出的格式。

```
{
    "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"
}
```

若要更新您的政策以指派特定 Amazon S3 儲存貯體的許可，請使用儲存####的值。

下列政策可讓您的角色從特定 Amazon S3 儲存貯體讀取和刪除和寫入。

JSON

```
{
  "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_*"
      ]
    }
  ]
}
```

內建連接器的許可

如果您的工作區使用內建連接器與其他 AWS 服務互動，您必須在此政策中包含這些服務的許可。如果您使用 com.amazon.iotsitewise.connector 元件類型，則必須包含 的許可 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，您必須授予角色許可，才能執行在元件類型中實作 函數的 Lambda 函數。如需詳細資訊，請參閱???。

下列範例示範如何將 包含在政策 AWS IoT SiteWise 中。

JSON

```
{
  "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 com.amazon.iotsitewise.connector 元件類型，且需要從中讀取屬性資料 AWS IoT SiteWise，則必須在政策中包含下列許可。

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

```
    ],
    "Effect": "Allow"
},
...
```

如果您使用 com.amazon.iotsitewise.connector 元件類型且需要寫入屬性資料 AWS IoT SiteWise，則必須在政策中包含下列許可。

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

如果您使用 com.amazon.iotsitewise.connector.edgevideo 元件類型，您必須包含 AWS IoT SiteWise 和 Kinesis Video Streams 的許可。下列範例政策示範如何在政策中包含 AWS IoT SiteWise 和 Kinesis Video Streams 許可。

```
...
{
    "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"
},
...
```

連接器到外部資料來源的許可

如果您建立的元件類型使用連接到外部資料來源的 函數，您必須授予您的服務角色許可，才能使用實作該函數的 Lambda 函數。如需建立元件類型和函數的詳細資訊，請參閱 ???。

下列範例為您的服務角色提供使用 Lambda 函數的許可。

JSON

```
{
  "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 AWS CLI、 和 IAM API 來建立角色、指派政策和信任關係的詳細資訊，請參閱建立角色以將許可委派給 AWS 服務。

修改工作區 IAM 角色以使用 Athena 資料連接器

若要使用 AWS IoT TwinMaker Athena 表格式資料連接器，您必須更新您的 AWS IoT TwinMaker 工作區 IAM 角色。將下列許可新增至工作區 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。若要將 Athena 與其他資料來源搭配使用，您必須為 Athena 設定 IAM 角色，請參閱 Athena 中的 Identity and Access Management。

```
{
    "Effect": "Allow",
    "Action": [
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "athena:GetTableMetadata",
        "athena:GetWorkGroup",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution"
    ],
    "Resource": [
        "athena resouces arn"
    ]
},// Athena permission
{
    "Effect": "Allow",
    "Action": [
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetDatabase",
        "glue:GetDatabases"
    ],
    "Resource": [
        "glue resouces 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 Athena IAM 組態的詳細資訊，請參閱 Athena 中的 Identity and Access Management。

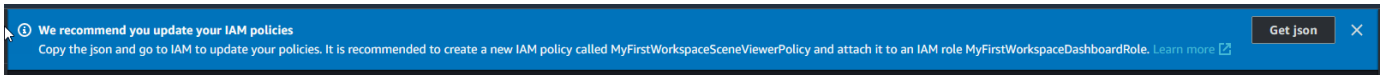
建立工作區

若要建立和設定您的第一個工作區，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全功能工作區，請嘗試範例 Github 儲存庫中的AWS IoT TwinMaker 範例設定。

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. （選用）新增工作區的描述。
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儲存貯體。
6. 在執行角色下，選擇自動產生新角色或您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， 會將政策 AWS IoT TwinMaker 連接至角色，以授予新服務角色存取其他服務的許可 AWS ，包括讀取和寫入您在上一個步驟中指定的 Amazon S3 儲存貯體的許可。如需指派許可給此角色的詳細資訊，請參閱 ???。
7. 選擇建立工作區。下列橫幅會出現在工作區頁面頂端。



8. 選擇取得 json。建議您將看到的 IAM 政策新增至為檢視 Grafana 儀表板的使用者和帳戶 AWS IoT TwinMaker 建立的 IAM 角色。此角色的名稱遵循此模式：*Workspace-nameDashboardRole*DashboardRole ，如需如何建立政策並將其連接至角色的指示，請參閱修改角色許可政策（主控台）。

   下列範例包含要新增至儀表板角色的政策。

   JSON

```
{
  "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": "*"
    }
```

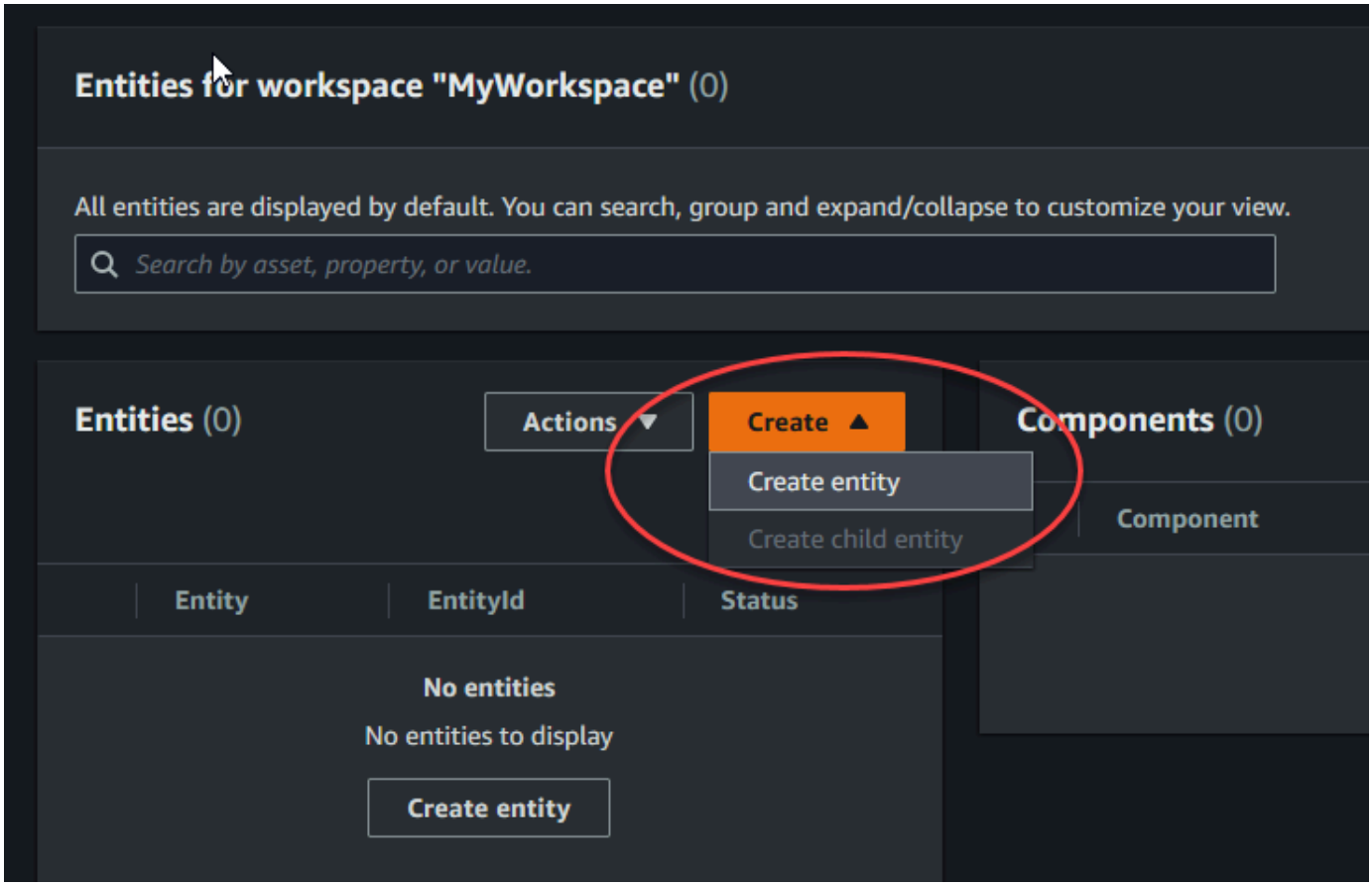
```
  ]
}
```

您現在可以使用第一個實體開始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，然後在左側窗格中選擇實體。
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. （選用）輸入實體的描述。
5. 選擇建立實體，

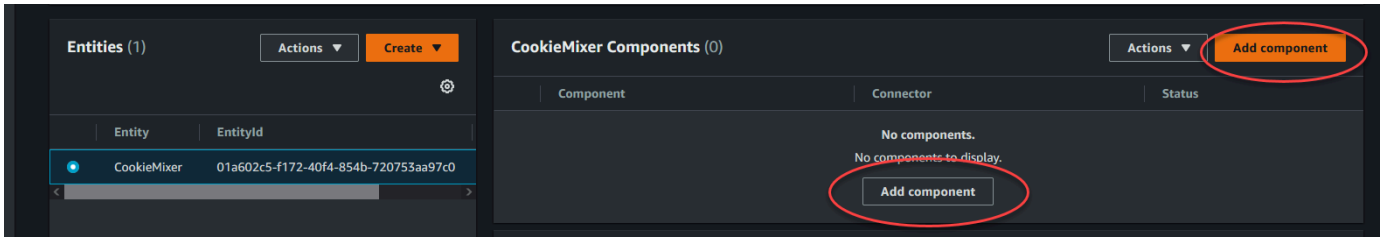
實體包含工作區中每個項目的資料。您可以透過新增 components 將資料放入實體。 AWS IoT TwinMaker 提供下列內建元件類型。

- 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。
- 警示：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屬性。
- Edge Connector for Kinesis Video Streams AWS IoT Greengrass：從 Edge Connector for KVS 提取影片資料 AWS IoT Greengrass。如需詳細資訊，請參閱AWS IoT TwinMaker 影片整合。

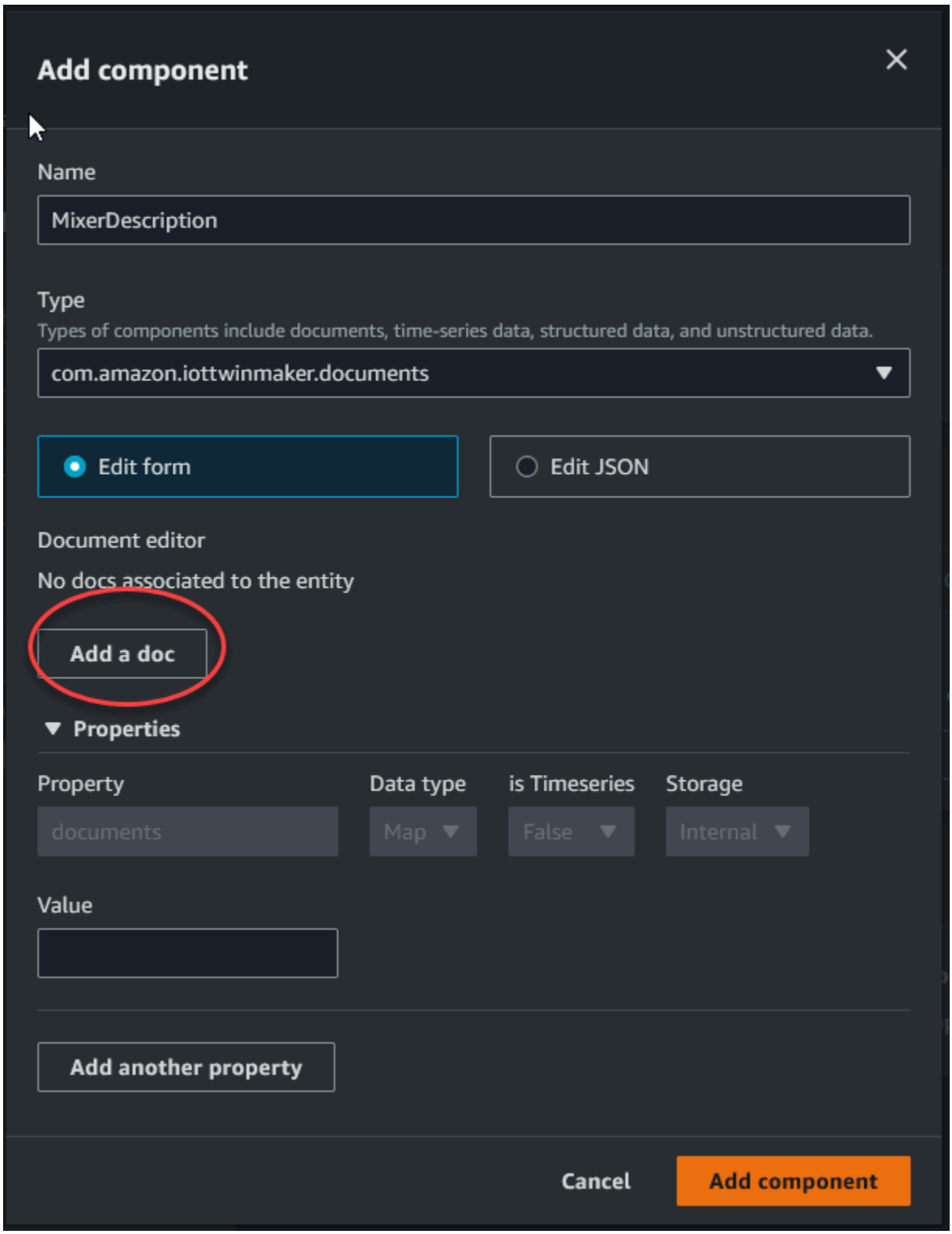
您可以在左側窗格中選擇元件類型，以查看這些元件類型及其定義。您也可以在元件類型頁面上建立新的元件類型。如需建立元件類型的詳細資訊，請參閱 使用和建立元件類型。

在此範例中，我們會建立簡單的文件元件，以新增您實體的描述性資訊。

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 混音器實體，我們會在**`MixerDescription`**名稱欄位中輸入 。



3. 選擇新增文件，然後輸入文件名稱和外部 URL 的值。使用 文件元件，您可以存放包含實體重要資訊的外部 URLs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。

設定 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 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或簡訊，並在電話鍵盤輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 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或簡訊，並在電話鍵盤輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 在註冊程序完成後， 會傳送確認電子郵件給您。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。

建立具有管理存取權的使用者

註冊 後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center和建立 管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS Management Console身分登入 。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)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者 （主控台） 啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

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 Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center《 使用者指南》中的使用預設值設定使用者存取權 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 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 Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》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 Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

使用和建立元件類型

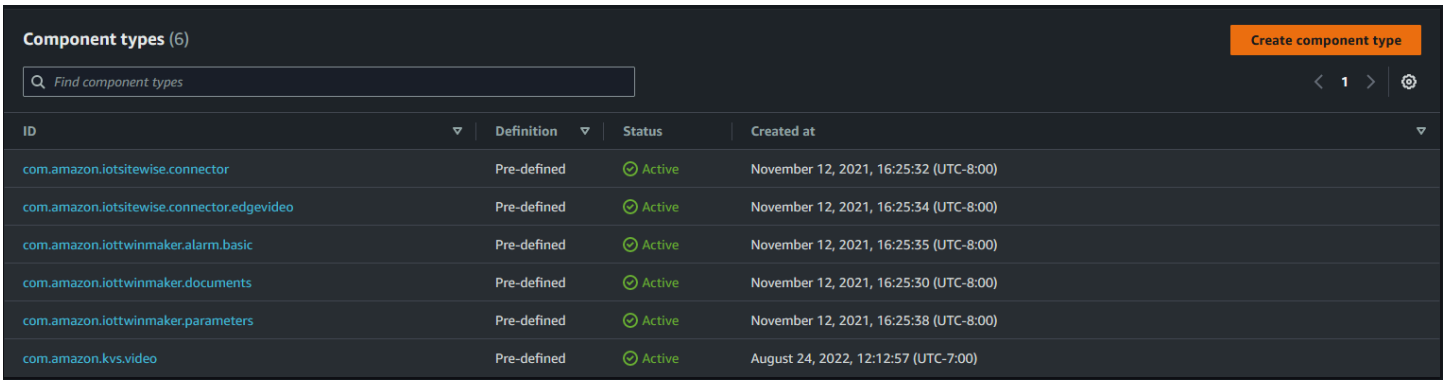
本主題將逐步引導您瞭解用來建立 AWS IoT TwinMaker 元件類型的值和結構。它會示範如何建立要求物件，您可以傳遞至 CreateComponentTypeAPI 或使用 AWS IoT TwinMaker 主控台中的元件類型編輯器。

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



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，GetPropertyValueHistoryAPI 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，GetPropertyValueHistoryAPI 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、DataType對映、清單和測量單位) 的物件。
`defaultValue`	指DataValue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DataConnector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 主控台中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。您也可以在邏輯上使用此資訊，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 大量操作

使用metadataTransferJob 大規模傳輸和管理 AWS IoT TwinMaker 資源。metadataTransferJob 可讓您在 AWS IoT TwinMaker 和 Amazon S3 之間執行大量操作 AWS IoT SiteWise 和傳輸資源。

您可以在下列案例中使用大量操作：

- 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。
- 將您的資產大量匯入至 AWS IoT TwinMaker 和 AWS IoT SiteWise。
- 從現有內科檔案大量匯入 AWS IoT TwinMaker 實體，例如 `revit`或 `BIM` 檔案。

主題

- 重要概念和術語
- 執行大量匯入和匯出操作
- AWS IoT TwinMaker 中繼資料傳輸任務結構描述

重要概念和術語

AWS IoT TwinMaker 大量操作使用以下概念和術語：

- 匯入：將資源 AWS IoT TwinMaker 移至工作區的動作。例如，從本機檔案、Amazon S3 儲存貯體中的檔案，或從 AWS IoT SiteWise 到 AWS IoT TwinMaker 工作區。
- 匯出：將資源從 AWS IoT TwinMaker 工作區移至本機電腦或 Amazon S3 儲存貯體的動作。
- 來源：您要從中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 工作區是匯出來源。
- 目的地：您要將資源移動到的所需位置。

  例如，Amazon S3 儲存貯體是匯出目的地，而 AWS IoT TwinMaker 工作區是匯入目的地。
- AWS IoT SiteWise 結構描述：用於匯入和匯出資源的結構描述 AWS IoT SiteWise。
- AWS IoT TwinMaker 結構描述：用於匯入和匯出資源的結構描述 AWS IoT TwinMaker。
- AWS IoT TwinMaker 最上層資源：現有 APIs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。特別是元件。
- 中繼資料：成功匯入或匯出 AWS IoT TwinMaker AWS IoT SiteWise 和資源所需的金鑰資訊。
- metadataTransferJob：執行 時建立的物件`CreateMetadataTransferJob`。

AWS IoT TwinMaker metadataTransferJob 功能

本主題說明執行大量操作時 AWS IoT TwinMaker 遵循的行為 – 如何處理metadataTransferJob。它還說明如何使用轉移資源所需的中繼資料來定義結構描述。 AWS IoT TwinMaker 大量操作支援下列功能：

- 最上層資源建立或取代： AWS IoT TwinMaker 將建立新的資源，或取代由資源 ID 唯一識別的所有現有資源。

  例如，如果 實體存在於系統中，實體定義將由 `Entity`金鑰下範本中定義的新實體定義取代。
- 子資源建立或取代：

  從 EntityComponent 層級，您只能建立或取代元件。實體必須已存在，否則，動作會產生 ValidationException。

  從屬性或關係層級，您只能建立或取代屬性或關係，且包含 EntityComponent 的 必須已存在。
- 子資源刪除：

  AWS IoT TwinMaker 也支援子資源刪除。子資源可以是元件、屬性或關係。

  如果您想要刪除元件，您必須從實體層級執行。

  如果您想要刪除屬性或關係，您必須從實體或 EntityComponent 層級執行。

  若要刪除子資源，請更新更高層級的資源，並省略子資源的定義。
- 不刪除頂層資源： AWS IoT TwinMaker 永遠不會刪除頂層資源。最上層資源是指實體或 ComponentType。
- 一個範本中沒有相同頂層資源的子資源定義：

  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、屬性或關係中使用相同的 ID。

  如果 EntityComponent 中使用 entityId 或 componentName 組合，您就無法在 EntityComponent、屬性或關係中使用相同的組合。

如果在屬性或關係中使用 entityId、 componentName、 propertyName 組合，則無法在屬性或關係中使用相同的組合。

- ExternalId 是選用的 AWS IoT TwinMaker：ExternalId 可用來協助您識別資源。

執行大量匯入和匯出操作

本主題涵蓋如何執行大量匯入和匯出操作，以及如何處理傳輸任務中的錯誤。它提供使用 CLI 命令的傳輸任務範例。

AWS IoT TwinMaker API 參考包含 CreateMetadataTransferJob 和其他 API 動作的相關資訊。

主題

- metadataTransferJob 先決條件
- IAM 許可
- 執行大量操作
- 錯誤處理
- 匯入中繼資料範本
- AWS IoT TwinMaker metadataTransferJob 範例

metadataTransferJob 先決條件

請先完成下列先決條件，再執行metadataTransferJob：

- 建立 AWS IoT TwinMaker 工作區。工作區可以是metadataTransferJob。如需建立工作區的資訊，請參閱 建立工作區。
- 建立 Amazon S3 儲存貯體以存放資源。如需使用 Amazon S3 的詳細資訊，請參閱什麼是 Amazon S3？

IAM 許可

當您執行大量操作時，您需要建立具有許可的 IAM 政策 AWS IoT TwinMaker，以允許在 Amazon S3 AWS IoT SiteWise和本機機器之間交換 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 TwinMaker AWS IoT SiteWise 和 Amazon S3 的政策陳述式列於此處：

- AWS IoT TwinMaker 政策：

  JSON

```
{
    "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：

JSON

```
{
    "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 政策範圍限定為僅存取單一 Amazon S3 儲存貯體，請參閱下列政策。

Amazon S3 單一儲存貯體範圍政策

```
{
    "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metadataTransferJob

若要控制使用者可存取的任務類型，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 AWS IoT TwinMaker 匯入和匯出在 Amazon S3 之間傳輸資源的任務。

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

執行大量操作

本節說明如何執行大量匯入和匯出操作。

將資料從 Amazon S3 匯入至 AWS IoT TwinMaker

1. 使用 AWS IoT TwinMaker metadataTransferJob 結構描述指定您要傳輸的資源。建立結構描述檔案並將其存放在 Amazon S3 儲存貯體中。

   如需範例結構描述，請參閱 匯入中繼資料範本。
2. 建立請求內文並將其儲存為 JSON 檔案。請求內文會指定傳輸任務的來源和目的地。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。

以下是請求內文的範例：

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

記錄您提供請求內文的檔案名稱，您需要在下一個步驟中使用它。在此範例中，請求內文名為 `createMetadataTransferJobImport.json`。

3. 執行下列 CLI 命令來叫用 `CreateMetadataTransferJob`（將 input-json 檔案名稱取代為您提供請求內文的名稱）：

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

這會建立metadataTransferJob，並開始轉移所選資源的程序。

從 匯出資料 AWS IoT TwinMaker 至 Amazon S3

1. 使用適當的篩選條件建立 JSON 請求內文，以選擇要匯出的資源。在此範例中，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:::amzn-s3-demo-bucket"
        }
    }
}
```

`filters` 陣列可讓您指定要匯出的資源。在此範例中，我們會依 `entity`、 和 進行篩選`componentType`。

請務必將 AWS IoT TwinMaker 工作區指定為來源，並將 Amazon S3 儲存貯體指定為中繼資料傳輸任務的目的地。

儲存您的請求內文並記錄檔案名稱，您會在下一個步驟中需要它。在此範例中，我們將請求內文命名為 `createMetadataTransferJobExport.json`。

2. 執行下列 CLI 命令來叫用 `CreateMetadataTransferJob`（將 input-json 檔案名稱取代為您提供請求內文的名稱）：

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

這會建立metadataTransferJob，並開始轉移所選資源的程序。

若要檢查或更新傳輸任務的狀態，請使用下列命令：

- 若要取消任務，請使用 CancelMetadataTransferJob API 動作。當您呼叫 CancelMetadataTransferJob 時，API 只會取消執行中的metadataTransferJob，而且任何已匯出或匯入的資源都不會受到此 API 呼叫的影響。
- 若要擷取特定任務的資訊，請使用 GetMetadataTransferJob API 動作。

  或者，您可以使用下列 CLI 命令，在現有的傳輸任務上呼叫 GetMetadataTransferJob：

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

  如果您在不存在的 AWS IoT TwinMaker 匯入或匯出任務上呼叫 GetMetadataTransferJob，您會收到`ResourceNotFoundException`錯誤回應。
- 若要列出目前的任務，請使用 ListMetadataTransferJobs API 動作。

  以下是將 ListMetadataTransferJobs 呼叫 AWS IoT TwinMaker 為 destinationType 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 destinationType 參數的值，以符合匯入或匯出任務的來源和目的地。

如需叫用這些 API 動作的 CLI 命令範例，請參閱 AWS IoT TwinMaker metadataTransferJob 範例。

如果您在傳輸任務期間遇到任何錯誤，請參閱 錯誤處理。

錯誤處理

建立並執行傳輸任務之後，您可以呼叫 GetMetadataTransferJob 來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
```

一旦看到任務狀態變成 COMPLETED，您就可以驗證任務的結果。GetMetadataTransferJob 會傳回名為 的物件MetadataTransferJobProgress，其中包含下列欄位：

- failedCount：表示傳輸過程中失敗的資源數量。
- skippedCount：表示轉移過程中略過的資源數量。
- succeededCount：表示傳輸程序期間成功的資源數量。
- totalCount：表示傳輸程序中涉及的資源總數。

此外，會傳回 reportUrl 元素，其中包含預先簽章的 URL。如果您的傳輸任務發生錯誤，您想要進一步調查，則可以使用此 URL 下載完整的錯誤報告。

匯入中繼資料範本

您可以使用單一大量匯入操作匯入許多元件、 componentTypes 或實體。本節中的範例示範如何執行此操作。

template: Importing entities

將下列範本格式用於匯入實體的任務：

```
{
  "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

將下列範本格式用於匯入 componentTypes 的任務：

```
{
  "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

將下列範本格式用於匯入元件的任務：

```
{
```

```
  "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 範例

使用下列命令來管理您的中繼資料傳輸：

- CreateMetadataTransferJob API 動作。

  CLI 命令範例：

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

- 若要取消任務，請使用 CancelMetadataTransferJob API 動作。

  CLI 命令範例：

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

  當您呼叫 CancelMetadataTransferJob 時，只會取消特定的中繼資料傳輸任務，而且任何已匯出或匯入的資源都不會受到影響。

- 若要擷取特定任務的資訊，請使用 GetMetadataTransferJob API 動作。

  CLI 命令範例：

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

- 若要列出目前的任務，請使用 ListMetadataTransferJobs API 動作。

  您可以使用 JSON 檔案篩選 ListMetadataTransferJobs 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條件：

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

     儲存它並記錄檔案名稱，在輸入 CLI 命令時需要它。

  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 中繼資料傳輸任務結構描述

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

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTTwinMaker",
  "description": "Metadata transfer job resource schema for IoTTwinMaker",
  "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
    }
  }
}
```

以下範例會建立名為 的新 componentType，`component.type.intial`並建立名為 的實體`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 GetPropertyValueHistoryAPI 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`並`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`properties`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 GetPropertyValueHistoryAPI 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 GetPropertyValueHistoryAPI 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 GetPropertyValueHistoryAPI 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 GetPropertyValueAPI 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。

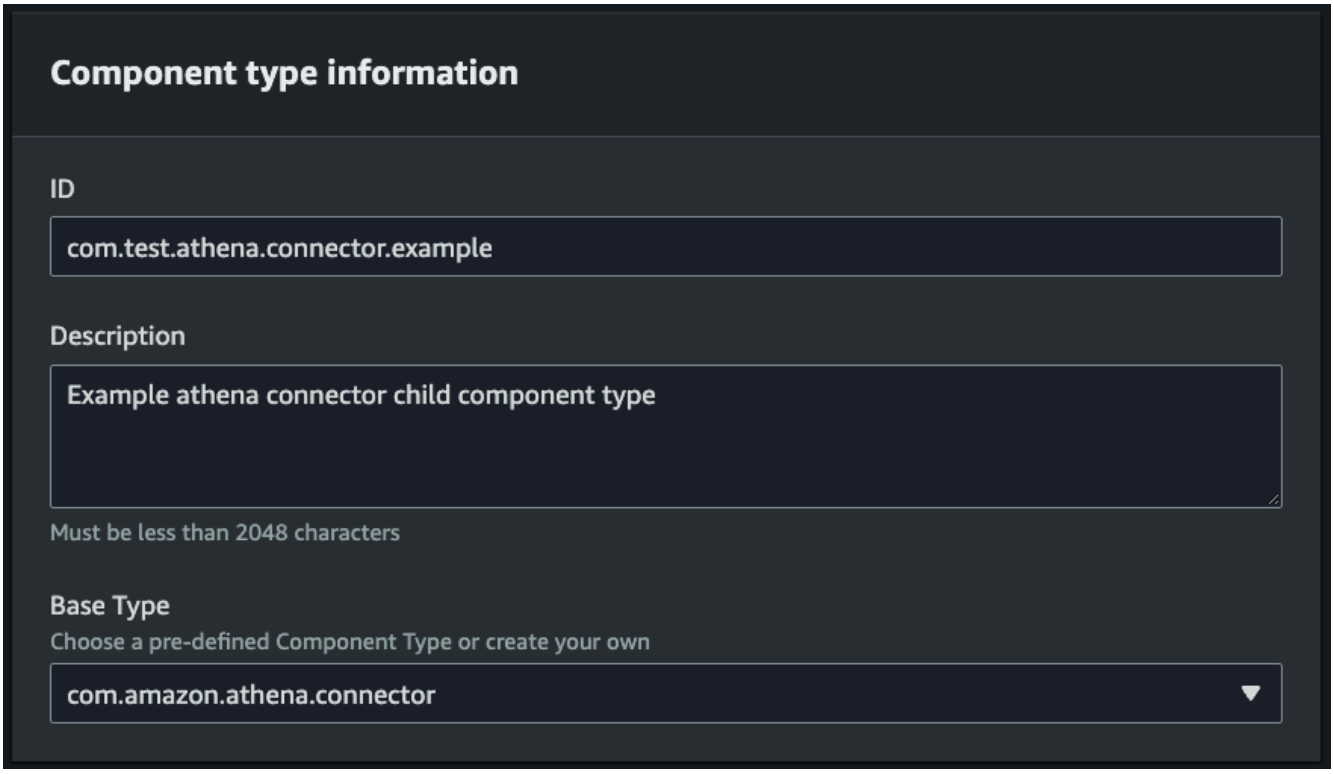
使用 Athena 資料連接器

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。

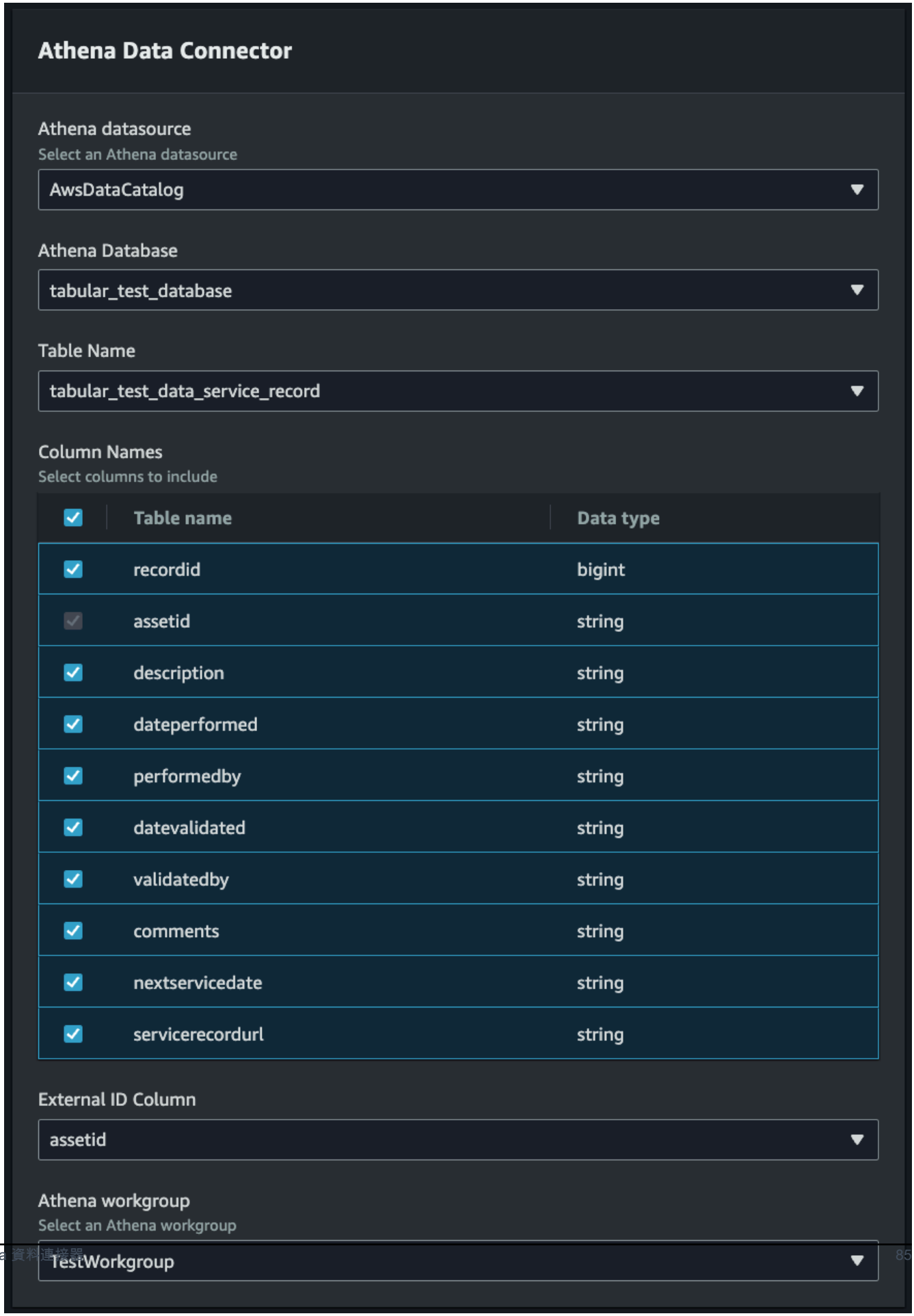
使用 Athena 資料連接器建立元件類型

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。



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 A thena 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
Table name
Data type
recordid
bigint
assetid
string
description
string
dateperformed
string
performedby
string
datevalidated
string
validatedby
string
comments
string
nextservicedate
string
serviceRecordUrl
string
External ID Column
assetid
Athena workgroup
Select an Athena workgroup
TestWorkgroup

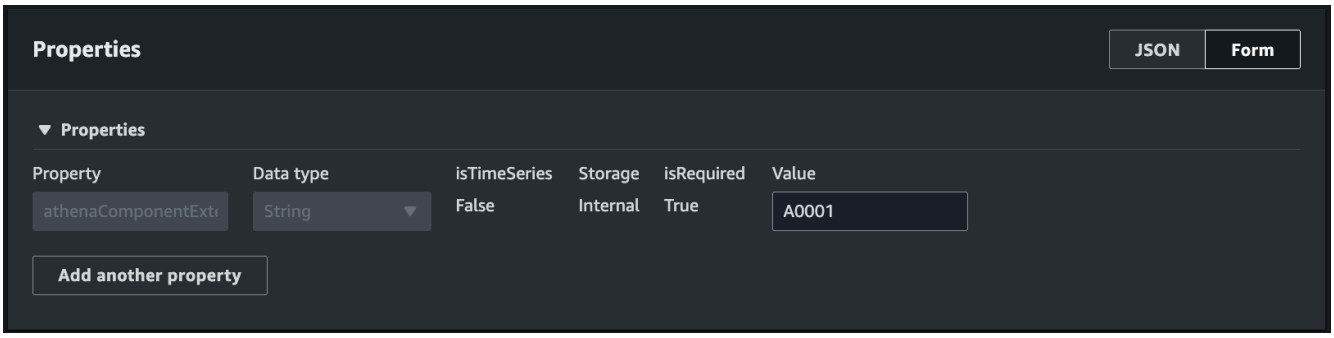
9. (選擇性) 將 AWS 標籤新增至這些資源，以便您可以將它們分組和組織。
10. 選擇「建立元件類型」以完成建立元件類型。

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。



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。 GetPropertyValue 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

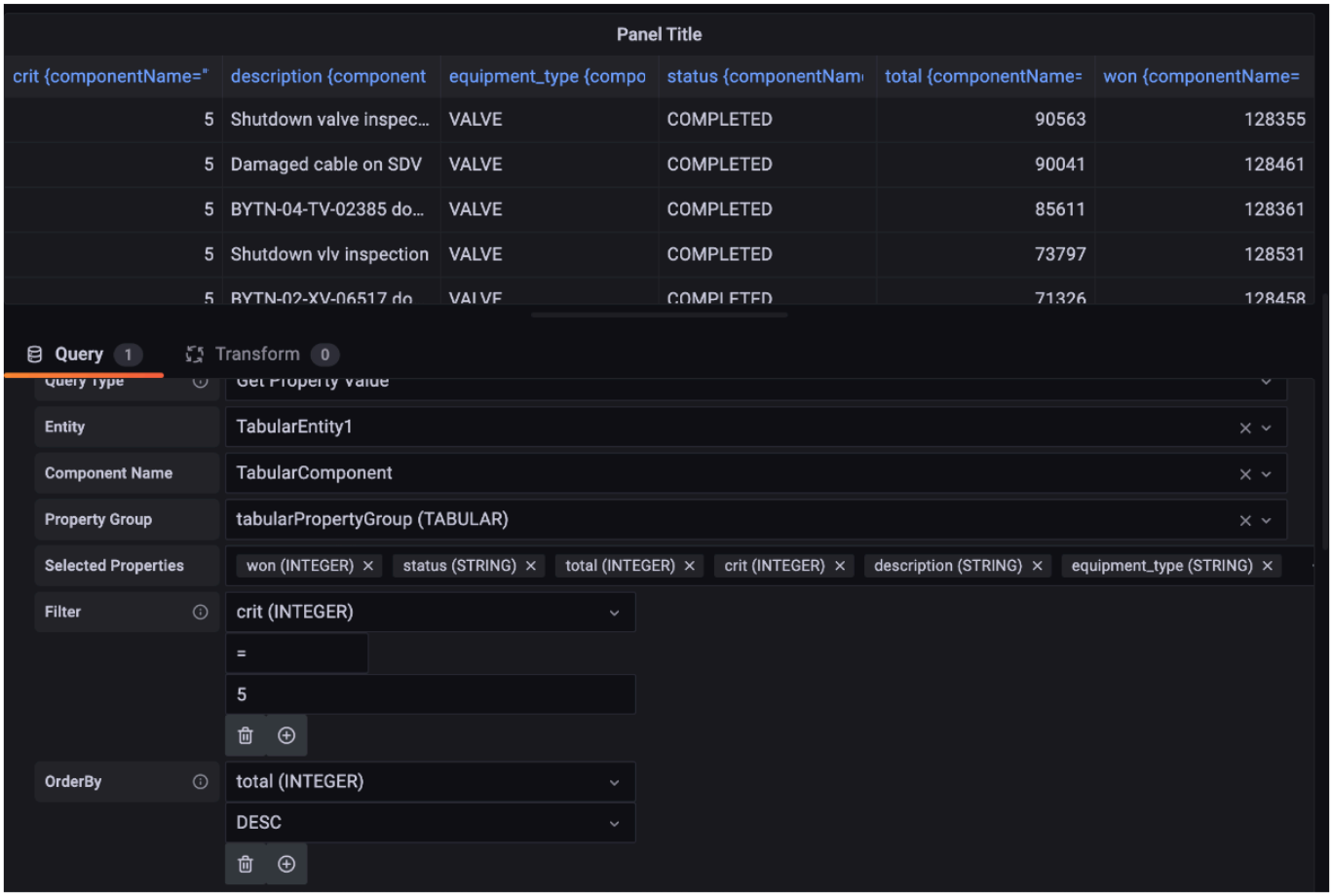
在設定 Grafana 面板以視覺化 Athena 表格資料之前，請先檢閱下列先決條件：

- 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，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。



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

1 2 3 4 5 6 7 ... 100

TelemetryAssetId	TelemetryAssetType	measure_name	time	measure_value::varchar	measure_value::double
Mixer_22_680b5b8e-1afe-4a77-87ab-834fbe5ba01e	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-834fbe5ba01e	Mixer	RPM	2022-04-19 00:28:00.241000000	-	59.9421195983887
Mixer_24_7ff0b75b-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_7ff0b75b-f0fa-43f0-bc89-b96337586d00	Mixer	RPM	2022-04-19 00:28:00.241000000	-	60.4532585144043
Mixer_15_0bb566cd-d6f3-4804-9fe1-7d2abcad82d0	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屏幕截圖中顯示的結果。

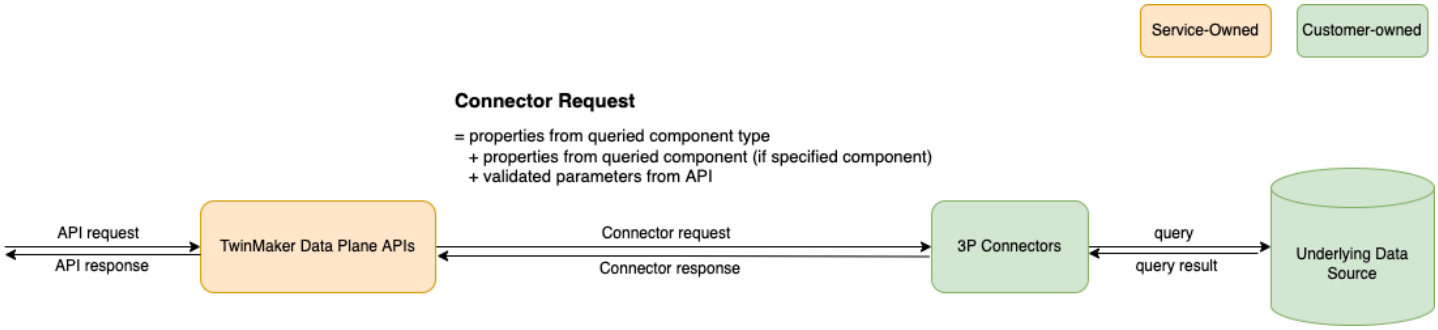
時間序列資料連接器資料流程

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、startTimeEndDateTime、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 蹤連接器的單一要求`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。

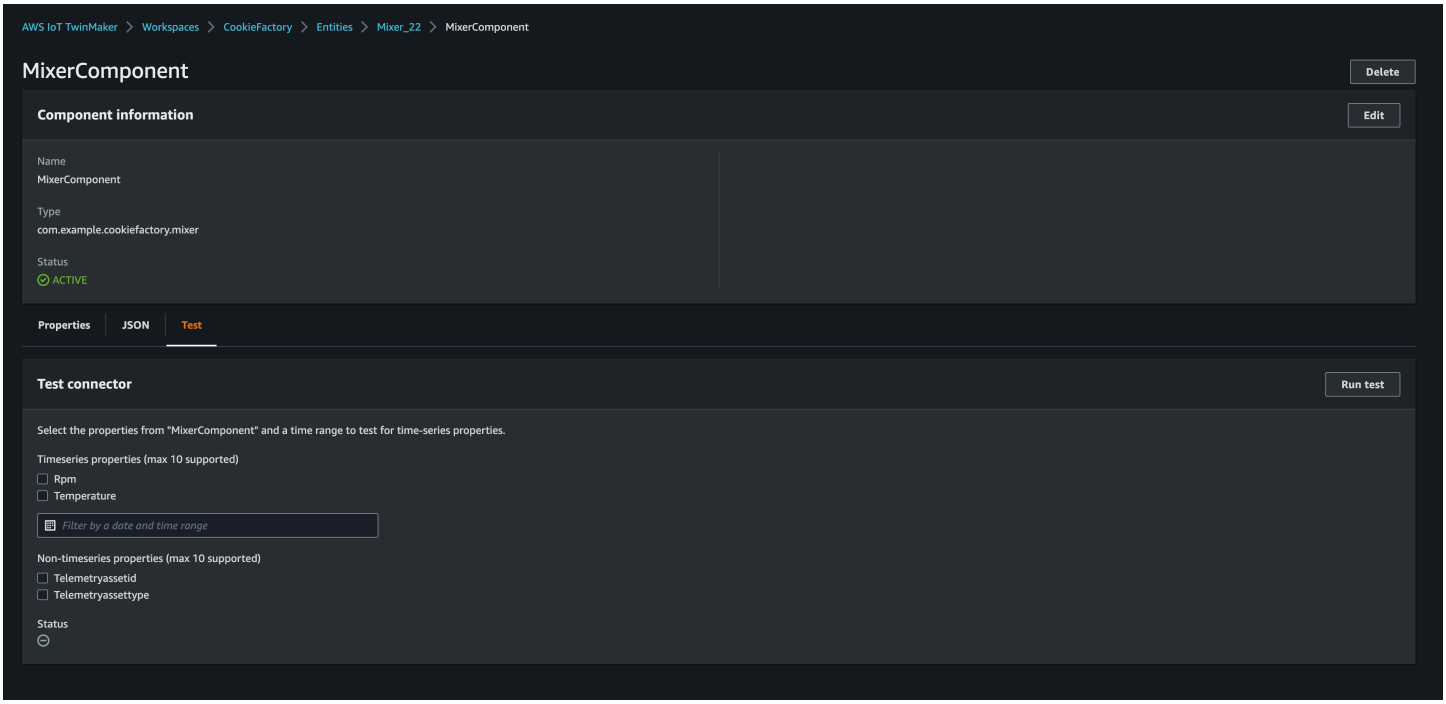
建立 AWS IoT TwinMaker 資源

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。

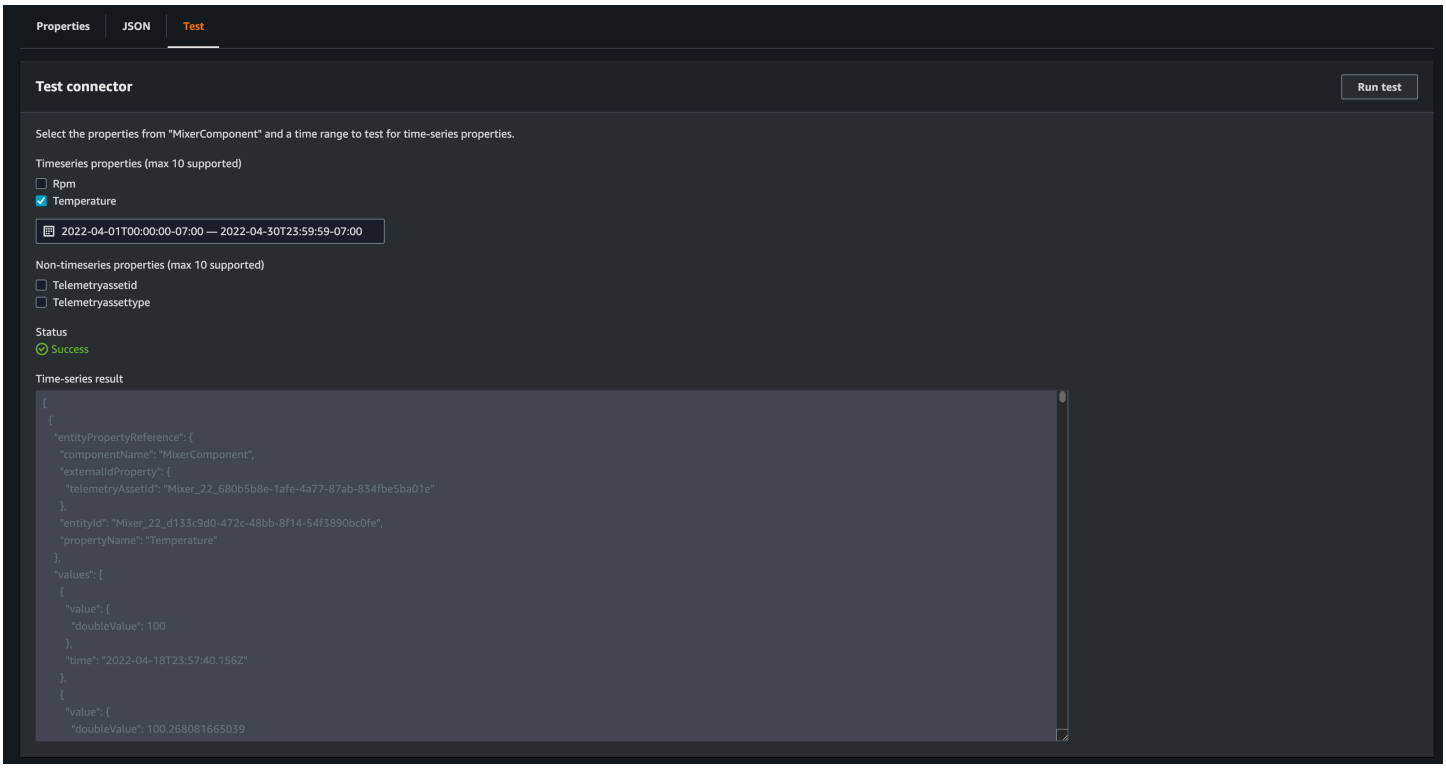
> **Note**
>
> 如果您遵循 GitHub 範例中的設定指示，則會自動提供所有 AWS IoT TwinMaker 資源。您可以檢查範AWS IoT TwinMaker GitHub 例中的元件類型定義。一旦任何元件使用元件類型，就無法更新元件類型的性質定義和函數。

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 GetPropertyValueHistoryAPI 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 GetPropertyValueHistoryAPI 和 non-time-series 屬性使用 GetPropertyValueAPI。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

場景是數位分身的三維視覺化。它們是您編輯數位分身的主要方式。了解如何將警示、時間序列資料、顏色浮水印、標籤和視覺規則新增至您的場景，讓您的數位孿生視覺效果與真實世界的使用案例保持一致。

本節涵蓋下列主題：

- 建立第一個場景之前
- 將資源上傳至 AWS IoT TwinMaker 資源庫
- 建立您的場景
- 將固定攝影機新增至實體
- 場景增強編輯
- 編輯您的場景
- 3D 並排模型格式
- 動態場景

建立第一個場景之前

場景依賴 資源來代表您的數位分身。這些資源是由 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，例如照明和電腦硬體，都會影響場景的效能 AWS IoT TwinMaker 。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。

在將資源匯入 之前最佳化資源 AWS IoT TwinMaker

您可以使用 AWS IoT TwinMaker 與數位分身即時互動。為了獲得最佳的場景體驗，我們建議您最佳化資源，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。複雜的模型幾何和網格可以降低效能。例如，工業 CAD 模型具有高度的細節。我們建議在 AWS IoT TwinMaker 場景中使用多邊形計數之前壓縮這些模型的網格並降低它們的多邊形計數。如果您要為 建立新的 3D 模型 AWS IoT TwinMaker，則應建立細節層級，並在所有模型中加以維護。從不影響使用案例視覺化或解釋的模型中移除詳細資訊。

若要壓縮模型並減少檔案大小，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。如果您的紋理不需要任何透明度，請考慮選擇 PEG 影像格式而非 PNG 格式。您可以使用開放原始碼紋理壓縮工具壓縮紋理檔案，例如 Basis Universal 紋理壓縮。

中效能的最佳實務 AWS IoT TwinMaker

若要使用 獲得最佳效能 AWS IoT TwinMaker，請注意下列限制和最佳實務。

- AWS IoT TwinMaker 場景渲染效能取決於硬體。效能因不同的電腦硬體組態而異。
- 我們建議您在 中所有物件的總多邊形計數低於 100 萬 AWS IoT TwinMaker。
- 我們建議每個場景總共 200 個物件。增加場景中的物件數量超過 200 可以降低場景影格速率。
- 我們建議場景中所有唯一 3D 資產的總大小不超過 100 MB。否則，根據您的瀏覽器和硬體，您可能會遇到載入時間緩慢或效能降低的情況。
- 根據預設，場景具有環境照明。您可以在場景中新增額外的光源，讓特定物件成為焦點，或在物件上投射陰影。我們建議每個場景使用一個光源。視需要使用燈光，並避免在場景中複寫真實世界的燈光。

進一步了解

使用這些資源來進一步了解可用來改善場景效能的最佳化技術。

- 如何將 OBJ 模型轉換和壓縮至 GLTF 以搭配 使用 AWS IoT TwinMaker
- 針對 Web 內容最佳化 3D 模型
- 最佳化場景以獲得更佳的 WebGL 效能

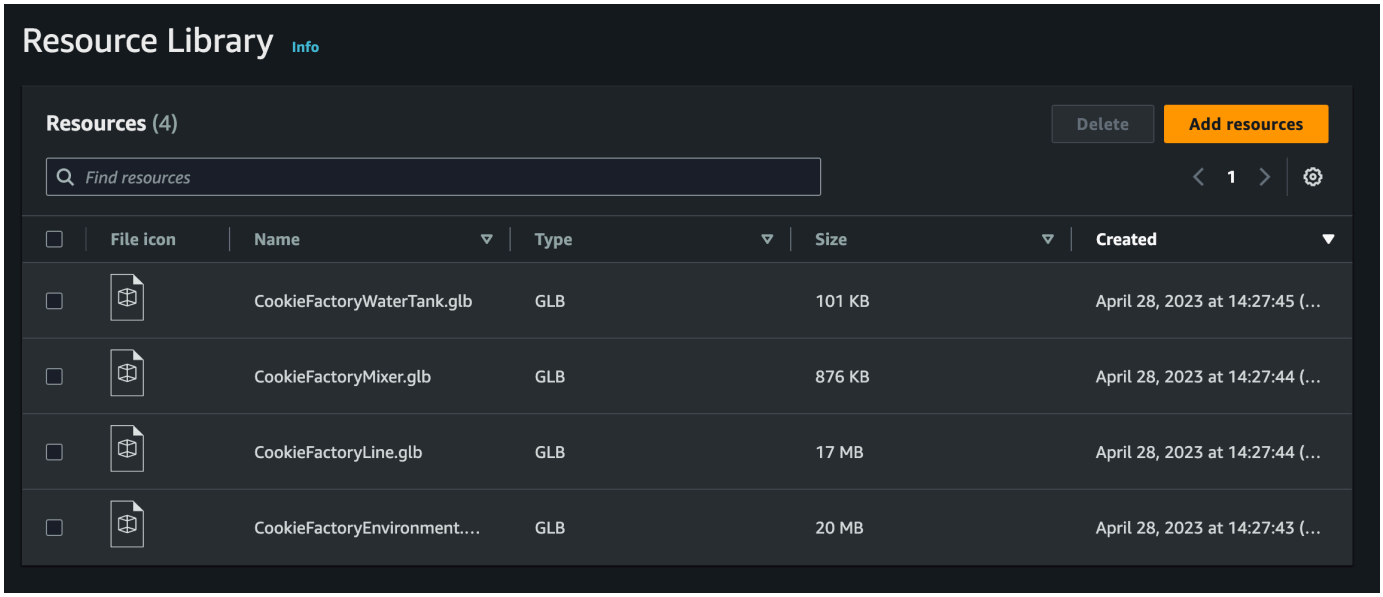
將資源上傳至 AWS IoT TwinMaker 資源庫

您可以使用 資源程式庫來控制和管理要放置在數位分身應用程式場景中的任何資源。若要 AWS IoT TwinMaker 了解資源，請使用 Resource Library 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庫。

1. 在左側導覽選單的工作區下，選取資源庫。
2. 選取新增資源，然後選擇您要上傳的檔案。



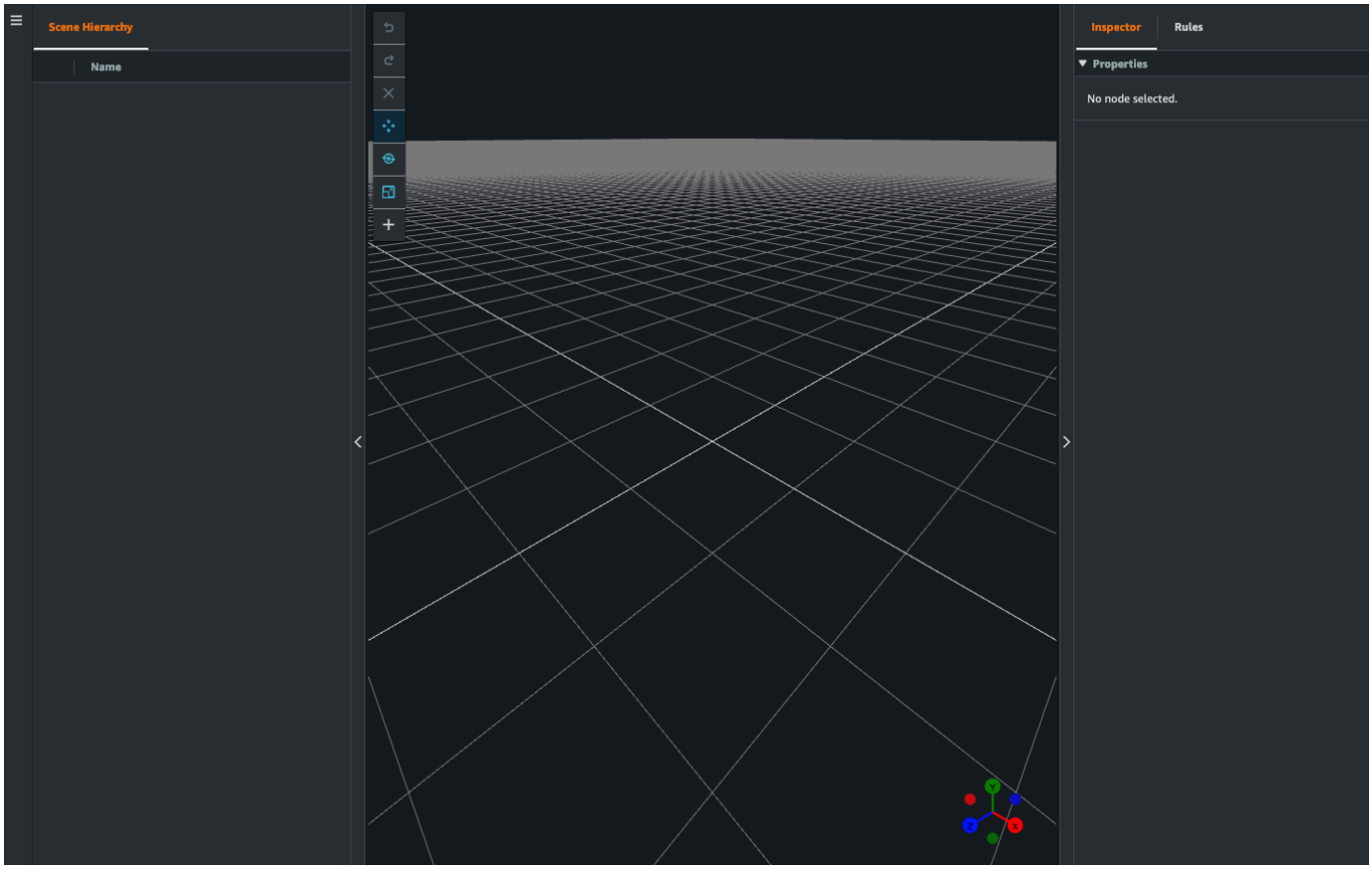
建立您的場景

在本節中，您將設定場景，以便編輯數位分身。您可以匯入上傳至資源庫的 3D 模型，然後新增小工具並將屬性資料繫結至物件，以完成您的數位分身。場景物件可包含整個建築物或空間，或位於其實體位置的個別設備。

> **Note**
>
> 建立場景之前，您必須建立工作區。

使用下列程序在 中建立場景 AWS IoT TwinMaker。

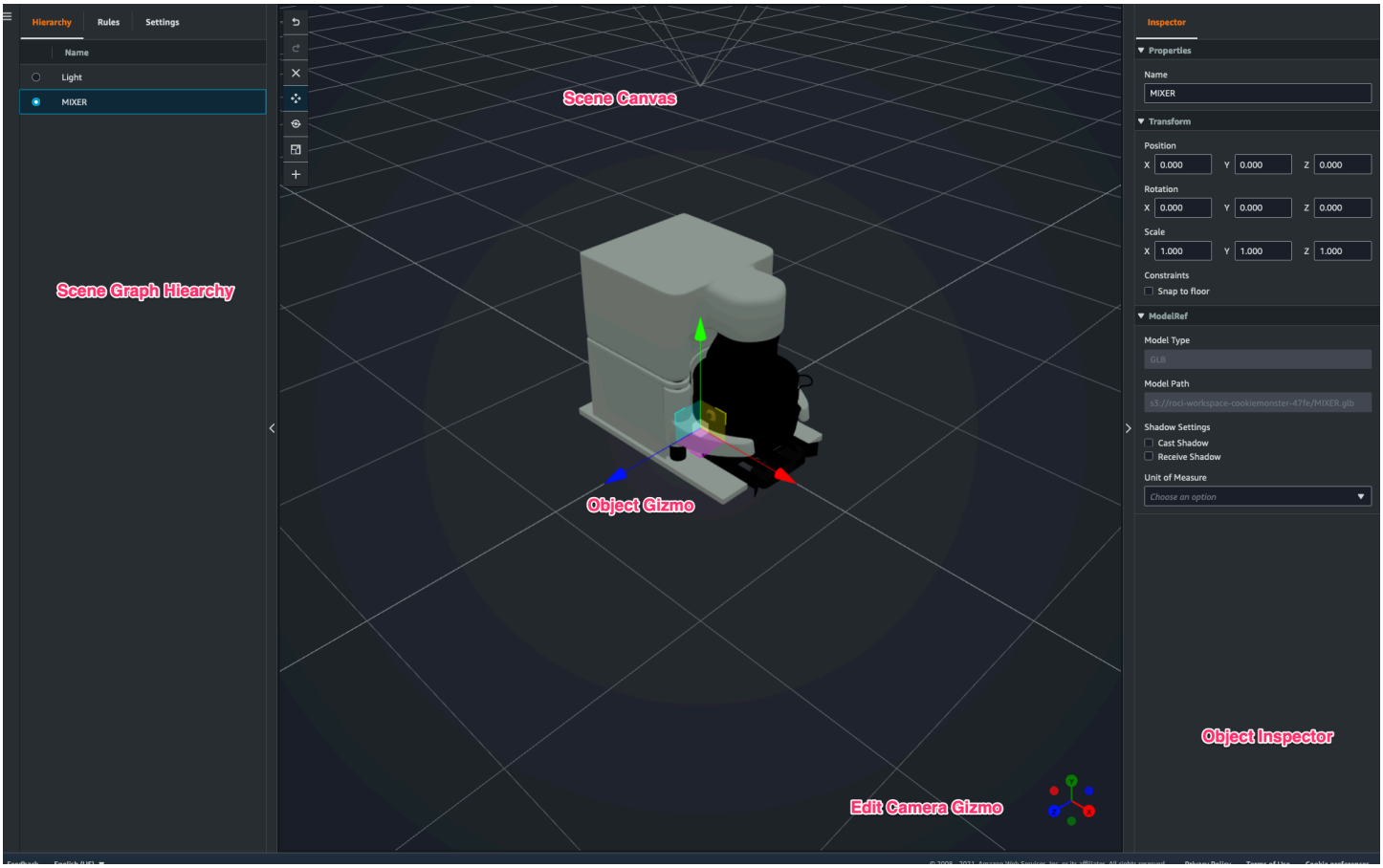
1. 若要開啟場景窗格，請在工作區的左側導覽中，選擇場景。
2. 請選擇 Create scene (建立場景)。新的場景建立窗格隨即開啟。
3. 在場景建立窗格中，輸入新場景的名稱和描述。如果您有標準或分層套件定價計劃，您可以選取場景類型。建議使用動態場景。
4. 當您準備好建立場景時，請選擇建立場景。新的場景會開啟並準備好供您使用。



在 AWS IoT TwinMaker 場景中使用 3D 導覽

AWS IoT TwinMaker 場景有一組導覽控制項，可用來有效率地導覽場景的 3D 空間。若要與場景呈現的 3D 空間和物件互動，請使用下列小工具和功能表選項。

- Inspector：使用 Inspector 視窗來檢視和編輯階層中所選實體或元件的屬性和設定。
- 場景畫布：場景畫布是 3D 空間，您可以在其中定位和定位您想要使用的任何 3D 資源。
- 場景圖階層：您可以使用此面板來查看場景中的所有實體。它會出現在視窗的左側。
- 物件 Gizmo：使用此 Gizmo 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。
- 編輯攝影機 Gizmo：使用編輯攝影機 Gizmo 快速檢視場景檢視攝影機的目前方向，並修改檢視角度。您可以在場景檢視的右下角找到此 Gizmo。
- 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拖曳您要移動的方向。若要旋轉 ，按一下滑鼠左鍵並拖曳以旋轉。若要縮放，請在滑鼠上使用滾輪，或捏合手指，並在筆記型電腦的軌跡鍵盤上將其分開。



階層窗格上的場景按鈕會依按鈕的配置順序列出下列函數：

- 復原：復原您在場景中的上次變更。
- 重做：重做場景中的上次變更。
- 加號 (+)：使用此按鈕來存取下列動作：新增空白節點、新增 3D 模型、新增標籤、新增光線和新增模型著色器。
- 變更導覽方法：存取場景攝影機導覽選項 Orbit 和 Pan。
- Trashcan （ 刪除 ）：使用此按鈕刪除場景中選取的物件。
- 物件操作工具：使用此按鈕來平移、旋轉和擴展選取的物件。

將固定攝影機新增至實體

您可以將固定攝影機檢視連接到 AWS IoT TwinMaker 場景中的實體。這些攝影機提供 3d 模型的固定視角，可讓您快速輕鬆地將場景中的觀點轉移到目標實體。

1. 在 AWS IoT TwinMaker 主控台中導覽至您的場景。

2. 在場景階層功能表中，選取要連接攝影機的實體。
3. 按下 + 按鈕，然後從下拉式清單選項中選取從目前檢視新增攝影機。將具有目前視角的攝影機套用至實體。
4. 在檢測器中，您可以設定攝影機並調整下列設定：
   - 攝影機名稱
   - 攝影機位置和旋轉
   - 相機焦距
   - 縮放層級
   - 近和遠裁切平面
5. 在放置相機之後存取相機。選取您在階層中新增攝影機的實體。尋找實體下列出的攝影機名稱。
6. 從實體選取置放的攝影機後，場景攝影機檢視會貼齊置放攝影機的設定視角。

場景增強編輯

AWS IoT TwinMaker 場景具有一組工具，用於增強和編輯和操作場景中存在的資源。

下列主題說明如何在 AWS IoT TwinMaker 場景中使用增強型編輯功能。

- 場景物件的目標放置
- 子模型選擇
- 在場景階層中編輯實體

場景物件的目標放置

AWS IoT TwinMaker 可讓您將物件精確放置並新增至場景。此增強型編輯功能可讓您更妥善地控制在場景中放置標籤、實體光源和模型的位置。

1. 在 AWS IoT TwinMaker 主控台中導覽至您的場景。
2. 按下 + 按鈕，然後從下拉式選項中選取其中一個選項。這可以是模型、光源、標籤或任何來自 + 選單的內容。

   當您在場景的 3d 空間中移動游標時，應該會在游標 周圍看到目標。
3. 使用目標在您的場景中精確放置元素。

子模型選擇

AWS IoT TwinMaker 可讓您在場景中選取 3d 模型的子模型，並將標準屬性套用至它們，例如標籤、燈光或規則。

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中的子模型。例如，模型可以是過濾系統，如坦克、管道或馬達等系統的個別部分會標記為過濾 3d 模型的子模型。

場景中支援的 3D 檔案格式：GLB 和 GLTF。

1. 在 AWS IoT TwinMaker 主控台中導覽至您的場景。
2. 如果您的場景中沒有模型，請務必從 + 功能表中選取 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方顯示任何子模型。

> **Note**
>
> 如果您沒有看到列出的任何子模型，則可能模型未設定為有任何子模型。

4. 若要切換子模型的可見性，請按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。使用檢測器功能表更新或變更子模型資料。
6. 若要將標籤、燈光、規則或其他屬性新增至子模型，請按 + ，同時在階層中選取子模型。

在場景階層中編輯實體

AWS IoT TwinMaker 場景可讓您直接編輯階層資料表中實體的屬性。下列程序顯示您可以透過階層功能表對實體執行的動作。

1. 在 AWS IoT TwinMaker 主控台中導覽至您的場景。
2. 開啟場景階層，然後選取您要操作之實體的子元素。
3. 選取元素後，按下 + 按鈕，然後從下拉式清單中選取其中一個選項：
   - 新增空白節點
   - 新增 3D 模型
   - 新增光線
   - 從目前檢視新增攝影機

- 新增標籤
- 新增模型著色器
- 新增動作指標

4. 從下拉式清單中選取其中一個選項後，系統會將選取項目套用到場景，做為步驟 2 中所選元素的子項。
5. 您可以透過選取子元素並在階層中拖曳至新的父系，來重新排序子元素和父系元素。

將註釋新增至實體

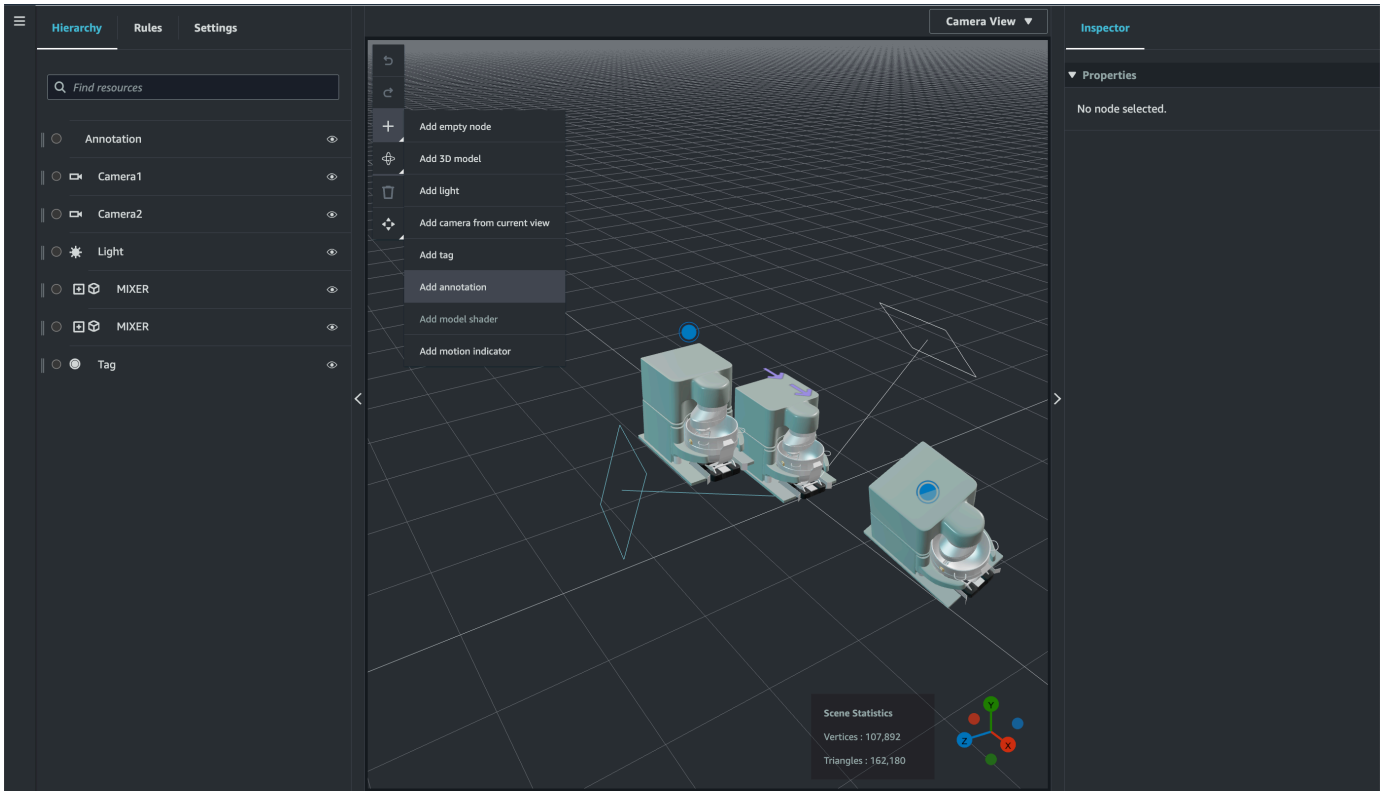
AWS IoT TwinMaker 場景編寫器可讓您註釋場景階層中的任何元素。註釋是以 Markdown 撰寫。

如需在 Markdown 中寫入的詳細資訊，請參閱 Markdown 語法、基本語法上的官方文件。

> **Note**
>
> AWS IoT TwinMaker 僅限 註釋和浮水印 Markdown 語法，而非 HTML。

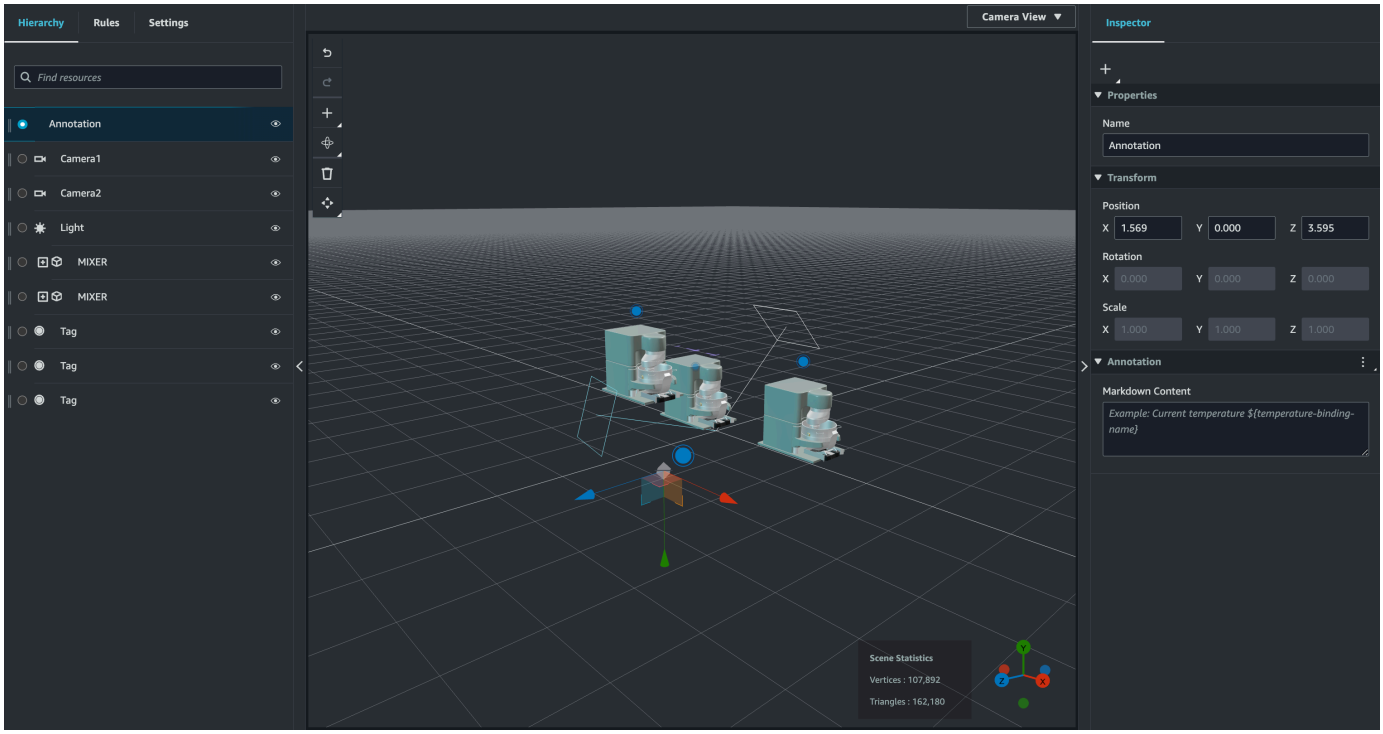
將註釋新增至實體

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釋的元素。如果未選取階層中的元素，則可以將註釋新增至根目錄。
3. 按下加號 + 按鈕，然後選擇新增註釋選項。

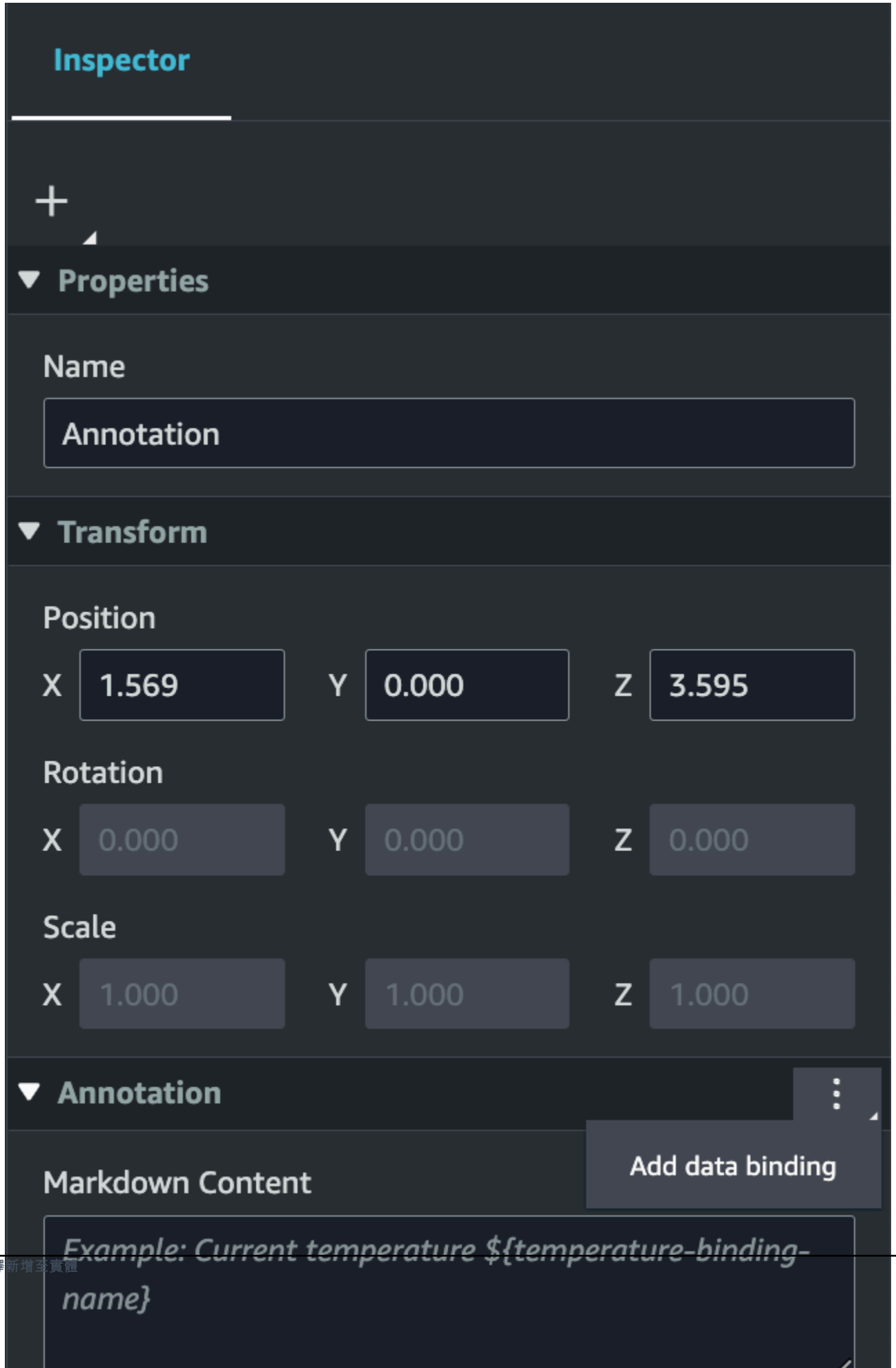


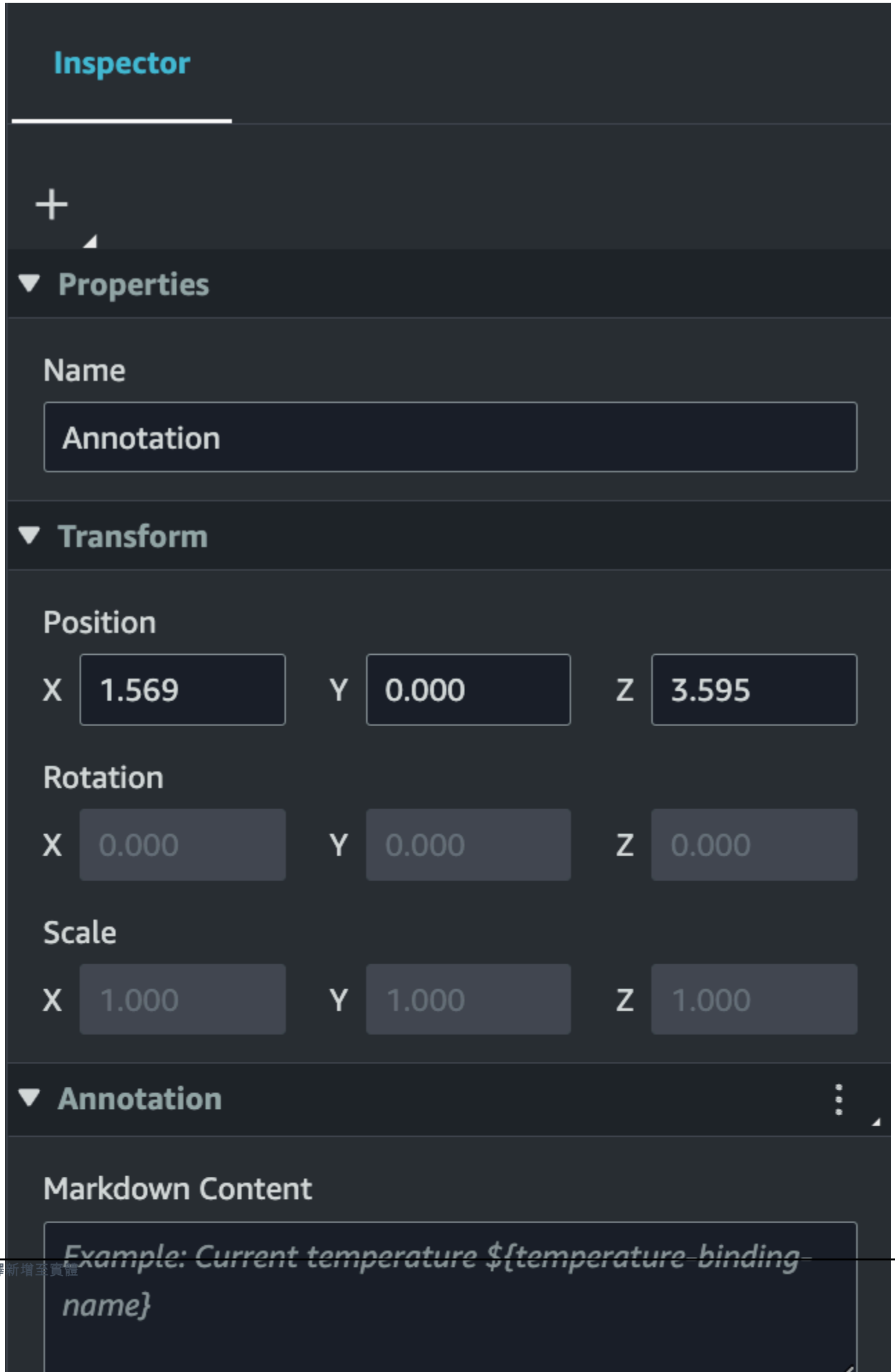
4. 在左側的檢查器視窗中，向下捲動至註釋區段。使用 Markdown 語法，撰寫您希望註釋顯示的文字。

   如需在 Markdown 中寫入的詳細資訊，請參閱 Markdown 語法、基本語法上的官方文件。



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，然後選取您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。

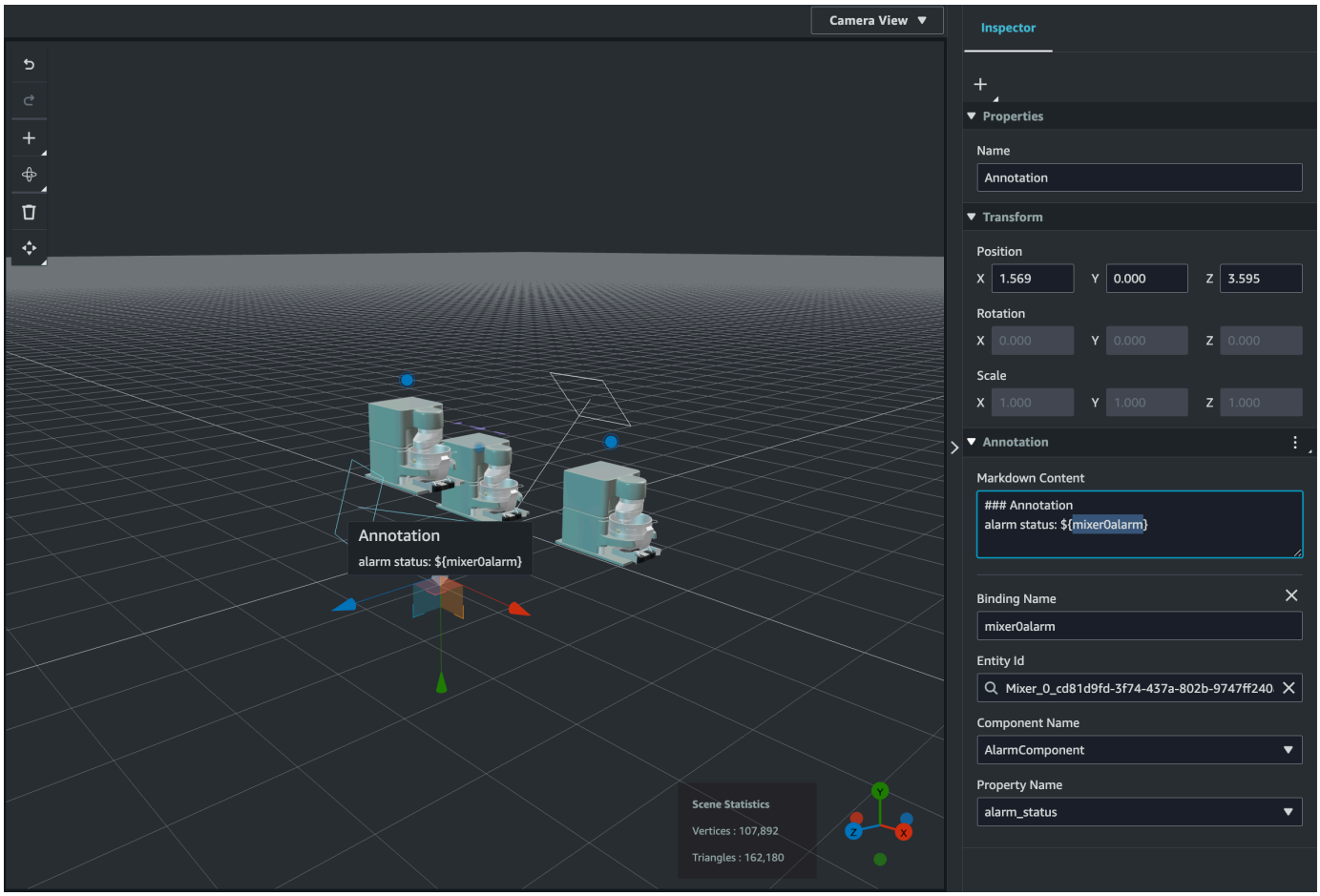

Inspector
Properties
Name
Annotation
Transform
Position
X 1.569
Y 0.000
Z 3.595
Rotation
X 0.000
Y 0.000
Z 0.000
Scale
X 1.000
Y 1.000
Z 1.000
Annotation
Add data binding
Markdown Content
Example: Current temperature ${temperature-binding-name}


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

6. 繫結名稱用於代表註釋的變數。

   輸入繫結名稱，透過 的 AWS IoT TwinMaker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，此浮水印會在具有語法 的註釋`mixer0alarm`中顯示 的值`${mixer0alarm}`。



將浮水印新增至標籤

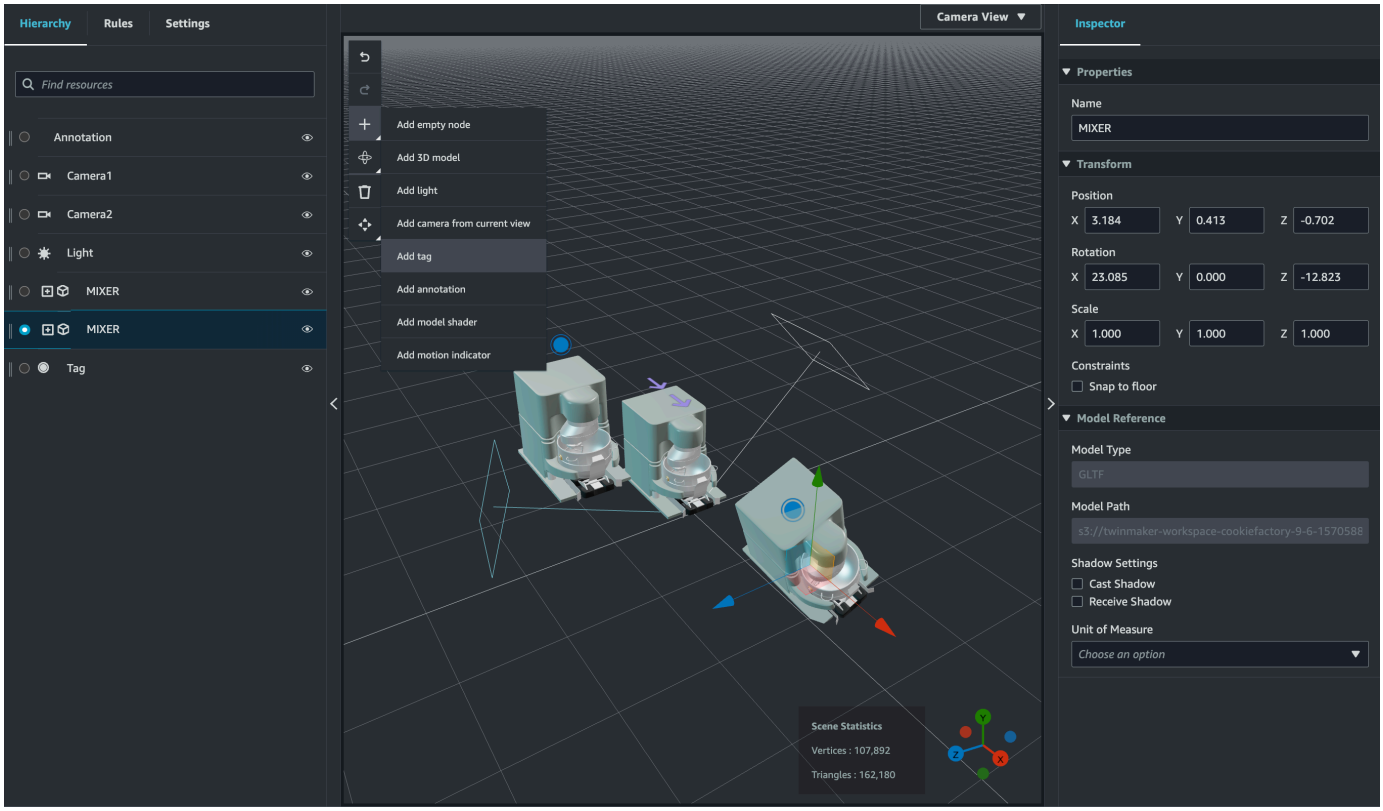
您可以為 AWS IoT TwinMaker 場景建立浮水印。場景浮水印與標籤相關聯，可用於浮水印與場景實體相關聯的關鍵資料。浮水印是在 Markdown 中編寫和轉譯。

如需在 Markdown 中寫入的詳細資訊，請參閱 Markdown 語法、基本語法上的官方文件。

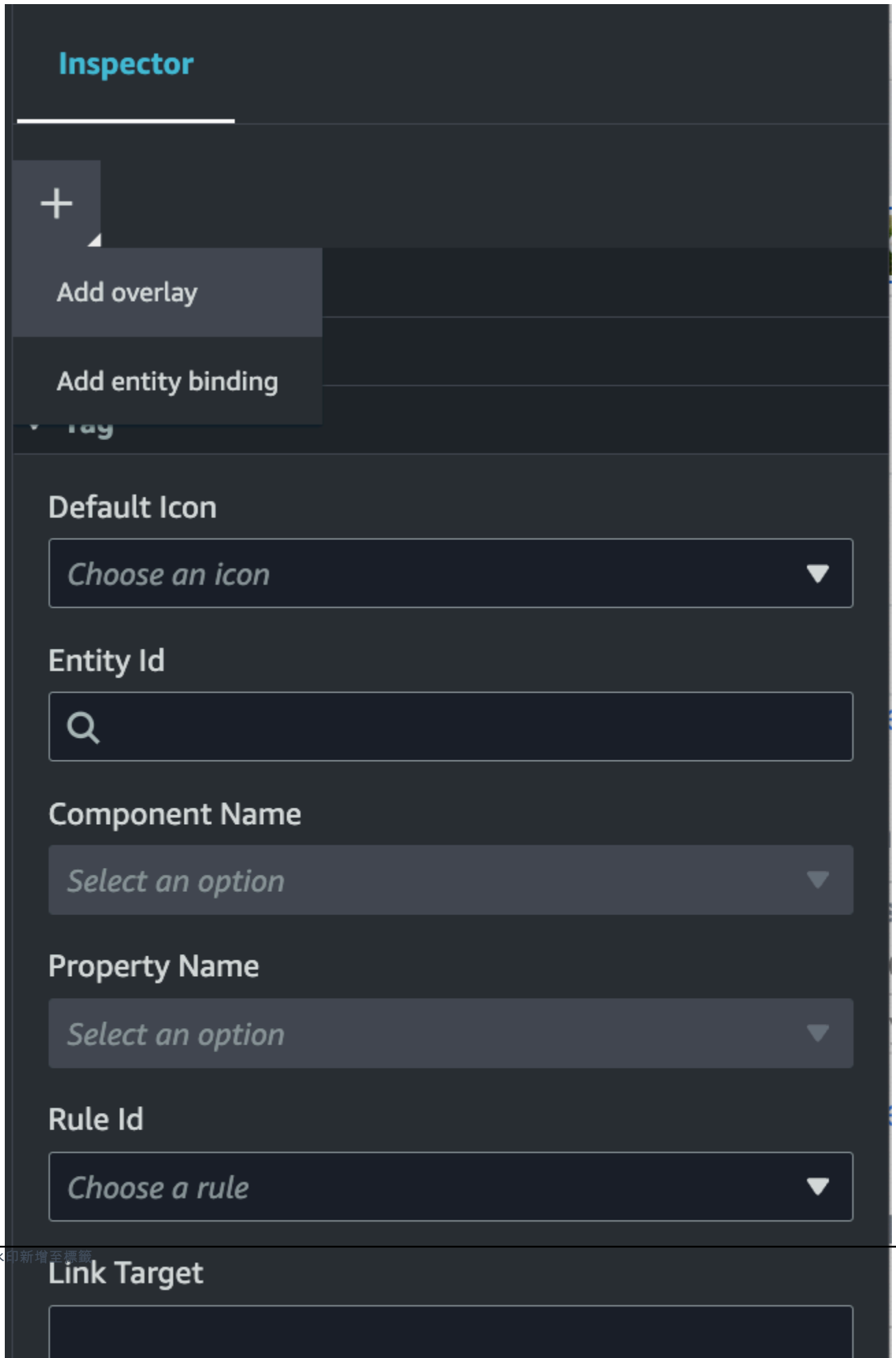
> **Note**
>
> 根據預設，只有在選取與其相關聯的標籤時，才會在場景中顯示浮水印。您可以在場景設定中切換，以便一次顯示所有浮水印。

1. 在 AWS IoT TwinMaker 主控台中導覽至您的場景。
2. AWS IoT TwinMaker 浮水印與標籤場景相關聯，您可以更新現有的標籤或新增標籤。

   按下加號 + 按鈕，然後選擇新增標籤選項。



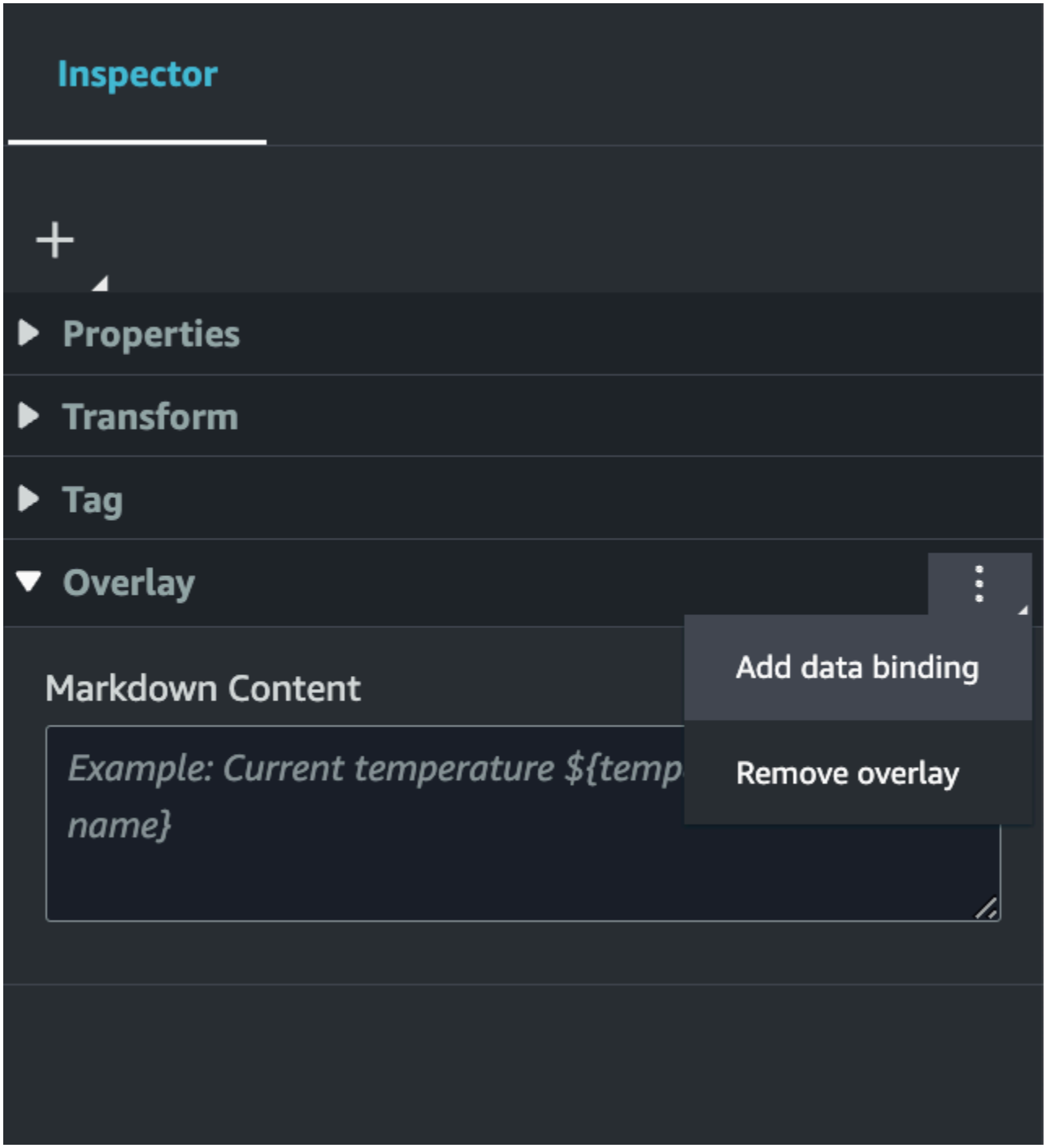
3. 在右側的檢查工具面板中，選取 + （加號）按鈕，然後選取新增浮水印。


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，撰寫您希望浮水印顯示的文字。

   如需在 Markdown 中寫入的詳細資訊，請參閱 Markdown 語法、基本語法上的官方文件。

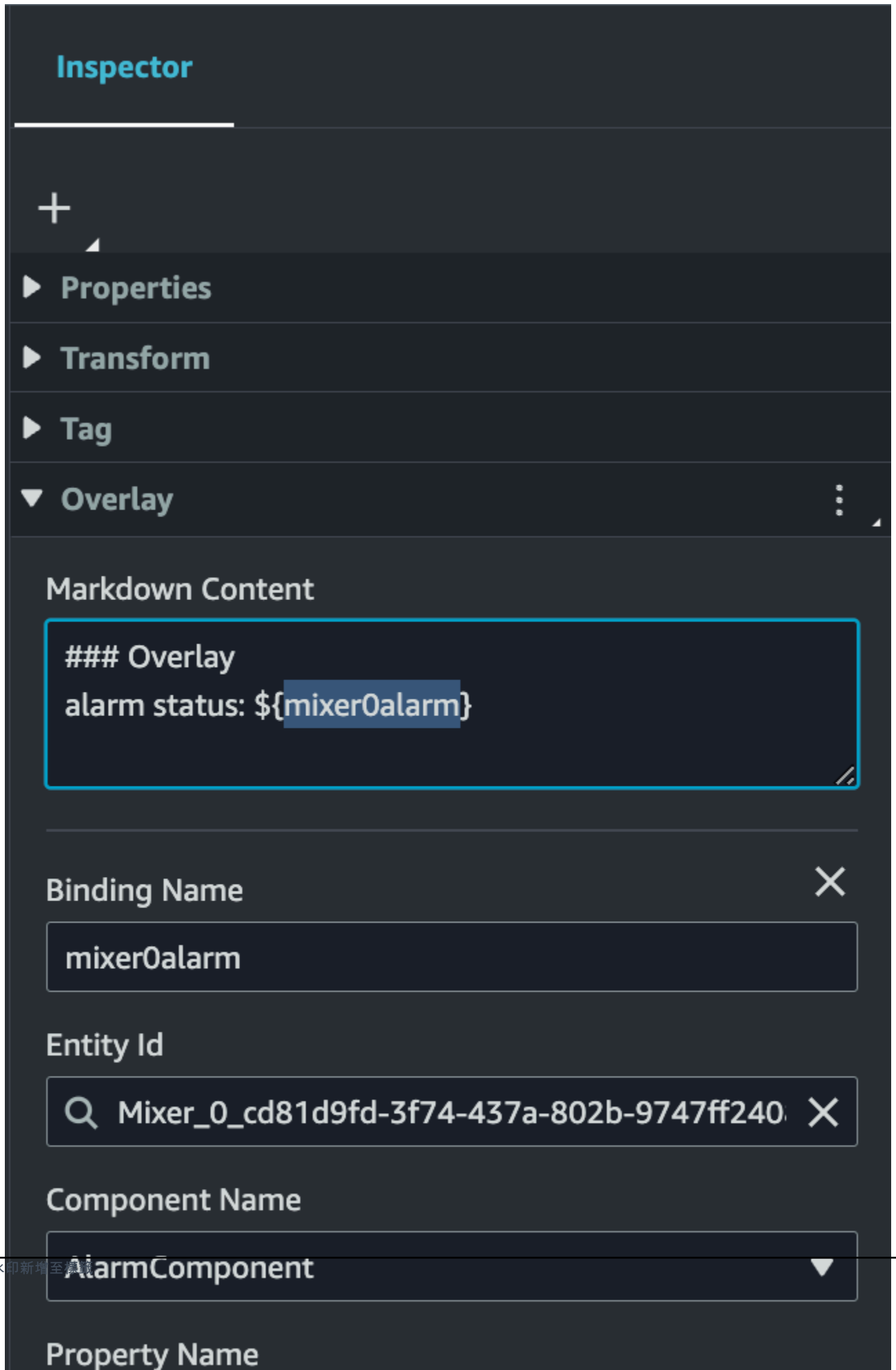
5. 若要將 AWS IoT TwinMaker 場景資料繫結至浮水印，請選取新增資料繫結。



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

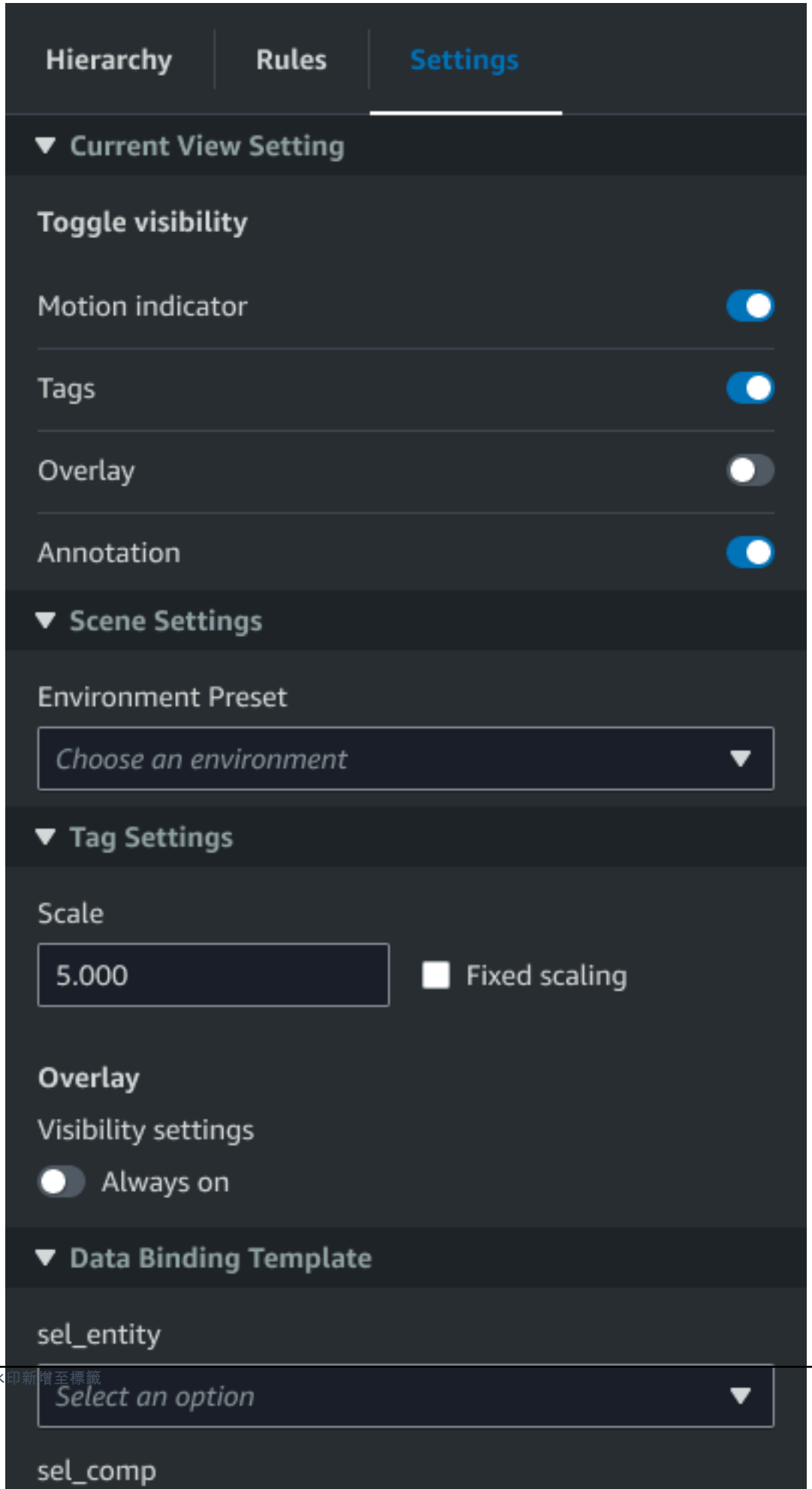
6. 您可以透過 的 AWS IoT TwinMaker變數語法，在浮水印中顯示實體時間序列資料的最新歷史值：${*variable-name*}。

   例如，此浮水印會在具有語法 的浮水印mixer0alarm中顯示 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

7. 若要啟用浮水印可見性，請開啟左上角的設定索引標籤，並確認浮水印的切換已開啟，以便一次顯示所有浮水印。

> **Note**
>
> 根據預設，只有在選取與其相關聯的標籤時，才會在場景中顯示浮水印。


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

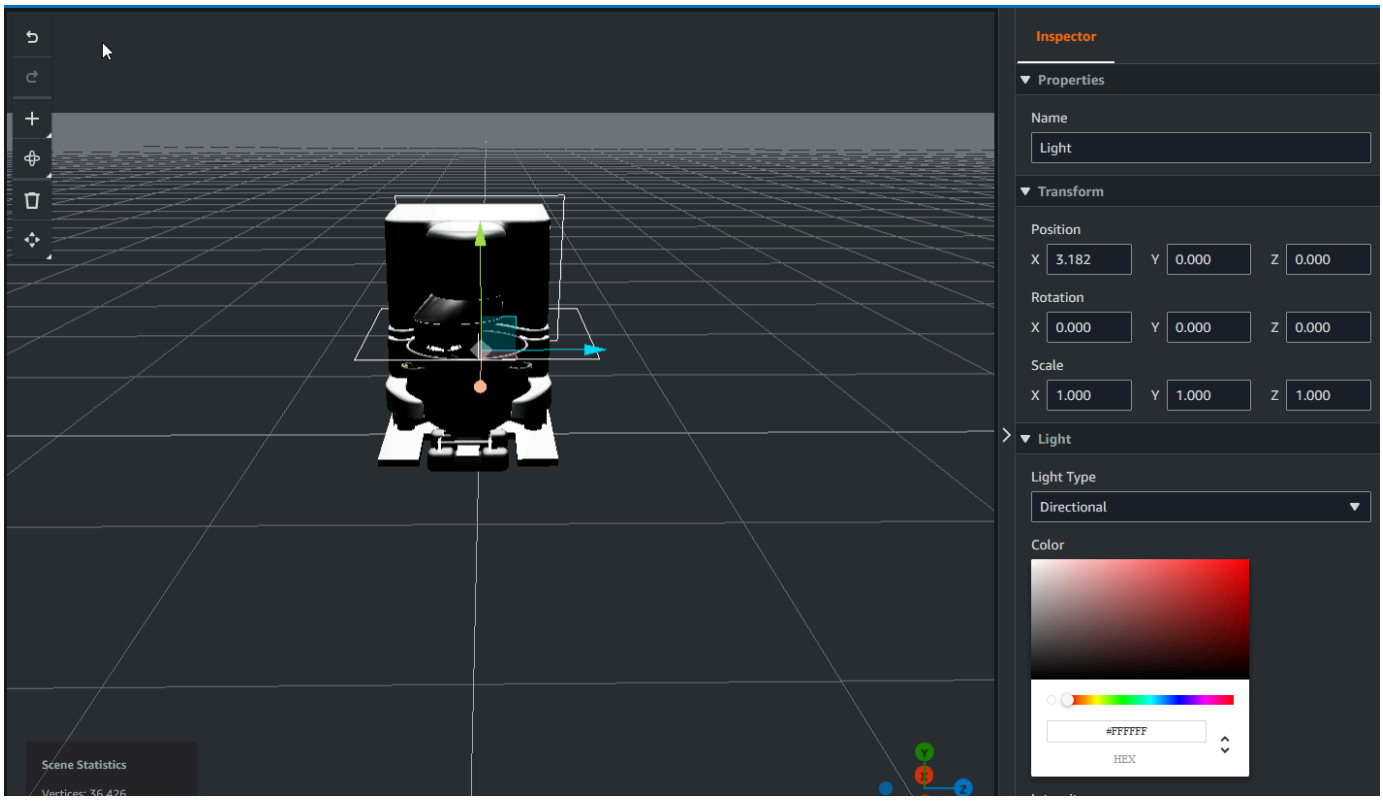
建立場景之後，您可以將實體、元件和設定擴增小工具新增至場景。使用實體元件和小工具來建立數位分身的模型，並提供符合您使用案例的功能。

將模型新增至您的場景

若要將模型新增至場景，請使用下列程序。

> **Note**
>
> 若要在場景中新增模型，您必須先將模型 AWS IoT TwinMaker 上傳至資源庫。如需詳細資訊，請參閱將資源上傳至 AWS IoT TwinMaker 資源庫。

1. 在場景編寫器頁面上，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。場景編譯器隨即開啟。
3. 選用：選擇加號 (+)，然後選擇新增光源。
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照明。若要避免影格率遺失，請考慮限制場景中放置的其他光線數量。

將模型著色器擴增的 UI 小工具新增至場景

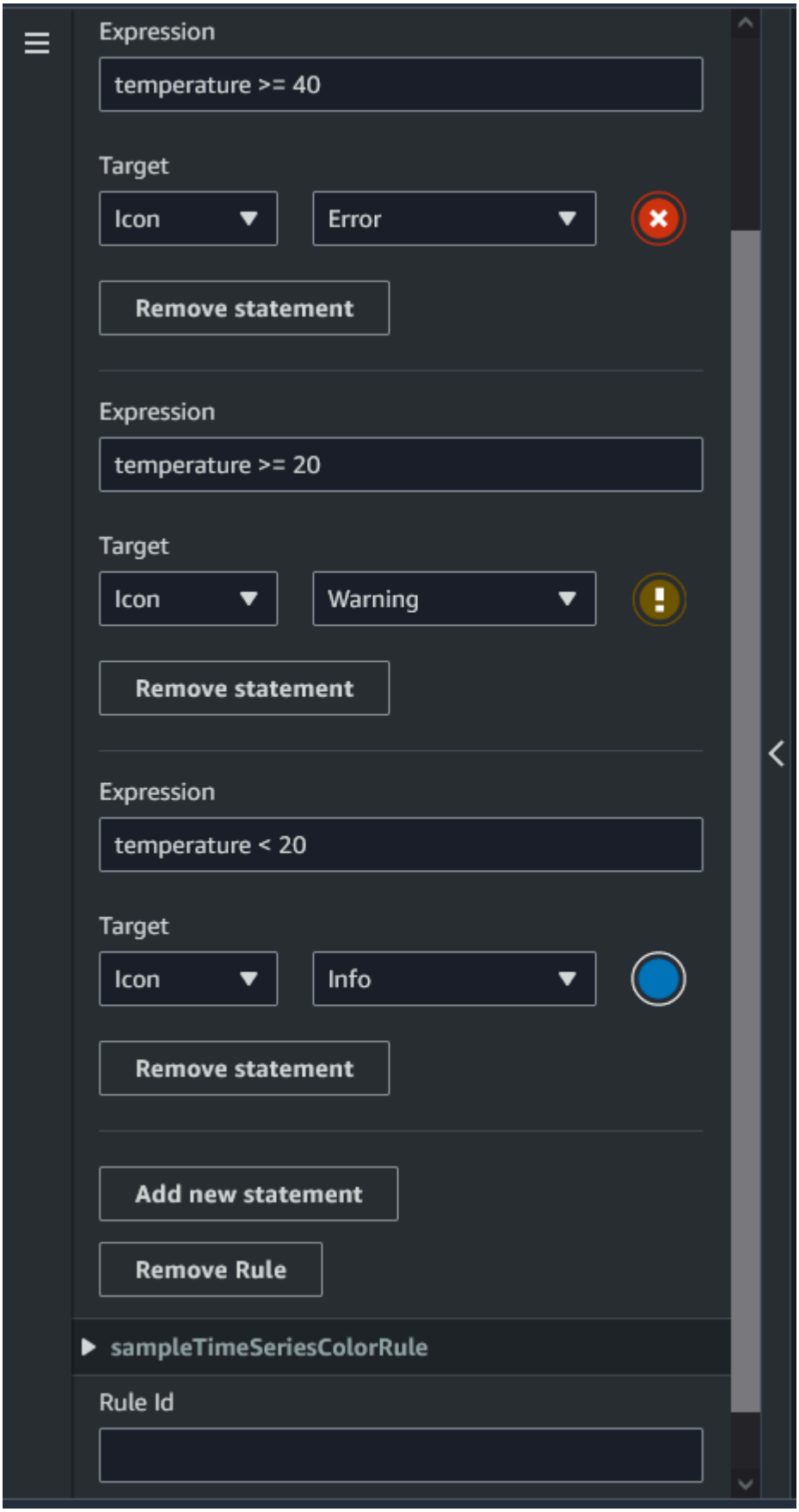
模型著色器小工具可以在您定義的條件下變更物件的顏色。例如，您可以建立顏色小工具，根據混音器的溫度資料變更場景中 Cookie 混音器的顏色。

使用下列程序將模型著色器小工具新增至選取的物件。

1. 在您要新增小工具的階層中選取物件。按下 + 按鈕，然後選擇模型著色器。
2. 若要新增新的視覺化規則群組，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. 選取您要繫結模型著色器的 entityID、 ComponentName 和 PropertyName。

為您的場景建立視覺化規則

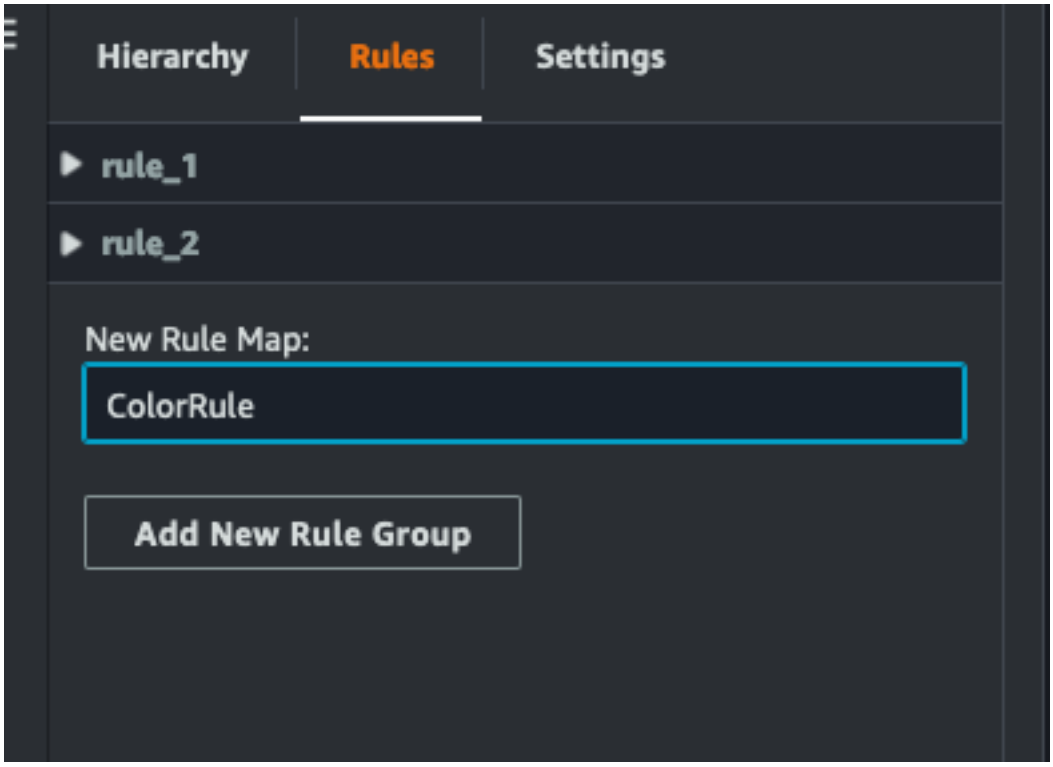
您可以使用視覺化規則映射來指定資料驅動條件，以變更擴增 UI 小工具的視覺化外觀，例如標籤或模型著色器。提供範例規則，但您也可以建立自己的規則。下列範例顯示視覺化規則。


Expression
temperature >= 40
Target
Icon
Error
Remove statement
Expression
temperature >= 20
Target
Icon
Warning
Remove statement
Expression
temperature < 20
Target
Icon
Info
Remove statement
Add new statement
Remove Rule
sampleTimeSeriesColorRule
Rule Id

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

下列程序說明如何為網格著色增強型 UI 層新增視覺規則群組。

1. 在主控台的規則索引標籤下，在文字欄位中輸入名稱，例如 ColorRule，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。規則表達式請使用下列語法：小於 `<`、大於 `>`、小於或等於 `<=`、大於或等於 `>=`，以及等於 `==`。（如需詳細資訊，請參閱 Apache Commons JEXL 語法。）
3. 將目標設為顏色。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。）

為您的場景建立標籤

標籤是新增到場景特定`x,y,z`座標位置的註釋。標籤使用實體屬性將場景部分連接到知識圖表。您可以使用標籤來設定場景中項目的行為或視覺效果，例如警示。

> **Note**
>
> 若要將功能新增至標籤，您可以將視覺化規則套用至標籤。

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，然後選擇新增標籤。
2. 為標籤命名。然後，若要套用視覺化規則，請選取視覺化群組 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 並排模型格式

在場景中使用 3D 圖磚

如果您在 中載入 3D 場景時遇到長時間等待， AWS IoT TwinMaker 或在導覽複雜的 3D 模型時轉譯效能不佳，則建議您將模型轉換為 3D 圖磚。本節說明 3D 圖磚格式和可用的第三方工具。繼續閱讀，以決定 3D 圖磚是否適合您的使用案例，並協助您開始使用。

複雜的模型使用案例

如果模型符合下列條件， 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航延遲：

- 大型：其檔案大小大於 100MB。
- 密集：由數百或數千個不同的網格組成。
- 複雜：網格幾何有數百萬個三角形可形成複雜形狀。

3D 圖磚格式

3D Tiles 格式是串流模型幾何和改善 3D 轉譯效能的解決方案。它可即時載入 AWS IoT TwinMaker 場景中的 3D 模型，並根據相機檢視中可見的內容載入模型的區塊，以最佳化 3D 互動。

3D Tiles 格式是由 Cesium 建立。Cesium 具有受管服務，可將 3D 模型轉換為稱為 Cesium Ion 的 3D 圖磚。這是目前建立 3D 圖磚的最佳解決方案，我們建議您針對支援格式的複雜模型使用。您可以在 Cesium 的定價頁面上註冊 Cesium，並根據業務需求選擇適當的訂閱計劃。

若要準備可新增至 AWS IoT TwinMaker 場景的 3D 並排模型，請遵循 Cesium Ion 所記錄的指示：

- 將模型匯入 Cesium Ion

將 Cesium 3D 圖磚上傳至 AWS

模型轉換為 3D 圖磚後，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. 建立和下載 3D Tiles 模型封存。
2. 將封存解壓縮至資料夾。
3. 將整個 3D Tiles 資料夾上傳至與 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 圖磚模型已成功上傳，您會在 AWS IoT TwinMaker 資源程式庫中看到類型為 的 Amazon S3 資料夾路徑`Tiles3D`。

> **Note**
>
> AWS IoT TwinMaker 資源庫不支援直接上傳 3D 圖磚模型。

在 中使用 3D 圖磚 AWS IoT TwinMaker

AWS IoT TwinMaker 知道上傳到工作區 S3 儲存貯體的任何 3D 圖磚模型。模型必須在同一 Amazon S3 目錄中具有可用的 `tileset.json`和所有相依檔案 (.gltf、.b3dm、.i3dm、.cmpt、.pnts)。Amazon S3 目錄路徑會出現在 資源程式庫中，類型為 `Tiles3D`。

若要將 3D 圖磚模型新增至場景，請依照下列步驟進行：

1. 在場景編寫器頁面上，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，選擇類型為 的 3D 並排模型路徑`Tiles3D`，然後選擇新增。
3. 按一下畫布，將模型放置在場景中。

3D 圖磚差異

3D 圖磚目前不支援幾何和語意中繼資料，這表示原始模型的網格階層不適用於子模型選取功能。您仍可將 Widget 新增至 3D Tiles 模型，但無法使用針對子模型微調的功能：模型著色器、分離的 3D 轉換或子模型網格的實體繫結。

建議針對做為場景背景內容的大型資產使用 3D 圖磚轉換。如果您希望進一步細分和註釋子模型，則應將其擷取為單獨的 glTF/glb 資產，並直接新增到場景。這可以透過 Blender 等免費和常見的 3D 工具來完成。

範例使用案例：

- 您有 1GB 的工廠模型，其中包含詳細的機器房間和地板、電箱和管道。當相關聯的屬性資料超過閾值時，電氣箱和管道需要發光紅色。
- 您可以在模型中隔離方塊和管道網格，並使用 Blender 匯出到單獨的 glTF。
- 您可以將沒有電氣和管道元素的工廠轉換為 3D 圖磚模型，並將其上傳至 S3。
- 您可以將 3D 並排模型和 glTF 模型新增至原始伺服器的 AWS IoT TwinMaker 場景 (0，0，0)。
- 您可以將模型著色器元件新增至 glTF 的電氣盒和管道子模型，根據屬性規則使網格變成紅色。

動態場景

AWS IoT TwinMaker 場景透過將場景節點和設定存放在實體元件中，釋放知識圖表的強大功能。使用 AWS IoT TwinMaker 主控台建立動態場景，以更輕鬆地管理、建置和轉譯 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表查詢「動態」轉譯。
- 如果您在 Grafana 或自訂應用程式中使用唯讀場景檢視器，您可以在 30 秒間隔內取得場景的更新。

靜態與動態場景

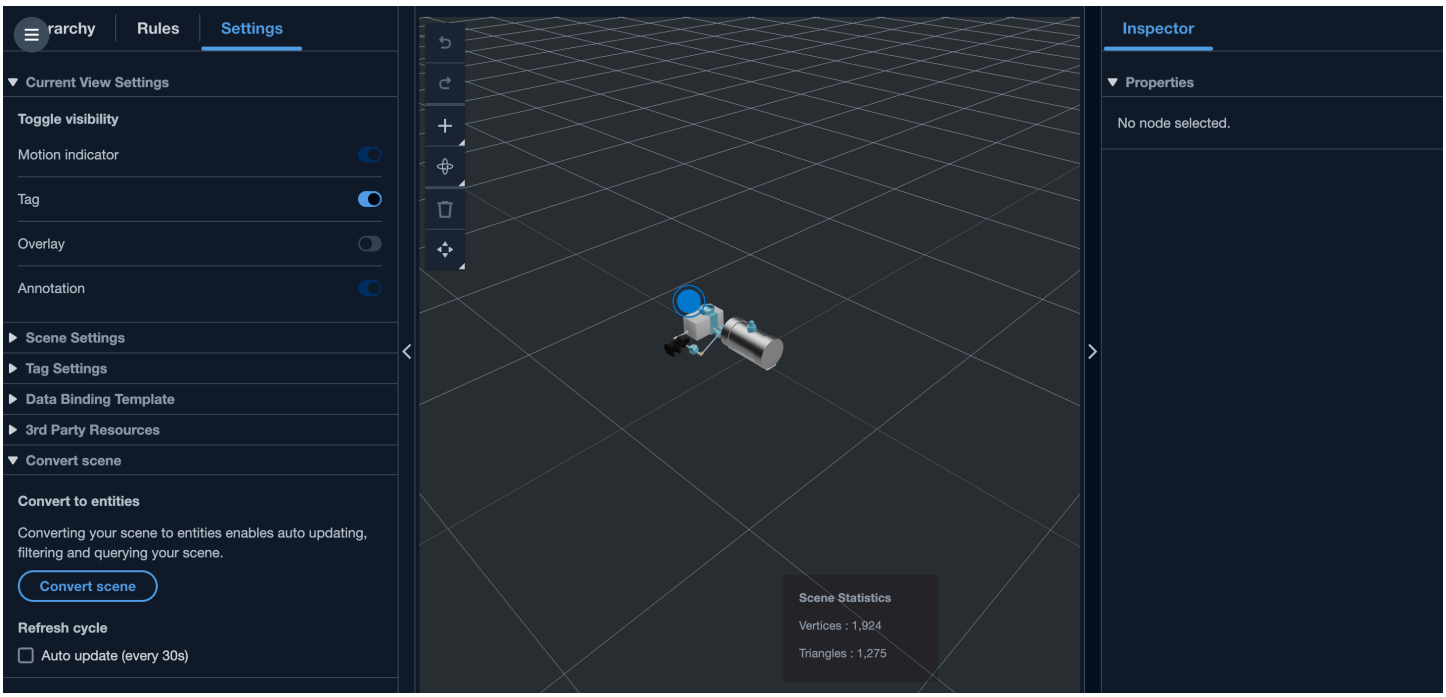
靜態場景由存放在 S3 中的場景 JSON 檔案組成，其中包含所有場景節點和設定的詳細資訊。場景的任何變更都必須對 JSON 文件進行，並儲存至 S3。如果您有基本定價計劃，靜態場景是唯一的選項。

動態場景是由場景 JSON 檔案組成，該檔案具有場景的全域設定，而所有其他場景節點和節點設定則以實體元件的形式存放在知識圖表中。只有標準和分層套件定價計劃才支援動態場景。如需如何升級定價計劃的資訊切換 AWS IoT TwinMaker 定價模式，請參閱)。

您可以依照下列步驟，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，按一下設定索引標籤。
- 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**
>
> 從靜態轉換為動態場景是無法復原的。



場景元件類型和實體

為了建立場景特定的實體元件，支援下列 1P 元件類型：

- com.amazon.iottwinmaker.3d.component.camera 一種存放攝影機小工具設定的元件類型。
- com.amazon.iottwinmaker.3d.component.dataoverlay 一種元件類型，用於存放註釋或標籤小工具的浮水印設定。
- com.amazon.iottwinmaker.3d.component.light 一種元件類型，可存放淺色小工具的設定。

- com.amazon.iottwinmaker.3d.component.modelref 一種元件類型，用於存放場景中所用 3D 模型的設定和 S3 位置。
- com.amazon.iottwinmaker.3d.component.modelshader 在 3D 模型上存放模型著色器設定的元件類型。
- com.amazon.iottwinmaker.3d.component.motionindicator 存放動作指標小工具設定的元件類型。
- com.amazon.iottwinmaker.3d.component.submodelref 一種元件類型，用於存放 3D 模型子模型的設定。
- com.amazon.iottwinmaker.3d.component.tag 存放標籤小工具設定的元件類型。
- com.amazon.iottwinmaker.3d.node 一種元件類型，可存放場景節點的基本設定，例如其 3D 轉換、名稱和一般屬性。

動態場景概念

動態場景實體存放在標記為 的全域實體下$SCENES。每個場景都由根實體和符合場景節點階層的子實體階層組成。根下的每個場景節點都有一個 com.amazon.iottwinmaker.3d.node 元件，以及節點類型的元件 (3D 模型、小工具等) 。

> **Warning**
>
> 請勿手動刪除任何場景實體，否則場景可能處於中斷狀態。如果您想要部分或完全刪除場景，請使用場景編寫器頁面來新增和刪除場景節點，並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`的數字孿生，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。

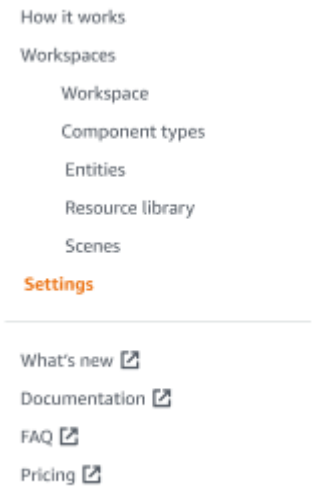
切換 AWS IoT TwinMaker 定價模式

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。

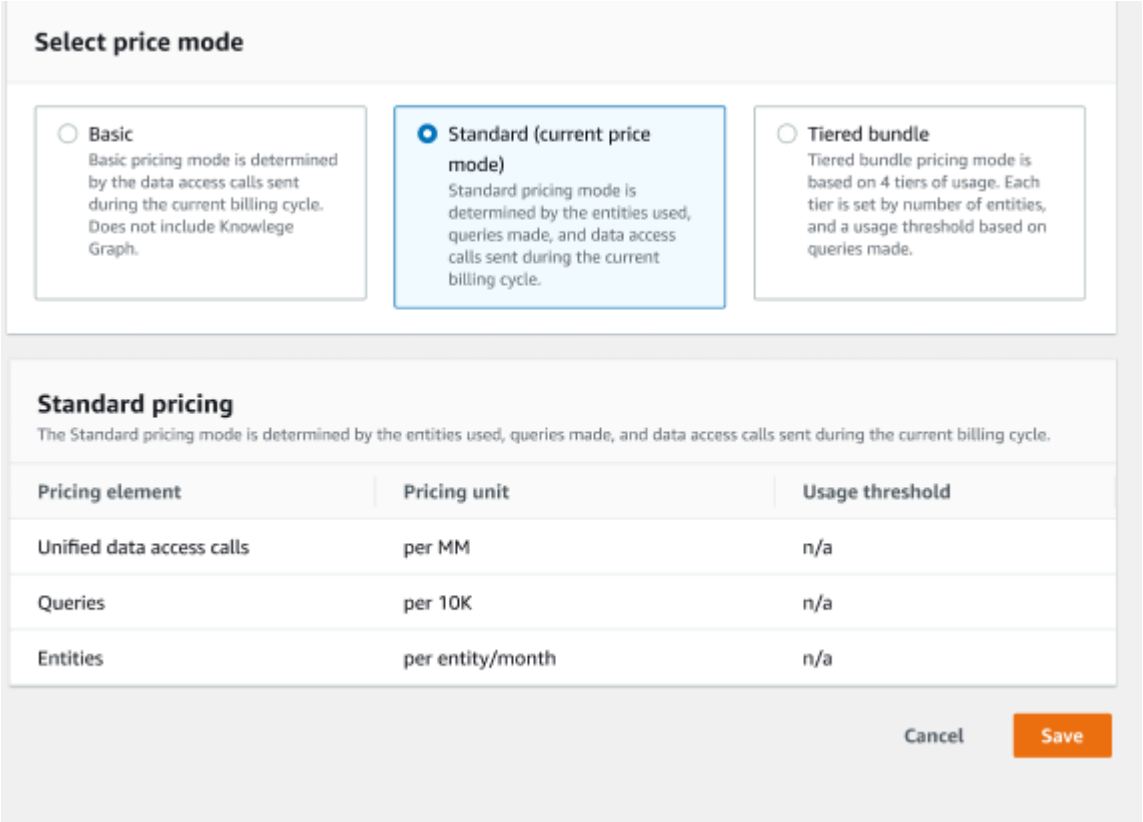
您可以隨時從以使用量為基礎的定價模式切換為階層式的定價模式，但變更會在下一個帳單週期開始生效。從以使用量為基準的訂價模式切換至以階層為基準的訂價模式後，您就無法針對接下來的三個使用週期切換回以使用量為基準的訂價模式。如果您從基本切換到標準，則變更會立即生效。如需詳細資訊和費用資訊，請參閱AWS IoT TwinMaker 定價

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。



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。



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。

> **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 知識圖表

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。您可以對實體、元件和元件類型執行查詢，以產生視覺化圖形，顯示資源 AWS IoT TwinMaker 之間的關係。

下列主題說明如何使用和整合知識圖表。

主題

- AWS IoT TwinMaker 知識圖表核心概念
- 如何執行 AWS IoT TwinMaker 知識圖表查詢
- 知識圖表場景整合
- 如何搭配 Grafana 使用 AWS IoT TwinMaker 知識圖表
- AWS IoT TwinMaker 知識圖表其他資源

AWS IoT TwinMaker 知識圖表核心概念

本主題涵蓋知識圖表功能的重要概念和詞彙。

知識圖表的運作方式：

知識圖表會建立實體及其元件與現有 CreateEntity 或 UpdateEntity APIs 之間的關係。關係只是在實體元件上定義的特殊資料類型 RELATIONSHIP 屬性。 AWS IoT TwinMaker 知識圖表會呼叫 ExecuteQuery API，根據實體中的任何資料或實體之間的關係進行查詢。知識圖表使用靈活的 PartiQL 查詢語言 （許多 AWS 服務使用），該語言有新增的圖形比對語法支援，可協助您撰寫查詢。進行呼叫後，您可以將結果檢視為資料表，或將其視覺化為已連線節點和邊緣的圖形。

知識圖表關鍵術語：

- 實體圖形：工作區內的節點和邊緣集合。
- 節點：工作區中的每個實體都會成為實體圖表中的節點。
- Edge：實體元件上定義的每個關係屬性都會成為實體圖表中的邊緣。此外，使用實體的 parentEntityId 欄位定義的階層父子關係也會成為實體圖表中具有「isChildOf」關係名稱的邊緣。所有邊緣都是方向邊緣。
- 關係： AWS IoT TwinMaker 關係是實體元件的特殊屬性類型。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，關係必須在實體的元件中定義。關係無法定義為隔離的資源。關係必須從一個實體指向另一個實體。

如何執行 AWS IoT TwinMaker 知識圖表查詢

使用 AWS IoT TwinMaker 知識圖表之前，請確定您已完成下列先決條件：

- 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。
- 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，您需要處於標準或分層套件定價模式。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

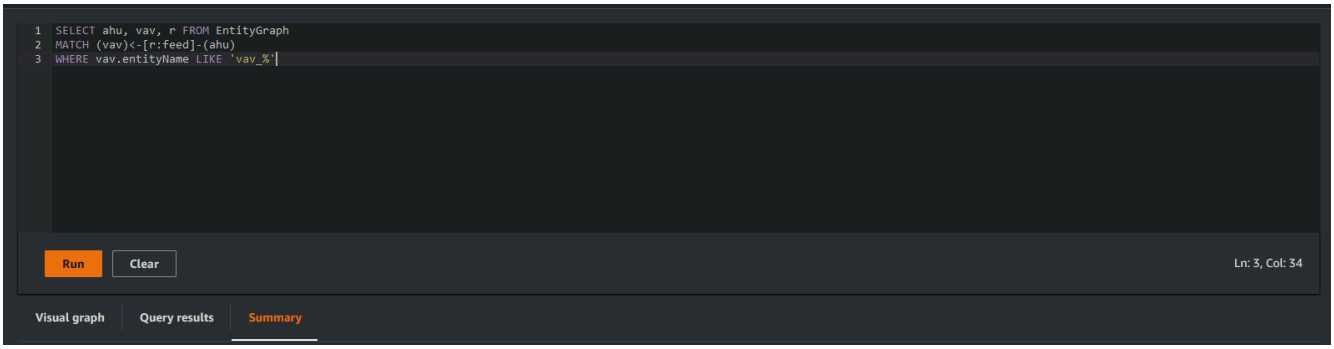
導覽至知識圖表查詢編輯器

1. 開啟 AWS IoT TwinMaker 主控台。
2. 開啟您要在其中使用知識圖表的工作區。
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。

執行查詢

執行查詢並產生圖形

1. 在查詢編輯器中，選擇編輯器索引標籤以開啟語法編輯器。
2. 在編輯器空間中，撰寫您要針對工作區資源執行的查詢。



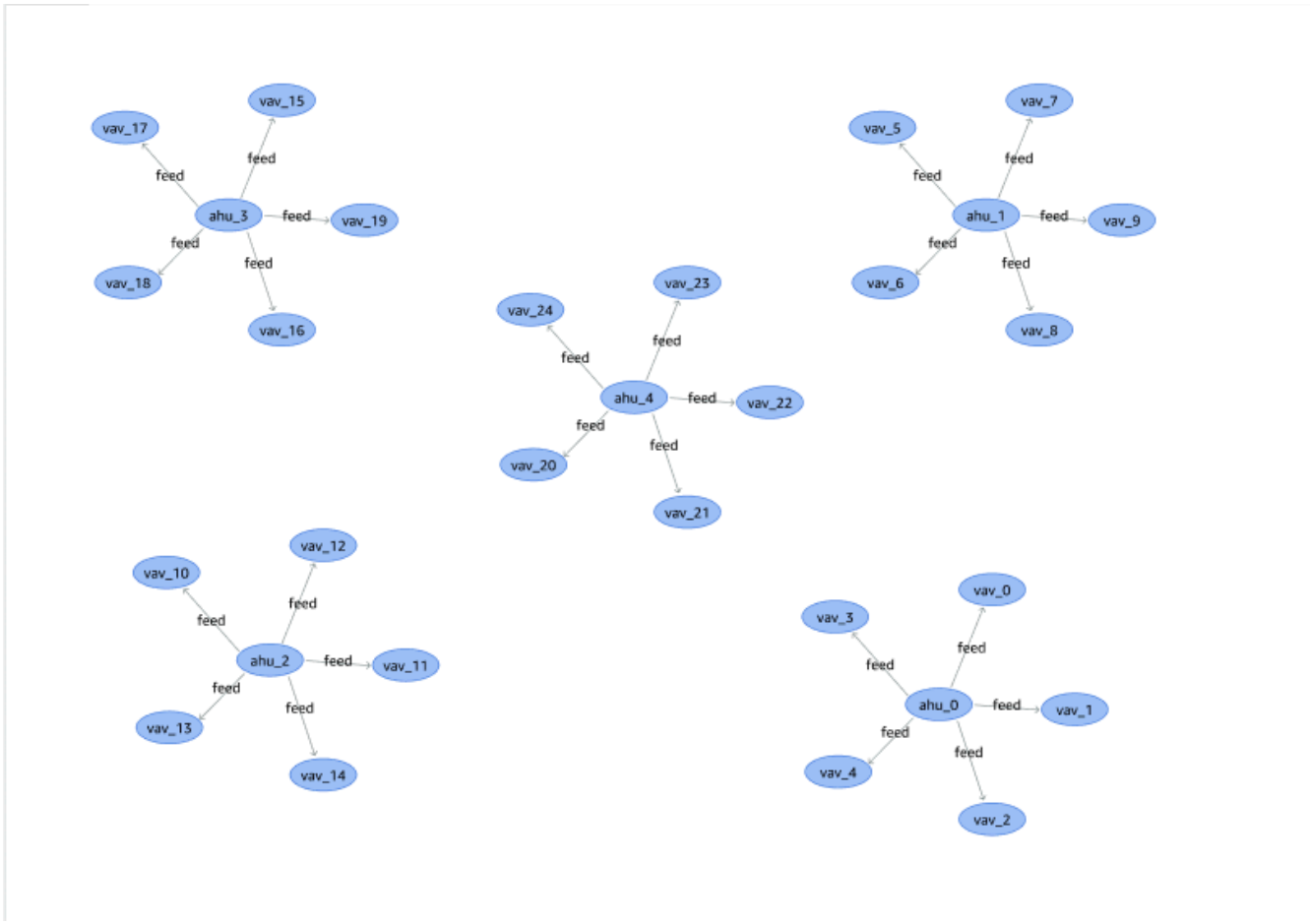
在顯示的範例中，請求會搜尋名稱vav_%中包含 的實體，然後使用下列程式碼，依實體feed之間的關係來組織這些實體。

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

> **Note**
>
> 知識圖表語法使用 PartiQL。如需此語法的資訊，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。

   圖形會根據您的請求產生。



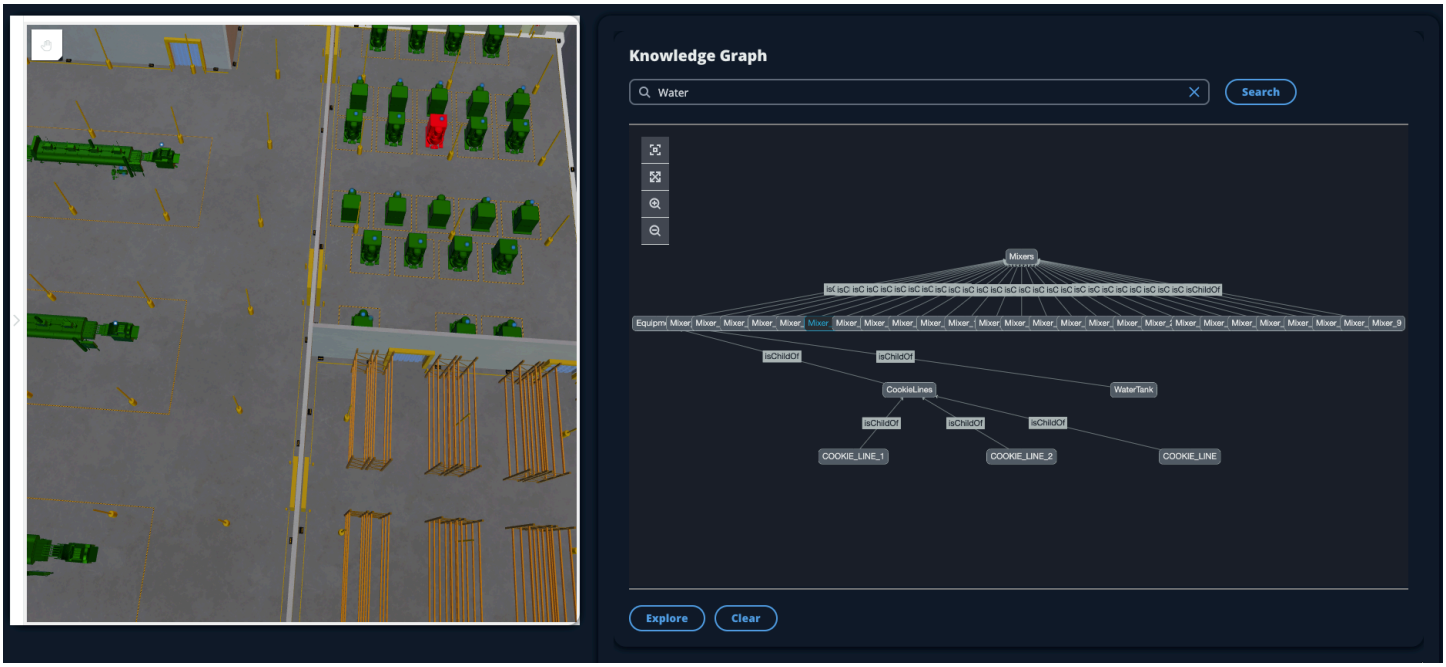
上面顯示的範例圖表是以步驟 2 中的查詢範例為基礎。

4. 查詢的結果也會在清單中顯示。選擇結果以檢視清單中的查詢結果。
5. 或者，選擇匯出為以 JSON 或 CSV 格式匯出查詢結果。

這涵蓋了 主控台中知識圖表的基本使用。如需示範知識圖表語法的詳細資訊和範例，請參閱 AWS IoT TwinMaker 知識圖表其他資源。

知識圖表場景整合

您可以使用 AWS IoT 應用程式套件元件來建置 Web 應用程式，將知識圖表整合到您的 AWS IoT TwinMaker 場景中。這可讓您根據場景中存在的 3D 節點（代表設備或系統的 3D 模型）產生圖形。若要建立從場景繪製 3D 節點圖形的應用程式，請先將 3D 節點繫結至工作區中的實體。透過此映射，會 AWS IoT TwinMaker 繪製場景中 3D 模型與工作區中實體之間的關係。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。



如需使用 AWS IoT 應用程式套件元件在 AWS IoT TwinMaker 場景中產生圖形的工作 Web 應用程式範例，請參閱 github 上的AWS IoT TwinMaker 範例反應應用程式。

AWS IoT TwinMaker 場景圖表先決條件

在建立在場景中使用 AWS IoT TwinMaker 知識圖表的 Web 應用程式之前，請完成下列先決條件：

- 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 AWS IoT 的應用程式套件元件。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。
- 熟悉知識圖表概念和關鍵術語。請參閱 AWS IoT TwinMaker 知識圖表核心概念。

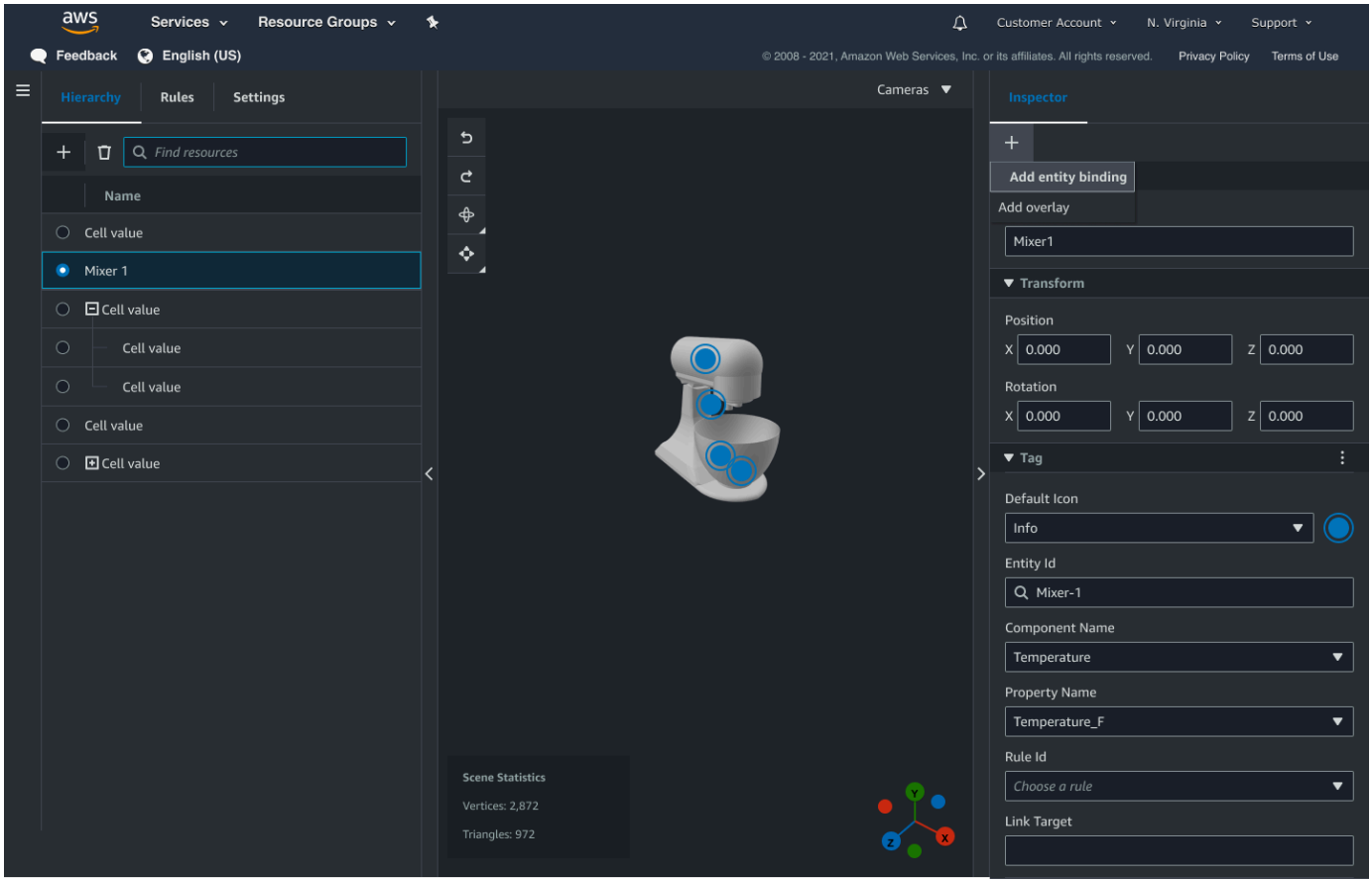
> **Note**
>
> 若要使用 AWS IoT TwinMaker 知識圖表和任何相關功能，您需要處於標準或分層套件定價模式。如需 AWS IoT TwinMaker 定價的詳細資訊，請參閱 切換 AWS IoT TwinMaker 定價模式。

在場景中繫結 3D 節點

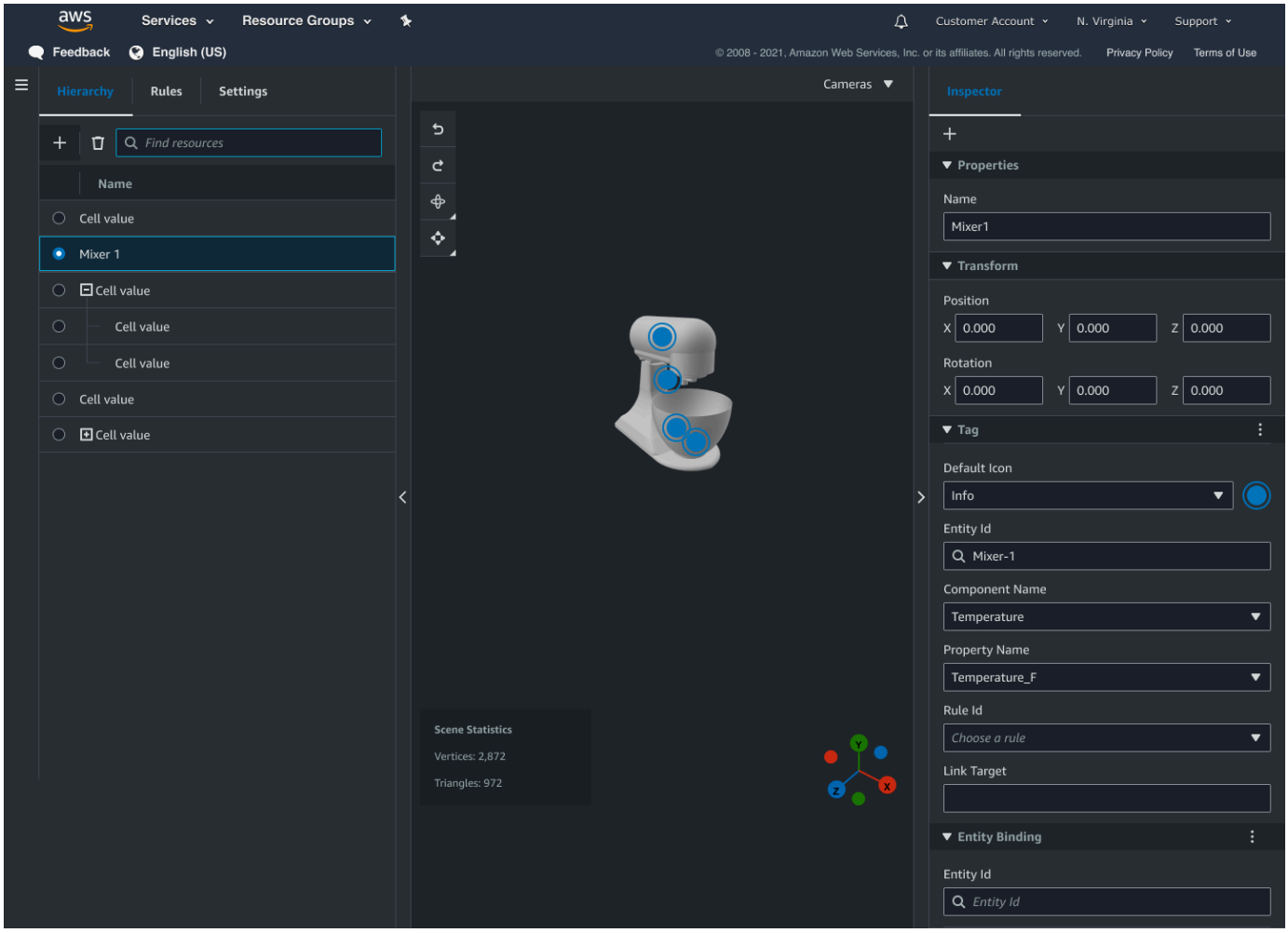
在建立整合知識圖表與場景的 Web 應用程式之前，請將場景中出現的 3D 模型繫結至相關聯的工作區實體，稱為 3D 節點。例如，如果您在場景中有混合器設備的模型，以及稱為 的對應實體`mixer_0`，請在混合器的模型與代表混合器的實體之間建立資料繫結，以便可以繪製模型和實體的圖形。

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，並選取具有您要繫結之 3D 節點的場景。
3. 在場景編寫器中選取節點 (3D 模型）。當您選取節點時，它會在畫面右側開啟檢測器面板。
4. 在檢測器面板中，導覽至面板頂端，然後選取 + 按鈕。然後選擇新增實體繫結選項。這會開啟下拉式清單，您可以在其中選取要繫結至目前所選節點的實體。



5. 從資料繫結下拉式功能表中，選取要映射至 3D 模型的實體 ID。針對元件名稱和屬性名稱欄位，選取您要繫結的元件和屬性。



一旦您選取實體 ID、元件名稱和屬性名稱欄位，繫結即完成。

6. 針對您要繪製圖形的所有模型和實體重複此程序。

> **Note**
>
> 您可以在場景標籤上執行相同的資料繫結操作，只需選取標籤而非實體，然後遵循相同的程序將標籤繫結至節點。

建立 Web 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表小工具的 Web 應用程式，讓您檢視場景並探索場景節點和實體之間的關係。

使用下列資源建立您自己的應用程式：

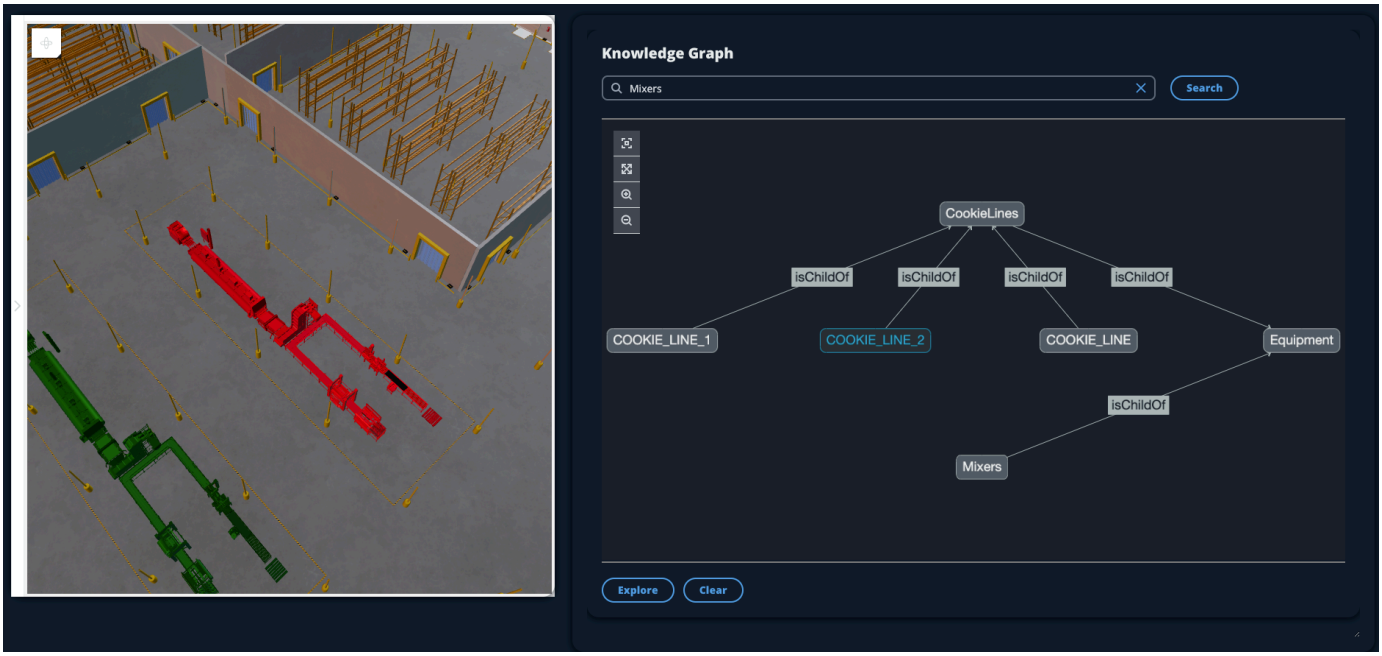
- AWS IoT TwinMaker 範例反應應用程式 github Readme 文件。
- github 上的 AWS IoT TwinMaker 範例反應應用程式來源。
- AWS IoT 應用程式套件入門文件。
- AWS IoT 應用程式套件影片播放器元件文件。
- AWS IoT 應用程式套件場景檢視器元件文件。

下列程序示範 Web 應用程式中場景檢視器元件的功能。

> **Note**
>
> 此程序是以 AWS IoT TwinMaker 範例反應應用程式中 AWS IoT 應用程式套件場景檢視器元件的實作為基礎。

1. 開啟 AWS IoT TwinMaker 範例反應應用程式的場景檢視器元件。在搜尋欄位中輸入實體名稱或部分實體名稱（區分大小寫搜尋），然後選取搜尋按鈕。如果模型繫結至實體 ID，則會反白顯示場景中的模型，並在場景檢視器面板中顯示實體的節點。

2. 若要產生所有關係的圖形，請在場景檢視器小工具中選取節點，然後選取探索按鈕。



3. 按下清除按鈕以清除您目前的圖形選擇並重新開始。

如何搭配 Grafana 使用 AWS IoT TwinMaker 知識圖表

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

AWS IoT TwinMaker 查詢編輯器先決條件

在 Grafana 中使用 AWS IoT TwinMaker 知識圖表之前，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。
- 設定 AWS IoT TwinMaker 以與 Grafana 搭配使用。如需說明，請參閱AWS IoT TwinMaker Grafana 儀表板整合。

> **Note**
>
> 若要使用 AWS IoT TwinMaker 知識圖表，您需要處於標準或分層套件定價模式。如需詳細資訊，請參閱切換 AWS IoT TwinMaker 定價模式。

AWS IoT TwinMaker 查詢編輯器許可

若要在 Grafana 中使用 AWS IoT TwinMaker 查詢編輯器，您必須擁有具有動作 許可的 IAM 角色`iottwinmaker:ExecuteQuery`。將該許可新增至工作區儀表板角色，如本範例所示：

JSON

```
{
    "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": "*"
        }
    ]
}
```

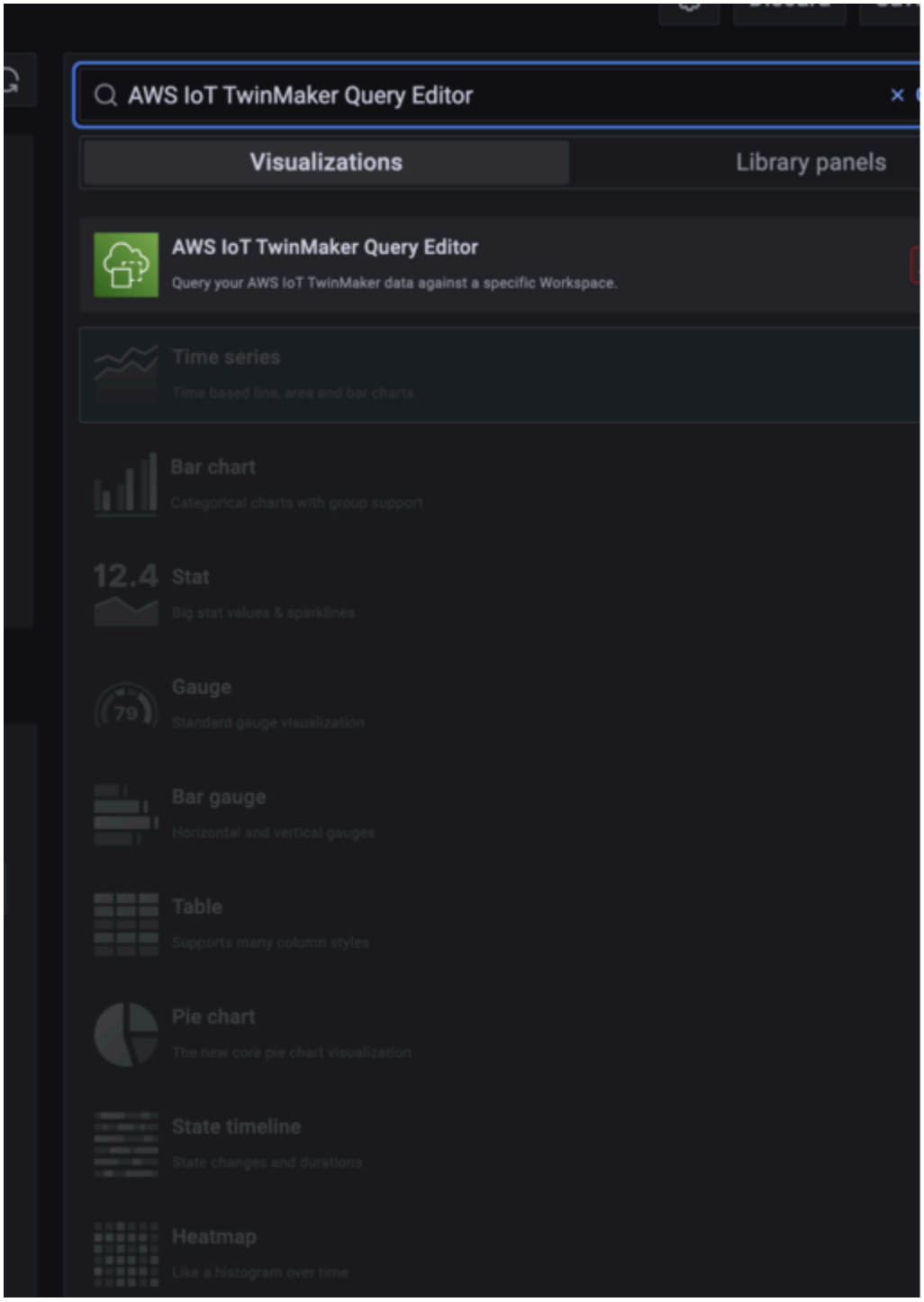
> **Note**
>
> 當您設定 AWS IoT TwinMaker Grafana 資料來源時，請務必將 角色用於擔任角色 ARN 欄位的此許可。新增後，您可以從工作區旁的下拉式清單中選取工作區。

如需詳細資訊，請參閱建立儀表板 IAM 角色。

設定 AWS IoT TwinMaker 查詢編輯器面板

為知識圖表設定新的 Grafana 儀表板面板

1. 開啟您的 AWS IoT TwinMaker Grafana 儀表板。
2. 建立新的儀表板面板。如需如何建立面板的詳細步驟，請參閱 Grafana 文件中的建立儀表板。
3. 從視覺化清單中，選取AWS IoT TwinMaker 查詢編輯器。



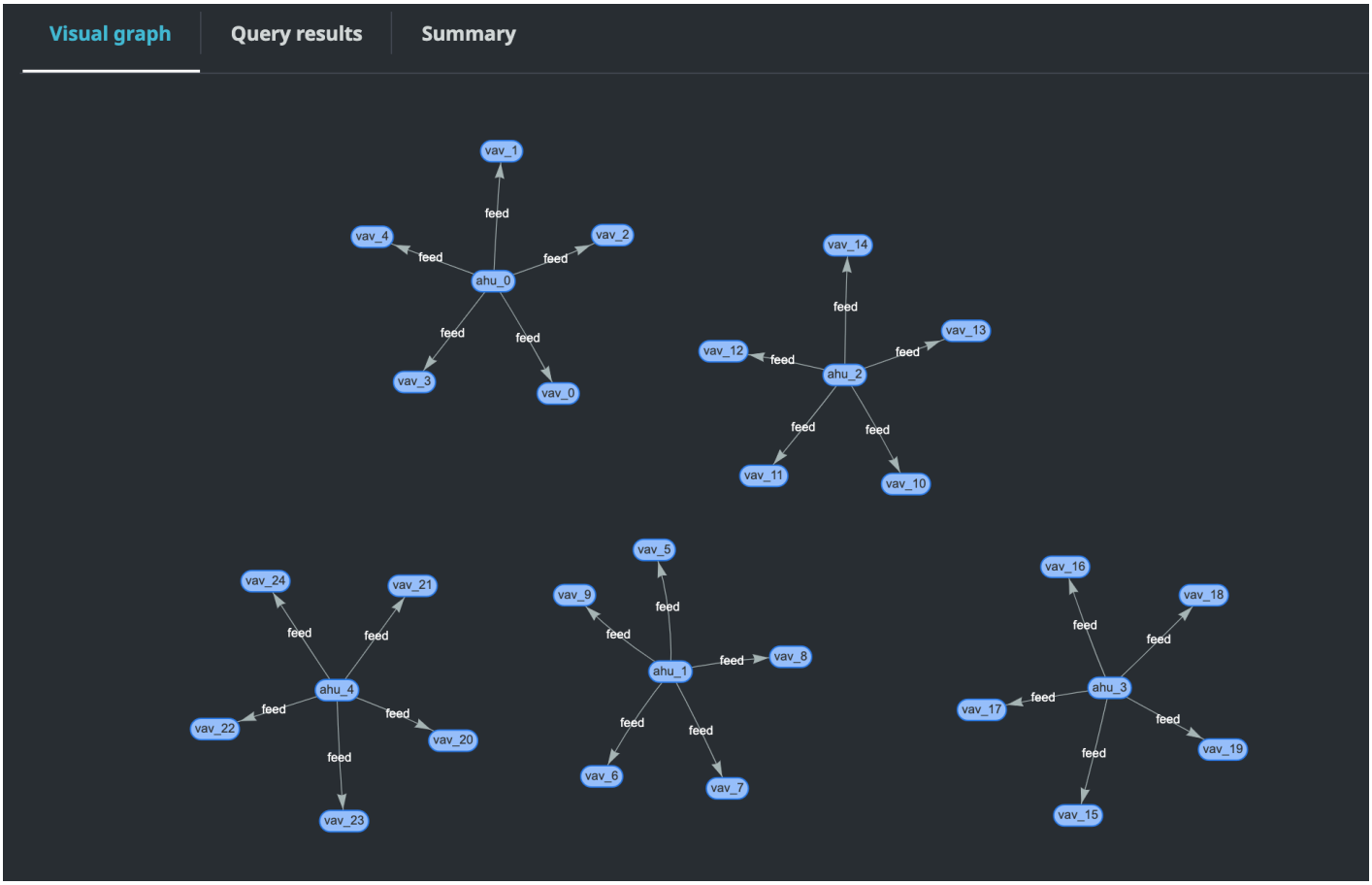
4. 選取要執行查詢的資料來源。
5. （選用）在提供的欄位中為新面板新增名稱。
6. 選取套用以儲存並確認您的新面板。

知識圖表面板的運作方式與 AWS IoT TwinMaker 主控台中提供的查詢編輯器類似。您可以在面板中執行、寫入和清除查詢。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

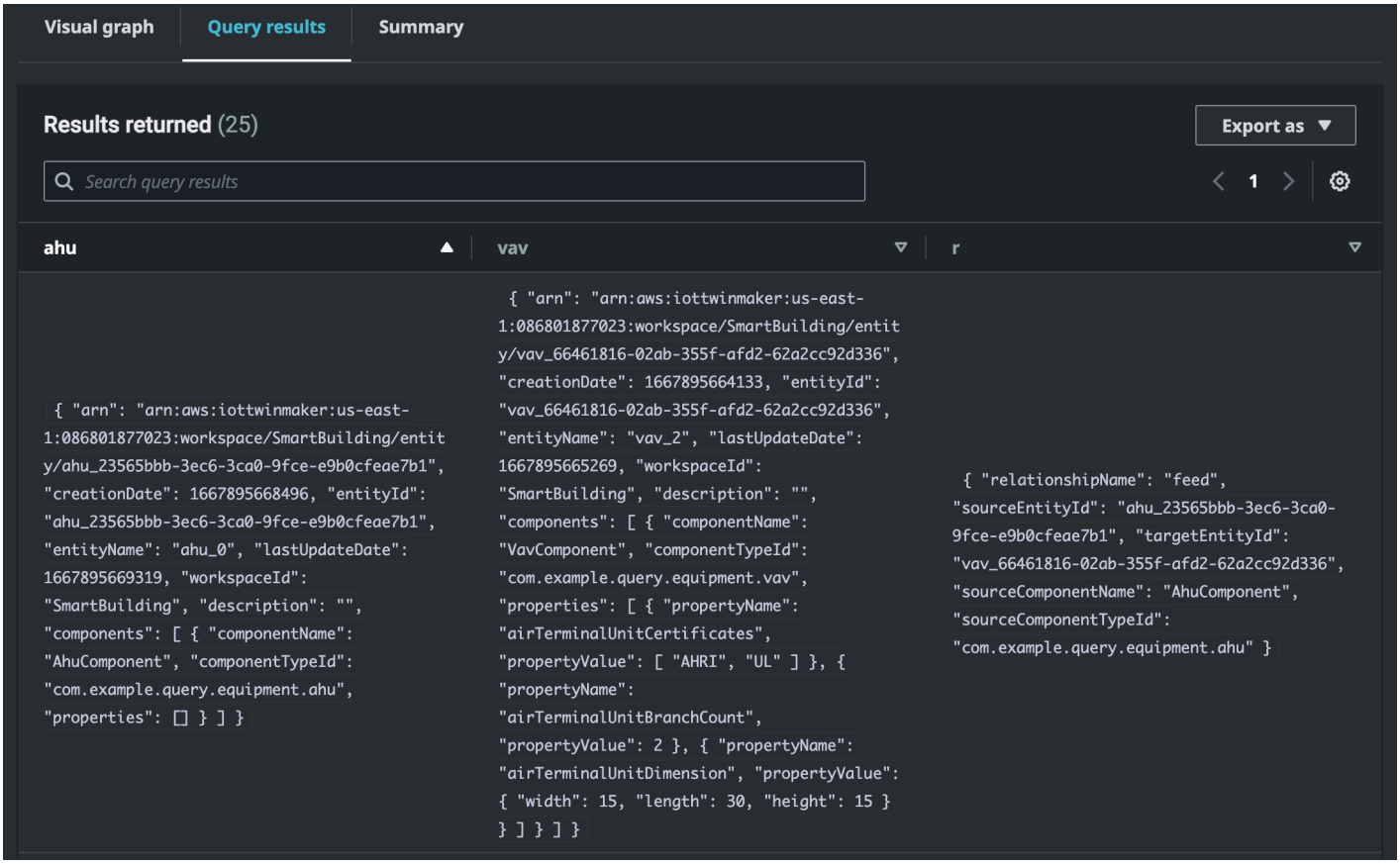
您的查詢結果會以三種方式顯示，如下圖所示：以圖形視覺化、在資料表中列出，或顯示為執行摘要。

- 圖形視覺化：



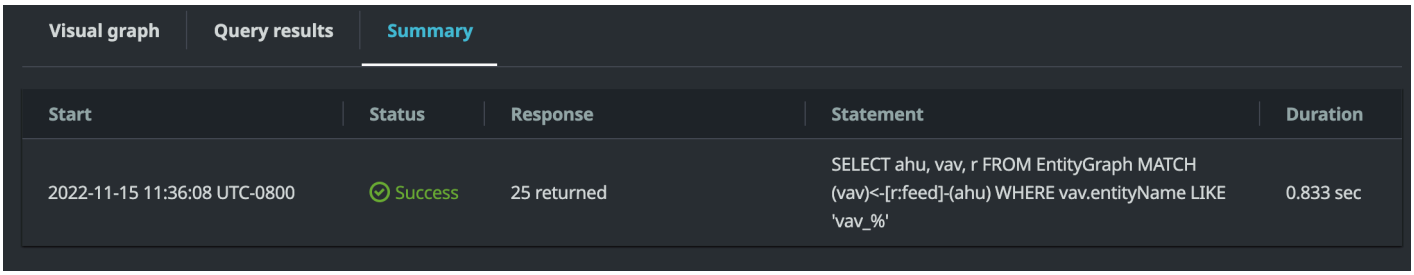
視覺化圖形只會針對結果中至少具有一個關係的查詢顯示資料。圖形會將實體顯示為節點，並將關係顯示為圖形中的導向邊緣。

- 表格式資料：



表格式資料格式會顯示所有查詢的資料。您可以搜尋資料表以取得特定結果或結果子集。資料可以 JSON 或 CSV 格式匯出。

- 執行摘要



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寫入查詢的 PartiQL 語法基本範例，以及提供知識圖表資料模型資訊之 PartiQL 文件的連結。

- PartiQL 圖形資料模型文件
- PartiQL 圖形查詢文件

這組範例顯示基本查詢及其回應。使用此做為撰寫您自己的查詢的參考。

基本查詢

- 使用篩選條件取得所有實體

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

此查詢會傳回工作區中名稱為 的所有實體room_0。

FROM 子句： EntityGraph 是包含工作區中所有實體及其關係的圖形集合。此集合是由 AWS IoT TwinMaker 根據您工作區中的實體自動建立和管理。

MATCH 子句：指定符合圖形一部分的模式。在這種情況下，模式(entity)符合圖形中的每個節點，並且繫結至實體變數。FROM 子句後面必須接著 MATCH子句。

WHERE 子句：指定節點 entityName 欄位的篩選條件，其中值必須符合 room_0。

SELECT 子句：指定entity變數，以便傳回整個實體節點。

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
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```


columnDescriptions 會傳回資料欄的中繼資料，例如名稱和類型。傳回的類型為 NODE。這表示已傳回整個節點。類型的其他值可以EDGE表示關係VALUE，也可以表示純量值，例如整數或字串。

rows 會傳回資料列清單。由於只有一個實體相符，rowData因此會傳回一個實體，其中包含實體中的所有欄位。

> **Note**
>
> 與只能傳回純量值的 SQL 不同，您可以使用 PartiQL 傳回物件 （ 做為 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 componentNamecomponentTypeId、 properties 以及 屬性層級欄位，例如 propertyName和 propertyValue，全部都是巢狀 JSON。

- 取得與篩選條件的所有關係：

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

此查詢會傳回關係名稱為 之工作區中的所有關係isLocationOf。

MATCH 子句：指定符合兩個節點 （ 由 表示( )) 的模式，這些節點由導向邊緣 （ 由 表示-[]->) 連接，並繫結至名為 的變數relationship。

WHERE 子句：指定邊緣 relationshipName 欄位的篩選條件，其中值為 isLocationOf。

SELECT 子句：指定關係變數，以便傳回整個邊緣節點。

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
    ]
}
```

中的資料欄類型columnDescriptions為 EDGE。

每個 rowData代表具有 等欄位的邊緣relationshipName。這與實體上定義的關係屬性名稱相同。sourceEntityId，sourceComponentName並sourceComponentTypeId提供有關關係屬性在哪個實體和元件上定義的資訊。targetEntityId 指定此關係指向的實體。

- 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'
```

此查詢會傳回與實體有isLocationOf關係之所有實體的所有room_0實體名稱。

MATCH 子句：指定符合任何兩個具有導向邊緣的節點 (e1、e2) 的模式 (r)。

WHERE 子句：指定關係名稱和來源實體名稱的篩選條件。

SELECT 子句：傳回e2節點中的 entityName 欄位。

回應

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

在 columnDescriptions 中，資料欄的類型為 ，VALUE因為 entityName 是字串。

floor_0傳回一個實體 。

配對

子MATCH句支援下列模式：

- 指向節點 'a' 的相符節點 'b'：

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

- 比對指向節點 'b' 的節點 'a'：

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

假設不需要在關係上指定篩選條件，則關係沒有變數繫結。

- 比對指向節點 'b' 的節點 'a' 和指向節點 'a' 的節點 'b'：

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

這將傳回兩個相符項目：一個從 'a' 到 'b'，另一個從 'b' 到 'a'，因此建議盡可能使用導向邊緣。

- 關係名稱也是 屬性圖形 的標籤EntityGraph，因此您可以直接在冒號 (：) 後面指定關係名稱，而不是在 WHERE子句rel.relationshipName中指定篩選條件。

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

- 鏈結：模式可以鏈結以符合多個關係。

```
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)
```

此查詢會比對 1 到 5 個躍點內節點「a」傳出邊緣的任何模式。允許的量化指標為：

{m,n} - 介於 m 和 n 重複之間

{m,} - m 或多個重複。

從：

實體節點可包含巢狀資料，例如元件本身包含其他巢狀資料，例如屬性。您可以透過將 MATCH 模式的結果解除巢狀來存取這些項目。

```
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'
```

將 點.入變數以存取巢狀欄位。逗號 (,) 用於取消巢狀化 （或聯結） 實體，其中包含這些元件內的元件，以及這些元件內的屬性。 AS 用於將變數繫結至未巢狀變數，以便在 WHERE或 SELECT子句中使用。此查詢會傳回包含名為 且元件類型 ID 為 之元件meeting中roomFunction值為 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，請使用逗號標記法進行聯結。

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

SELECT：

- 傳回節點：

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

- 傳回邊緣：

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

- 傳回純量值：

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

  使用 將輸出欄位命名為別名，以格式化輸出欄位的名稱AS。在這裡，roomfunction傳回的不是回應中的propertyValue資料欄名稱。

- 傳回別名：

  ```
  SELECT floor.entityName AS floorName, luminaire.entityName as luminaireName
  FROM EntityGraph MATCH (floor)-[:isLocationOf]-(room)-[:hasPart]-
  (lightingZone)-[:feed]-(luminaire)
  ```

```
WHERE floor.entityName = 'floor_0'
AND luminaire.entityName like 'lumin%'
```

強烈建議使用別名明確、提高可讀性，並避免查詢中的任何模棱兩可之處。

WHERE：

- 支援的邏輯運算子為 AND、 NOT和 OR。
- 支援的比較運算子為 <、<=、=>=、 >和 !=。
- 如果您想要在相同欄位中指定多個OR條件，請使用 IN關鍵字。
- 篩選實體、元件或屬性欄位：

```
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 configuration 屬性。unit 以下是組態映射中的 金鑰， Celsius是 值。

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

- 檢查映射屬性是否包含指定的索引鍵和值：

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

- 檢查映射屬性是否包含指定的金鑰：

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

- 檢查清單屬性是否包含指定的值：

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

- 使用 lower()函數進行不區分大小寫的比較。根據預設，所有比較都會區分大小寫。

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

LIKE：

如果您不知道欄位的確切值，並且可以對指定的欄位執行全文搜尋，則很有用。 %代表零或多個。

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

- 修正搜尋： %room%
- 字首搜尋： room%
- 尾碼搜尋： %room
- 如果您的值中有「%」，請在 中放置逸出字元LIKE，並使用 指定逸出字元ESCAPE。

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

DISTINCT：

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

- DISTINCT 關鍵字會消除最終結果中的重複項目。

  DISTINCT 不支援複雜資料類型。

COUNT

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

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

  例如，不支援 COUNT(DISTINCT e.entityId)。

PATH

使用路徑投影查詢時支援下列模式投影：

- 變數躍點查詢

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

  此查詢會比對任何模式的節點中繼資料，並在 1 到 3 個躍點內從節點傳出邊緣。

- 已修正跳轉查詢

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

此查詢符合並將實體的中繼資料和傳入邊緣投影至 b。

- 無導向的查詢

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

此查詢會以透過 b 連接 和 c 的 1 個跳轉模式比對和投影節點的中繼資料。

```
{
    "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查詢回應僅包含透過 b 識別 和 c 之間每個路徑/模式的所有節點和邊緣的中繼資料。

LIMIT 和 OFFSET：

```
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。

LIMIT 和 maxResults：

下列範例顯示的查詢總共傳回 500 個結果，但每個 API 呼叫一次只會顯示 50 個結果。當您需要限制顯示的結果數量時，可以使用此模式，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 呼叫傳回的結果數量，而不限制傳回的結果總數，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邊界內讀取結果。

  在查詢max-results中使用 可讓您減少顯示的結果數量，而不會限制實際傳回的結果數量。

下面的查詢會逐一查看結果的下一頁。此查詢使用 ExecuteQuery API 呼叫傳回資料列 51-100，其中 會指定下一頁的結果 next-token– 在此情況下，字符為："H7kyGmvK376L"。

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

- next-token 字串會指定結果的下一頁。如需詳細資訊，請參閱 ExecuteQuery API 動作。

AWS IoT TwinMaker 知識圖表查詢有下列限制：

限制名稱	配額	可調整
查詢執行逾時	10 秒	否
躍點數量上限	10	是
自我數量上限 JOIN	20	是
預計欄位數量上限	20	是
條件式表達式數目上限 (AND、OR、NOT)	10	是
LIKE 表達式模式的長度上限（包括萬用字元和逸出）	20	是
可在 IN子句中指定的項目數量上限	10	是
的最大值 OFFSET	3000	是

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

與 的資產同步 AWS IoT SiteWise

AWS IoT TwinMaker 支援資產 AWS IoT SiteWise 和資產模型的資產同步 （ 資產同步 ）。使用 AWS IoT SiteWise 元件類型，資產同步會取得現有的 AWS IoT SiteWise 資產和資產模型，並將這些資源轉換為 AWS IoT TwinMaker 實體、元件和元件類型。下列各節會逐步解說如何設定資產同步，以及哪些 AWS IoT SiteWise 資產和資產模型可以同步到您的 AWS IoT TwinMaker 工作區。

主題

- 搭配 使用資產同步 AWS IoT SiteWise
- 自訂和預設工作區之間的差異
- 從 同步的資源 AWS IoT SiteWise
- 分析同步狀態和錯誤
- 刪除同步任務
- 資產同步限制

搭配 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。根據您正在使用的工作區類型，遵循適當的程序。

> **Important**
>
> the section called “自訂和預設工作區之間的差異” 如需自訂和預設工作區之間差異的相關資訊，請參閱 。

主題

- 使用自訂工作區
- 使用 IoTSiteWiseDefaultWorkspace

使用自訂工作區

請先檢閱這些先決條件，再開啟資產同步。

先決條件

使用 之前 AWS IoT SiteWise，必須先完成下列項目：

- 您有一個 AWS IoT TwinMaker 工作區。
- 您在 中有資產和資產模型 AWS IoT SiteWise。如需詳細資訊，請參閱建立資產模型。
- 具有下列 AWS IoT SiteWise 動作讀取許可的現有 IAM 角色：
  - `ListAssets`
  - `ListAssetModels`
  - `DescribeAsset`
  - `DescribeAssetModel`
- IAM 角色必須具有下列 的寫入許可 AWS IoT TwinMaker：
  - `CreateEntity`
  - `UpdateEntity`
  - `DeleteEntity`
  - `CreateComponentType`
  - `UpdateComponentType`
  - `DeleteComponentType`
  - `ListEntities`
  - `GetEntity`
  - `ListComponentTypes`

  使用下列 IAM 角色做為必要角色的範本：

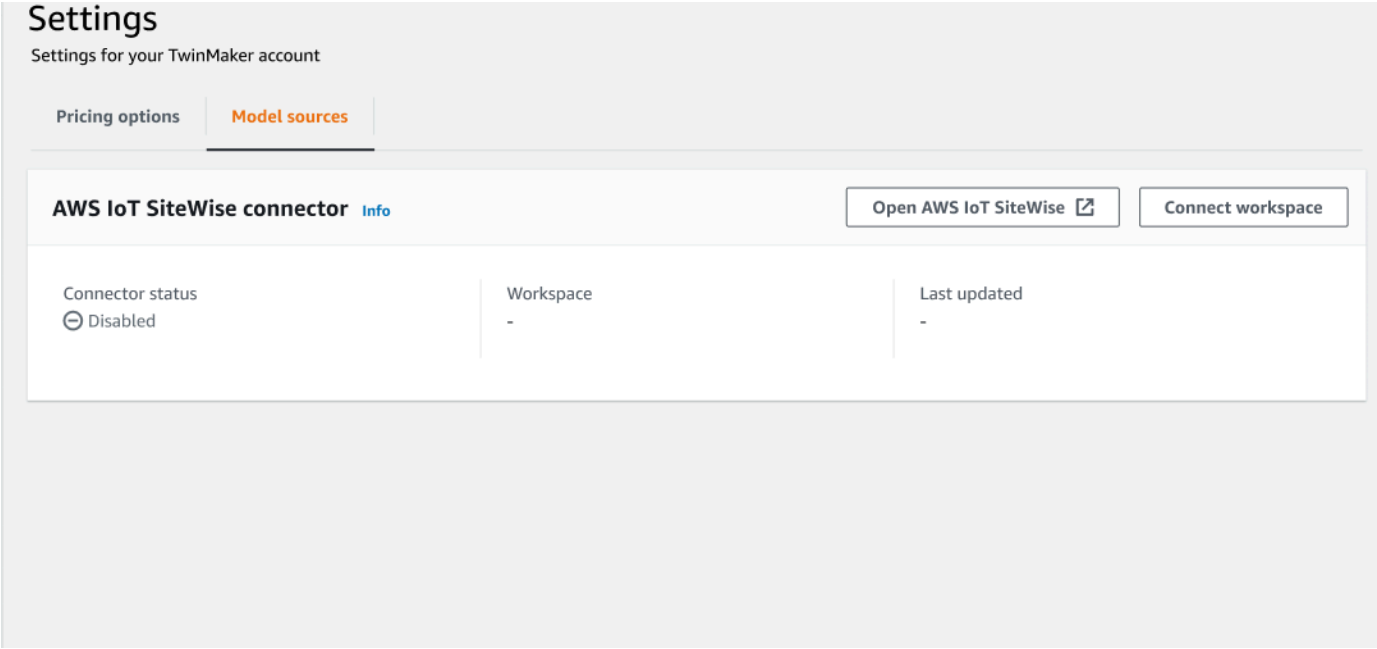
```
// trust relationships
 {
    {
        "Version": "2012-10-17",
        "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
            "Service": [
            "iottwinmaker.amazonaws.com"
```

```
            ]
        },
            "Action": "sts:AssumeRole"
        }
    ]
}

// permissions - replace ACCOUNT_ID, REGION, WORKSPACE_ID with actual values
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "SiteWiseAssetReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAsset"
            ],
            "Resource": [
                "arn:aws:iotsitewise:REGION:ACCOUNT_ID:asset/*"
            ]
        },
        {
            "Sid": "SiteWiseAssetModelReadAccess",
            "Effect": "Allow",
            "Action": [
                "iotsitewise:DescribeAssetModel"
            ],
            "Resource": [
                "arn:aws:iotsitewise:REGION:ACCOUNT_ID: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:REGION:ACCOUNT_ID:workspace/WORKSPACE_ID",
                "arn:aws:iottwinmaker:REGION:ACCOUNT_ID:workspace/WORKSPACE_ID/*"

            ]
        }
    ]
}
```

使用下列程序來開啟和設定 AWS IoT SiteWise 資產同步。

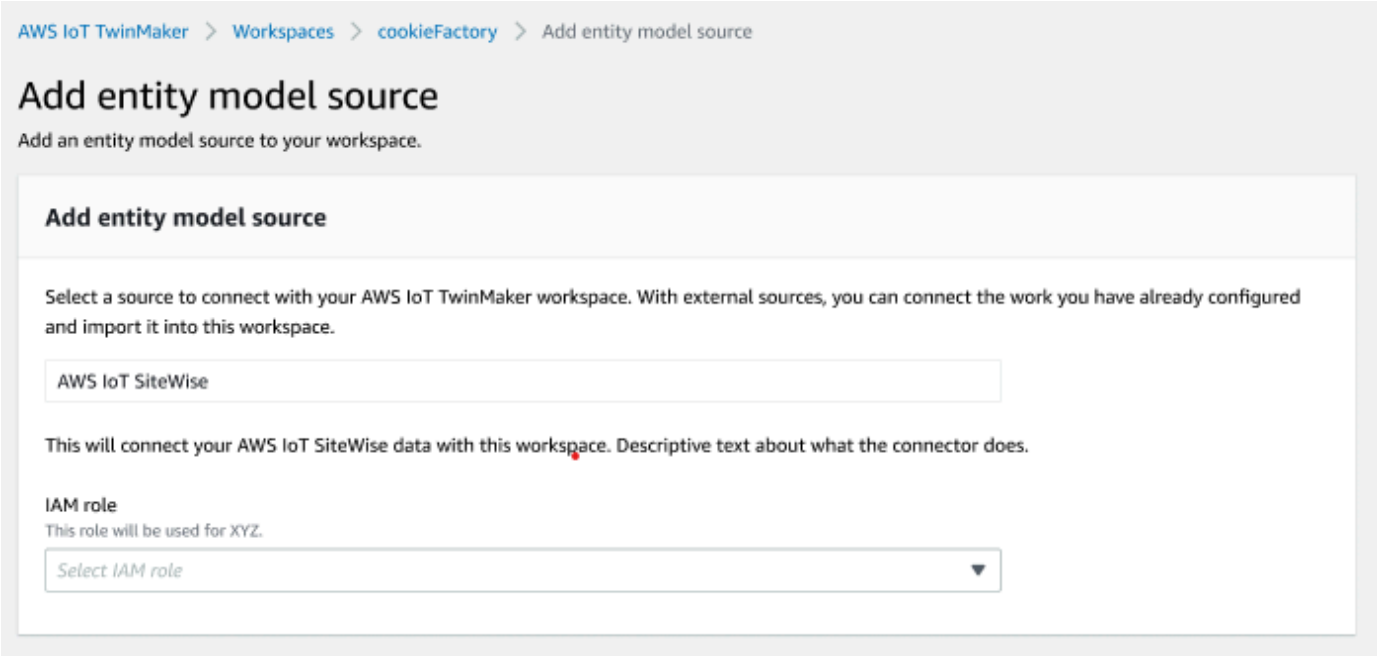
1. 在 AWS IoT TwinMaker 主控台中，導覽至設定頁面。
2. 開啟模型來源索引標籤。



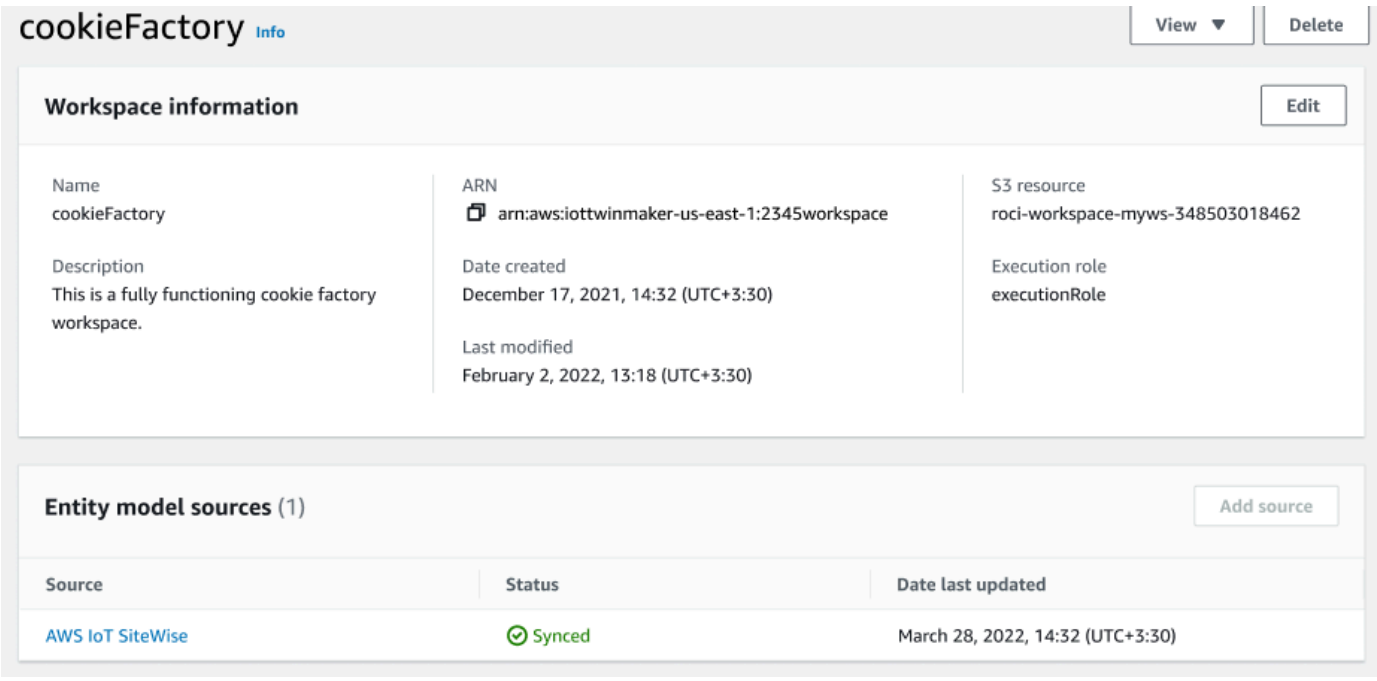
3. 選擇連線工作區，將您的 AWS IoT TwinMaker 工作區連結至您的 AWS IoT SiteWise 資產。

> **Note**
>
> 您只能搭配單一 AWS IoT TwinMaker 工作區使用資產同步。如果您想要在不同工作區中同步，您必須中斷同步與某個工作區的連線，並連線至另一個工作區。

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。



6. 在新增實體模型來源頁面上，確認來源欄位顯示 AWS IoT SiteWise。選取您建立做為 IAM 角色先決條件的 IAM 角色。
7. 您現在已開啟 AWS IoT SiteWise 資產同步。您應該會在選取的工作區頁面頂端看到合規橫幅，確認資產同步處於作用中狀態。您也應該會在實體模型來源區段中看到列出的同步來源。



使用 IoTSiteWiseDefaultWorkspace

當您選擇加入AWS IoT SiteWiseAWS IoT TwinMaker 整合時，IoTSiteWiseDefaultWorkspace會建立名為 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 來建立名為 的工作區IoTSiteWiseDefaultWorkspace。

先決條件

建立 之前IoTSiteWiseDefaultWorkspace，請確定您已執行下列動作：

- 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，並確認其具有 的許可iotsitewise:EnableSiteWiseIntegration。

  如有需要，請將許可新增至角色或使用者：

  JSON

```
{
    "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。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幾個顯著差異。

- 當您建立預設工作區時，Amazon S3 位置和 IAM 角色是選用的。

> **Note**
>
> 您可以使用 UpdateWorkspace提供 Amazon S3 位置和 IAM 角色。

- IoTSiteWiseDefaultWorkspace 沒有要同步資源的資源計數限制 AWS IoT SiteWise AWS IoT TwinMaker。
- 當您從 同步資源時 AWS IoT SiteWise，其 SyncSource 會是 SITEWISE_MANAGED。這包括 Entities和 ComponentTypes。
- 等新 AWS IoT SiteWise 功能CompositionModel僅在 中提供IoTSiteWiseDefaultWorkspace。

有幾個特定限制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 AWS IoT TwinMaker 刪除 中的對應資源。

從 同步的資源 AWS IoT SiteWise

本主題列出您可以從 同步 AWS IoT SiteWise 到 AWS IoT TwinMaker 工作區的資產。

> **Important**
>
> 自訂和預設工作區之間的差異 如需自訂和預設工作區間差異的相關資訊，請參閱 。

自訂和預設工作區

下列資源已同步，並可在自訂和預設工作區中使用：

資產模型

AWS IoT TwinMaker 在 中為每個資產模型建立新的元件類型 AWS IoT SiteWise。

- 資產模型TypeId的 元件將使用下列其中一種模式：

    - 自訂工作區 - iotsitewise.assetmodel:*assetModelId*
    - 預設工作區 - *assetModelId*
- 資產模型中的每個屬性都是元件類型中的新屬性，具有下列其中一個命名模式：
    - 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 類型的新屬性，LIST而 nestedType 位於 元件類型RELATIONSHIP中。階層會對應至名稱字首為下列其中一項的 屬性：
    - 自訂工作區 - *Hierarchy_hierarchyId*
    - 預設工作區 - *hierarchyId*

資產

AWS IoT TwinMaker 會為其中的每個資產建立新的實體 AWS IoT SiteWise。

- 與 assetId中的 entityId 相同 AWS IoT SiteWise。
- 這些實體有一個名為 的單一元件sitewiseBase，其元件類型對應至此資產的資產模型。
- 任何資產層級覆寫，例如設定屬性別名或度量單位，都會反映在 實體中 AWS IoT TwinMaker。

僅限預設工作區

下列資產已同步，且僅適用於預設工作區：IoTSiteWiseDefaultWorkspace。

AssetModelComponents

AWS IoT TwinMaker 會為每個 AssetModelComponents 中的 建立新的元件類型 AWS IoT SiteWise。

- 資產模型TypeId的 元件使用以下模式：assetModelId。
- 資產模型中的每個屬性都是元件類型中的新屬性，屬性名稱為 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 類型的新屬性，LIST而 nestedType 位於 元件類型RELATIONSHIP中。階層會映射至名稱字首為 的 屬性hierarchyId。

AssetModelCompositeModel

AWS IoT TwinMaker 會為每個AssetModelCompositeModel輸入建立新的元件類型 AWS IoT SiteWise。

- 資產模型TypeId的 元件使用以下模式：assetModelId_assetModelCompositeModelId。
- 資產模型中的每個屬性都是元件類型中的新屬性，屬性名稱為 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 會為每個 AssetCompositeModel 中的 建立新的複合元件 AWS IoT SiteWise。

- 與 assetModelCompositeModelId中的 componentName 相同 AWS IoT SiteWise。

資源未同步

下列資源不會同步：

非同步資產和資產模型

- 警示模型將同步為 compositeModels但與警示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。只會同步在資產模型中建模的屬性。
- 屬性、測量、轉換、彙總和中繼資料計算的屬性值，例如公式和視窗不會同步。只會同步有關屬性的中繼資料，例如別名、度量單位和資料類型。您可以使用一般 AWS IoT TwinMaker 資料連接器 API GetPropertyValueHistory 查詢這些值。

在 中使用同步實體和元件類型 AWS IoT TwinMaker

從中同步資產後 AWS IoT SiteWise，同步的元件類型只會讀取 AWS IoT TwinMaker。任何更新或刪除動作都必須在 中完成 AWS IoT SiteWise， AWS IoT TwinMaker 如果 syncJob 仍在作用中，這些變更會同步至 。

同步的實體和 AWS IoT SiteWise 基本元件也會唯讀 AWS IoT TwinMaker。您可以將其他非同步元件新增至同步的實體，只要沒有實體層級屬性，例如描述或entityName已更新。

有些限制適用於如何與同步實體互動。您無法在同步實體階層中的同步實體下建立子實體。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 或者您刪除同步任務，其他元件會與實體一起刪除。

您可以在 Grafana 儀表板中使用這些同步實體，並將其新增為場景編寫器中的標籤，例如一般實體。您也可以為這些同步的實體發出知識圖表查詢。

> **Note**
>
> 不會收取未修改的同步實體費用，但如果已進行變更，則會向您收取這些實體的費用 AWS IoT TwinMaker。例如，如果您將非同步元件新增至同步的實體，該實體現在會計費 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” 如需自訂和預設工作區之間差異的相關資訊，請參閱 。

同步任務狀態

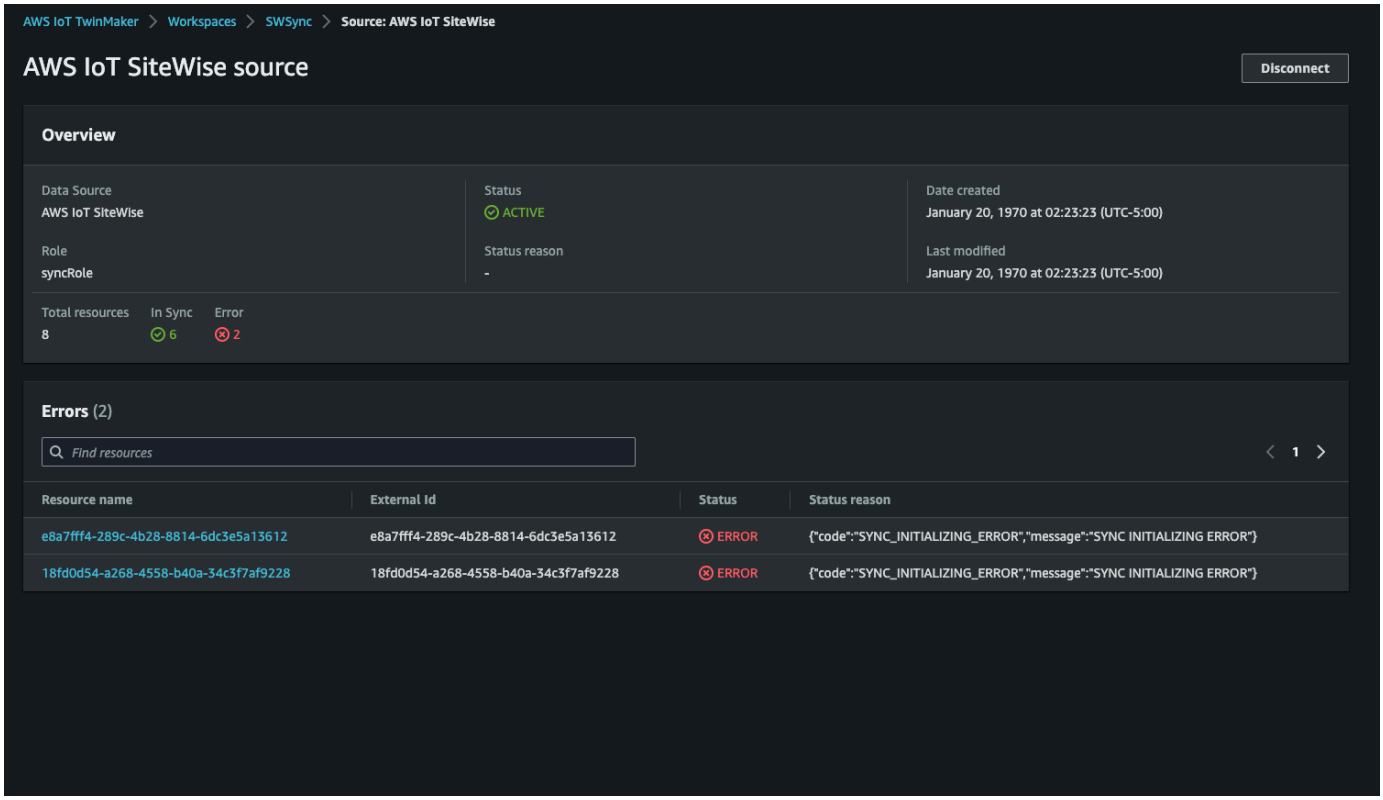
根據同步任務的狀態，同步任務具有下列其中一種狀態。

- 同步任務`CREATING`狀態表示任務正在檢查許可，並從 載入資料 AWS IoT SiteWise 以準備同步。
- 同步任務`INITIALIZING`狀態表示 中的所有現有資源 AWS IoT SiteWise 都會同步至 AWS IoT TwinMaker。如果使用者擁有大量資產和資產模型，此步驟可能需要更長的時間才能完成 AWS IoT SiteWise。您可以在 AWS IoT TwinMaker 主控台中檢查同步任務，或呼叫 `ListSyncResources` API 來監控已同步的資源數量。
- 同步任務`ACTIVE`狀態表示初始化步驟已完成。任務現在已準備好同步來自 的任何新更新 AWS IoT SiteWise。
- 同步任務`ERROR`狀態表示具有上述任何狀態的錯誤。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任務，並使用新角色建立新的任務。

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資料表存取。模型來源頁面會顯示無法同步的資源清單。大多數錯誤都會由同步任務自動重試，但如果資源需要 動作，則會保持 `ERROR` 狀態。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的錯誤，請使用下列程序。

1. 在 AWS IoT TwinMaker 主控台中導覽至您的工作區。
2. 選取實體模型 AWS IoT SiteWise 來源模態中列出的來源，以開啟資產同步詳細資訊頁面。



3. 如上述螢幕擷取畫面所示，任何持續發生錯誤的資源都會列在錯誤表格中。您可以使用此表格來追蹤和修正與特定資源相關的錯誤。

可能的錯誤包括下列項目：

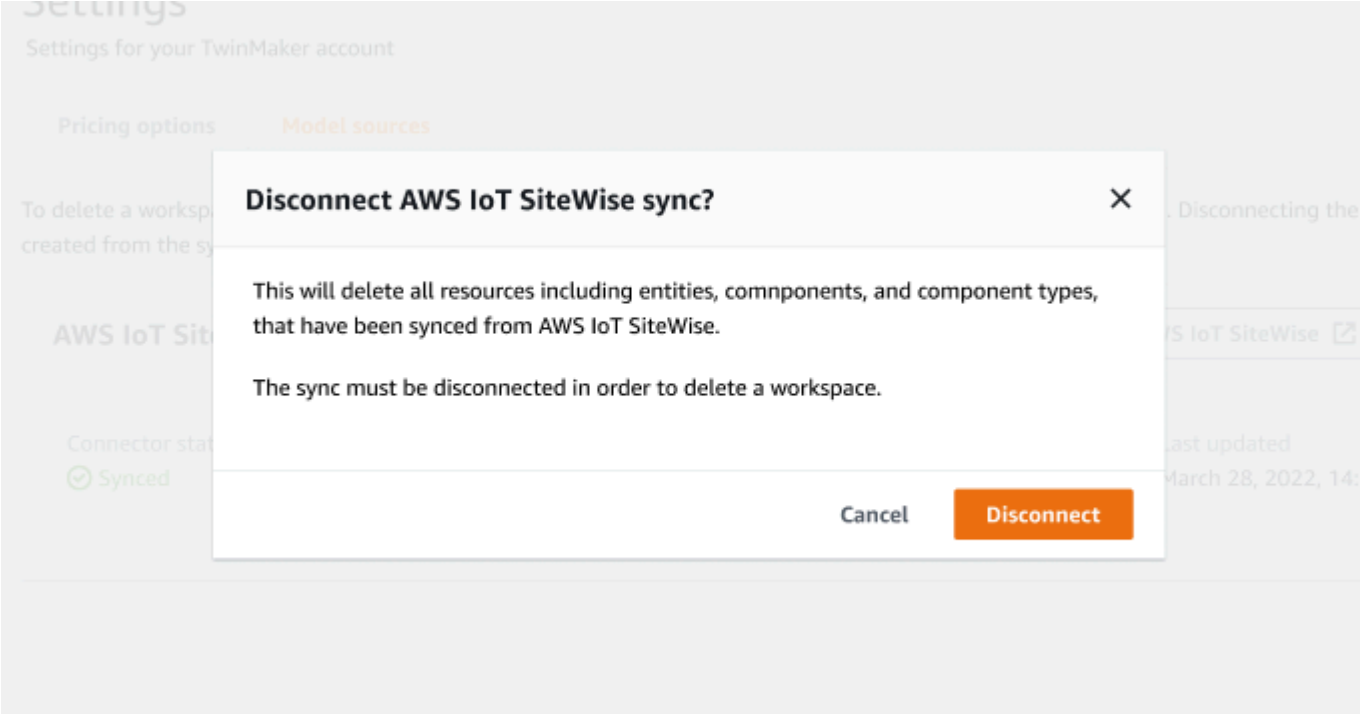
- 雖然 AWS IoT SiteWise 支援重複的資產名稱，但 AWS IoT TwinMaker 僅在 R00T層級支援它們，而不是在相同的父實體下。如果您在 的父實體下有兩個同名資產 AWS IoT SiteWise，其中一個資產無法同步。若要修正此錯誤，請在同步 AWS IoT SiteWise 之前刪除其中一個資產，或在不同的父項資產下移動其中一個資產。
- 如果您已有與資產 ID 具有相同 ID 的 AWS IoT SiteWise 實體，該資產會無法同步，直到您刪除現有的實體為止。

刪除同步任務

使用下列程序刪除同步任務。

> **Important**
>
> the section called “自訂和預設工作區之間的差異” 如需自訂和預設工作區之間差異的相關資訊，請參閱 。

1. 導覽至 AWS IoT TwinMaker 主控台。
2. 開啟您要從中刪除同步任務的工作區。
3. 在實體模型來源下，選取要 AWS IoT SiteWise 開啟來源詳細資訊頁面的來源。
4. 若要停止同步任務，請選擇中斷連線。確認您的選擇，以完全刪除同步任務。



刪除同步任務後，您可以在相同或不同的工作區中再次建立同步任務。

如果工作區中有任何同步任務，則無法刪除工作區。在刪除工作區之前，請先刪除同步任務。

如果在刪除同步任務期間發生任何錯誤，同步任務會保持 DELETING 狀態，並自動重試。如果有任何與刪除資源相關的錯誤，您現在可以手動刪除任何同步的實體或元件類型。

> **Note**
>
> 首先 AWS IoT SiteWise 刪除從 同步的任何資源，然後刪除同步任務本身。

資產同步限制

> **Important**
>
> the section called “自訂和預設工作區之間的差異” 如需自訂和預設工作區之間差異的相關資訊，請參閱 。

由於AWS IoT SiteWise 配額高於預設AWS IoT TwinMaker 配額，因此我們正在增加從 同步的實體和元件類型的下列限制 AWS IoT SiteWise。

- 工作區中的 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 100，000 個同步的實體，因為它只能從中同步 100，000 個資產 AWS IoT SiteWise。
- 每個父實體最多 2000 個子實體。它會同步每個單一父系資產 2000 個子資產。

  > **Note**
  >
  > GetEntity API 只會傳回階層屬性的前 50 個子實體，但您可以使用 GetPropertyValue API 來分頁和擷取所有子實體的清單。

- 每個同步元件 600 個屬性 AWS IoT SiteWise，可同步具有 600 個屬性和階層的資產模型。

> **Note**
>
> 這些限制僅適用於同步的實體。如果您需要增加非同步資源的這些限制，請請求增加配額。

AWS IoT TwinMaker Grafana 儀表板整合

AWS IoT TwinMaker 透過應用程式外掛程式支援 Grafana 整合。使用 Grafana 10.4.0 版和更新版本與您的數位分身應用程式互動。 AWS IoT TwinMaker 外掛程式提供自訂面板、儀表板範本和資料來源，以連線至您的數位分身資料。

如需如何加入 Grafana 並設定儀表板許可的詳細資訊，請參閱下列主題：

主題

- Grafana 場景檢視器的 CORS 組態
- 設定您的 Grafana 環境
- 建立儀表板 IAM 角色
- 建立 AWS IoT TwinMaker 影片播放器政策

> **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 使用者介面從儲存貯體載入資源。如需指示，請參閱Grafana 場景檢視器的 CORS 組態。

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
- 影片播放器面板

Grafana 場景檢視器的 CORS 組態

Grafana AWS IoT TwinMaker 外掛程式需要 CORS （跨來源資源共用） 組態，允許 Grafana 使用者介面從 Amazon S3 儲存貯體載入資源。如果沒有 CORS 組態，由於 Grafana 網域無法存取 Amazon

S3 儲存貯體中的資源，因此您會在場景檢視器上收到錯誤訊息「載入 3D 場景失敗並發生網路故障」。

若要使用 CORS 設定 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. 在儲存貯體清單中，選擇您用作 AWS IoT TwinMaker 工作區資源儲存貯體的儲存貯體名稱。
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 組態，方法是將 Grafana 工作區網域取代*GRAFANA-WORKSPACE-DOMAIN*為您的網域。

> **Note**
>
> 您需要將星號*字元保留在 "AllowedOrigins": JSON 元素的開頭。

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

6. 選取儲存變更以完成 CORS 組態。

如需搭配 Amazon S3 儲存貯體的 CORS 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 Managed Grafana 進行全受管服務，或設定您自己管理的 Grafana 環境。透過 Amazon Managed Grafana，您可以快速部署、操作和擴展開放原始碼 Grafana 以滿足您的需求。或者，您可以設定自己的基礎設施來管理 Grafana 伺服器。

如需 Grafana 環境選項的詳細資訊，請參閱下列主題：

- Amazon Managed Grafana
- 自我管理的 Grafana

Amazon Managed Grafana

Amazon Managed Grafana 提供 AWS IoT TwinMaker 外掛程式，讓您可以快速 AWS IoT TwinMaker 與 Grafana 整合。由於 Amazon Managed Grafana 會為您管理 Grafana 伺服器，因此您可以視覺化資料，而無需建置、封裝或部署任何硬體或任何其他 Grafana 基礎設施。如需 Amazon Managed Grafana 的詳細資訊，請參閱什麼是 Amazon Managed Grafana？。

> **Note**
>
> Amazon Managed Grafana 目前支援 Grafana 外掛程式的 1.3.1 AWS IoT TwinMaker 版。

Amazon Managed Grafana 先決條件

若要 AWS IoT TwinMaker 在 Amazon Managed 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 Managed Grafana 工作區時， AWS IoT TwinMaker 不會列出 。不過，外掛程式已安裝在所有工作區。您可以在開放原始碼 Grafana

AWS IoT TwinMaker 外掛程式清單中找到外掛程式。您可以在資料來源頁面上選擇新增資料來源來尋找 AWS IoT TwinMaker 資料來源。

當您建立 Amazon Managed Grafana 工作區時，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。這是您將用來設定 Grafana 所有 AWS IoT TwinMaker 資料來源的身分驗證提供者選項。Amazon Managed Grafana 不支援自動新增 的許可 AWS IoT TwinMaker，因此您必須手動設定這些許可。如需設定手動許可的詳細資訊，請參閱 建立儀表板 IAM 角色。

自我管理的 Grafana

您可以選擇託管自己的基礎設施來執行 Grafana。如需有關在機器本機執行 Grafana 的資訊，請參閱安裝 Grafana。 AWS IoT TwinMaker 外掛程式可在公有 Grafana 目錄中取得。如需在 Grafana 環境中安裝此外掛程式的詳細資訊，請參閱AWS IoT TwinMaker 應用程式。

當您在本機執行 Grafana 時，您無法輕鬆共用儀表板或提供存取權給多個使用者。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。此資源會逐步引導您在 Cloud9 上託管 Grafana 環境，並在公有端點上託管 Amazon EC2。

您必須決定要使用哪個身分驗證提供者來設定 TwinMaker 資料來源。您可以根據預設登入資料鏈結來設定環境的登入資料（請參閱使用預設登入資料提供者鏈結)。預設登入資料可以是任何使用者或角色的永久登入資料。例如，如果您在 Amazon EC2 上執行 Grafana，則預設憑證鏈可以存取 Amazon EC2 執行角色，這會是您的身分驗證提供者。在 的步驟中，需要身分驗證提供者的 IAM Amazon Resource Name (ARN)建立儀表板 IAM 角色。

建立儀表板 IAM 角色

使用 AWS IoT TwinMaker，您可以在 Grafana 儀表板上控制資料存取。Grafana 儀表板使用者應具有不同的許可範圍來檢視資料，在某些情況下應寫入資料。例如，警示運算子可能沒有檢視影片的許可，而管理員具有所有資源的許可。Grafana 透過資料來源定義許可，其中提供登入資料和 IAM 角色。 AWS IoT TwinMaker 資料來源會擷取具有該角色許可的 AWS 登入資料。如果未提供 IAM 角色，Grafana 會使用登入資料的範圍，這無法由 減少 AWS IoT TwinMaker。

若要在 Grafana 中使用 AWS IoT TwinMaker 儀表板，您可以建立 IAM 角色並連接政策。您可以使用下列範本來協助您建立這些政策。

建立 IAM 政策

在 IAM 主控台*YourWorkspaceId*DashboardPolicy中建立名為 的 IAM 政策。此政策可讓您的工作區存取 Amazon S3 儲存貯體和資源 AWS IoT TwinMaker 。您也可以決定使用 AWS IoT Greengrass Edge Connector for Amazon Kinesis Video Streams，這需要針對元件設定的 Kinesis Video Streams 和 AWS IoT SiteWise 資產的許可。若要符合您的使用案例，請選擇下列其中一個政策範本。

1。沒有影片許可政策

如果您不想使用 Grafana Video Player 面板，請使用下列範本建立政策。

JSON

```
{
  "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": "*"
    }
  ]
```

```
}
```

每個工作區都會建立 Amazon S3 儲存貯體。它包含要在儀表板上檢視的 3D 模型和場景。SceneViewer 面板會從此儲存貯體載入項目。

2. 縮小視訊許可政策的範圍

   若要限制 Grafana 中影片播放器面板的存取，請依標籤將您的 AWS IoT Greengrass Edge Connector for Amazon Kinesis Video Streams 資源分組。如需縮小影片資源許可範圍的詳細資訊，請參閱建立 AWS IoT TwinMaker 影片播放器政策。

3. 所有影片許可

   如果您不想將影片分組，您可以從 Grafana Video Player 存取所有影片。有權存取 Grafana 工作區的任何人都可以播放您帳戶中任何串流的影片，並具有對任何 AWS IoT SiteWise 資產的唯讀存取權。這包括未來建立的任何資源。

   使用下列範本建立政策：

   JSON

```
{
  "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*"
        }
      }
    }
  ]
}
```

此政策範本提供下列許可：

- 唯讀存取 S3 儲存貯體以載入場景。
- 僅讀取工作區中所有實體和元件 AWS IoT TwinMaker 的 存取權。
- 唯讀存取，以串流您帳戶中的所有 Kinesis Video Streams 影片。
- 唯讀存取您帳戶中所有 AWS IoT SiteWise 資產的屬性值歷史記錄。

- 資料擷取至標記 金鑰`EdgeConnectorForKVS`和值 的 AWS IoT SiteWise 資產的任何屬性`workspaceId`。

從邊緣標記您的攝影機 AWS IoT SiteWise 資產請求影片上傳

在 Grafana 中使用影片播放器，使用者可以手動請求將影片從邊緣快取上傳到 Kinesis Video Streams。您可以為與 AWS IoT Greengrass Edge Connector for Amazon Kinesis Video Streams 相關聯的任何 AWS IoT SiteWise 資產，以及以金鑰 標記的任何資產開啟此功能`EdgeConnectorForKVS`。

標籤值可以是以下列任何字元分隔的 workspaceIds 清單：`. : + = @ _ / -`。例如，如果您想要跨 AWS IoT TwinMaker 工作區使用與 AWS IoT Greengrass Edge Connector for Amazon Kinesis Video Streams 相關聯的 AWS IoT SiteWise 資產，您可以使用遵循此模式的標籤：`WorkspaceA/WorkspaceB/WorkspaceC`。Grafana 外掛程式會強制執行 AWS IoT TwinMaker workspaceId 用於分組 AWS IoT SiteWise 資產資料擷取。

將更多許可新增至儀表板政策

AWS IoT TwinMaker Grafana 外掛程式會使用身分驗證提供者，在您建立的儀表板角色上呼叫 AssumeRole。在內部，外掛程式會使用 AssumeRole 呼叫中的工作階段政策來限制您有權存取的最高許可範圍。如需工作階段政策的詳細資訊，請參閱工作階段政策。

這是您在 AWS IoT TwinMaker 工作區的儀表板角色上可以擁有的最大允許政策：

JSON

```
{
  "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*"
        }
      }
    }
  ]
```

```
}
```

如果您新增Allow更多許可的陳述式，則無法在 AWS IoT TwinMaker 外掛程式上運作。這是設計來確保外掛程式使用最低必要許可。

不過，您可以進一步縮小許可範圍。如需相關資訊，請參閱建立 AWS IoT TwinMaker 影片播放器政策。

建立 Grafana Dashboard IAM 角色

在 IAM 主控台中，建立名為 的 IAM 角色*YourWorkspaceId*DashboardRole。將*YourWorkspaceId*DashboardPolicy連接至角色。

若要編輯儀表板角色的信任政策，您必須授予 Grafana 身分驗證提供者在儀表板角色AssumeRole上呼叫 的許可。使用下列範本更新信任政策：

JSON

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

如需建立 Grafana 環境和尋找身分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 影片播放器政策

以下是政策範本，其中包含 Grafana 中 AWS IoT TwinMaker 外掛程式所需的所有影片許可：

JSON

```
{
  "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*"
        }
      }
    }
  ]
}
```

如需完整政策的詳細資訊，請參閱 建立 IAM 政策主題中的所有影片許可政策範本。

縮小對 資源的存取範圍

Grafana 中的影片播放器面板會直接呼叫 Kinesis Video Streams 和 IoT SiteWise，以提供完整的影片播放體驗。若要避免未經授權存取與您的 AWS IoT TwinMaker 工作區無關的資源，請將條件新增至工作區儀表板角色的 IAM 政策。

縮小 GET 許可的範圍

您可以透過標記資源來縮小 Amazon Kinesis Video Streams 和 AWS IoT SiteWise 資產的存取範圍。您可能已經根據 workspaceId AWS IoT TwinMaker 標記 AWS IoT SiteWise 相機資產，以啟用影片上傳請求功能，請參閱從邊緣主題上傳影片。您可以使用相同的標籤鍵/值對來限制對 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，您可以將此條件新增至 中的 kinesisvideo 和 iotsitewise 陳述式*YourWorkspaceId*DashboardPolicy：

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

```
  }
}
```

實際使用案例：將攝影機分組

在此案例中，您有大量攝影機監控在工廠中製作 Cookie 的程序。Cookie batter 的批次是在 Batter Room 中製作，Batter 在 Freezer Room 中凍結，而 Cookie 在 Baking Room 中製作。其中每個房間都有攝影機，而不同的運算子團隊會分別監控每個程序。您希望每個運算子群組都獲得各自房間的授權。為 Cookie 工廠建置數位分身時，會使用單一工作區，但攝影機許可需要依房間範圍。

您可以根據攝影機群組 ID groupingId 標記攝影機群組，以達成此許可分離。在此案例中，groupingIds 是 BatterRoom、FreezerRoom 和 BakingRoom。每個房間中的攝影機都連接到 Kinesis Video Streams，並且應該有一個標籤：Key = `EdgeConnectorForKVS`，Value = `BatterRoom`。此值可以是以下列任何字元分隔的分組清單：`. : + = @ _ / -`

若要修改 *YourWorkspaceId*`DashboardPolicy`，請使用下列政策陳述式：

```
...,
{
  "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影片播放和 AWS IoT SiteWise 屬性歷史記錄對群組中特定資源的存取。是由您的使用案例*groupingId*所定義。在我們的案例中，它是 roomId。

Scope down AWS IoT SiteWise BatchPutAssetPropertyValue 許可

提供此許可會開啟影片播放器中的影片上傳請求功能。當您上傳影片時，您可以在 Grafana 儀表板的面板上選擇提交，以指定時間範圍並從 提交請求。

若要提供 iotsitewise：BatchPutAssetPropertyValue 許可，請使用預設政策：

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

透過使用此政策，使用者可以針對 AWS IoT SiteWise 攝影機資產上的任何屬性呼叫 BatchPutAssetPropertyValue。您可以在陳述式條件中指定特定 propertyId 來限制授權。

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

Grafana 中的影片播放器面板會將資料擷取至名為 VideoUploadRequest 的測量屬性，以開始將影片從邊緣快取上傳至 Kinesis Video Streams。在 AWS IoT SiteWise 主控台中尋找此屬性的 propertyId。若要修改 *YourWorkspaceId*DashboardPolicy，請使用下列政策陳述式：

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

此陳述式會將擷取資料限制為已標記 AWS IoT SiteWise 攝影機資產的特定屬性。如需詳細資訊，請參閱 AWS IoT SiteWise 如何使用 IAM。

將 AWS IoT SiteWise 警示連接至 AWS IoT TwinMaker Grafana 儀表板

> **Note**
>
> 此功能為公開預覽版，可能會有所變更。

AWS IoT TwinMaker 能夠將 AWS IoT SiteWise 和 事件警示匯入 AWS IoT TwinMaker 元件。這可讓您查詢警示狀態並設定警示閾值，而無需實作自訂資料連接器以進行 AWS IoT SiteWise 資料遷移。您可以使用 AWS IoT TwinMaker Grafana 外掛程式將警示狀態視覺化，並在 Grafana 中設定警示閾值，而無需對警示進行 API 呼叫 AWS IoT TwinMaker 或直接與 AWS IoT SiteWise 警示互動。

> **Note**
>
> 終止支援通知： 將於 2026 AWS 年 5 月 20 日結束對 的支援 AWS IoT Events。2026 年 5 月 20 日之後，您將無法再存取 AWS IoT Events 主控台或 AWS IoT Events 資源。如需詳細資訊，請參閱AWS IoT Events 終止支援。

AWS IoT SiteWise 警示組態先決條件

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

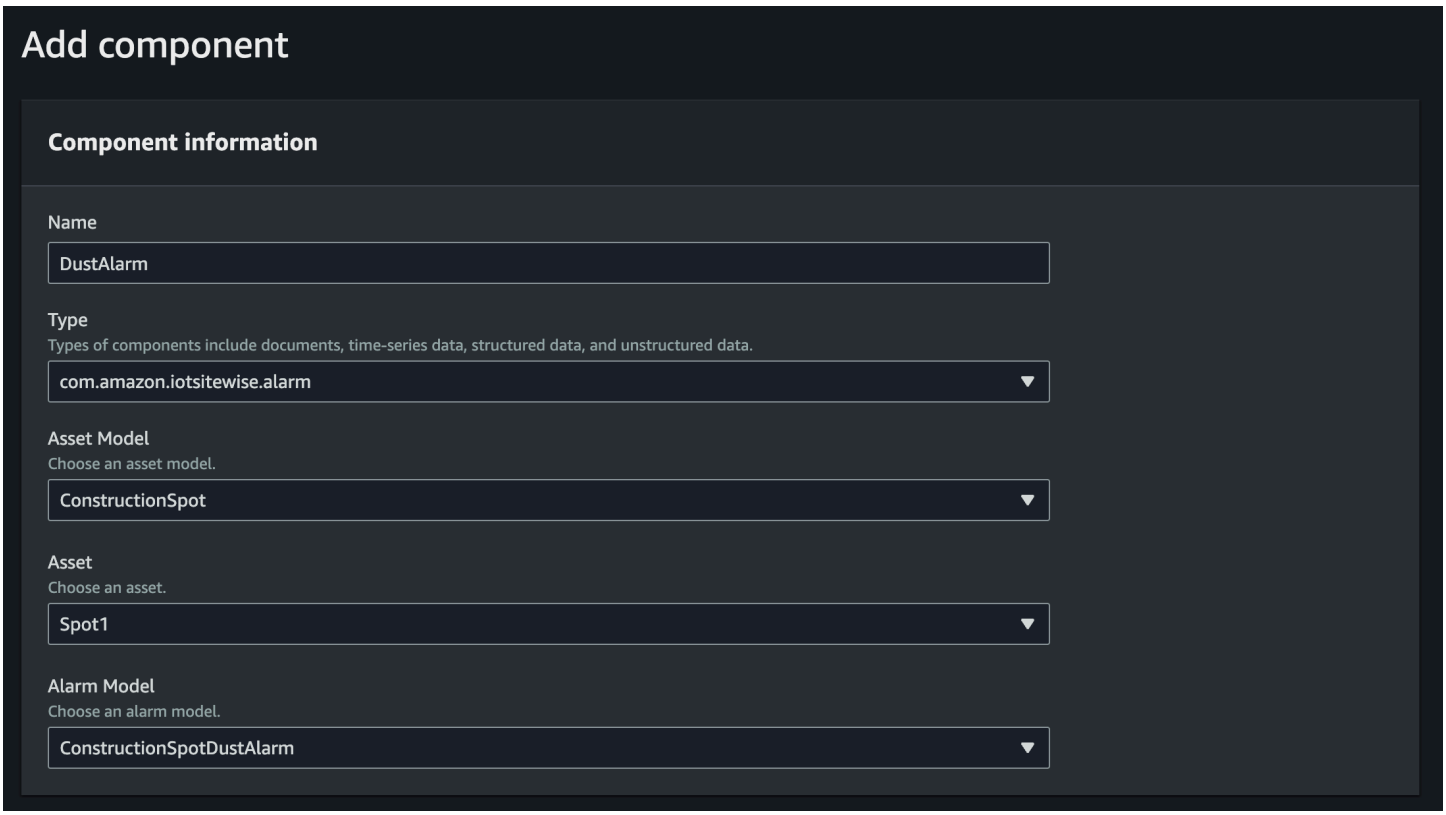
- 熟悉 AWS IoT SiteWise模型和資產系統。如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的建立資產模型和建立資產。
- 熟悉 IoT Events 警示模型，以及如何將它們連接到 AWS IoT SiteWise 模型。如需詳細資訊，請參閱 AWS IoT SiteWise 使用者指南中的定義 AWS IoT 事件警示。
- AWS IoT TwinMaker 與 Grafana 整合，讓您可以存取 Grafana 中的 AWS IoT TwinMaker 資源。如需詳細資訊，請參閱 AWS IoT TwinMaker Grafana 儀表板整合。

定義 AWS IoT SiteWise 警示元件 IAM 角色

AWS IoT TwinMaker 使用工作區 IAM 角色在 Grafana 中查詢和設定警示閾值。 AWS IoT TwinMaker 工作區角色需要下列許可，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 AWS IoT TwinMaker 主控台中，建立代表您 AWS IoT SiteWise 資產的實體。請務必使用 新增該實體的元件`com.amazon.iotsitewise.alarm`做為元件類型，並挑選對應的資產和警示模型。



上述螢幕擷取畫面是使用 類型建立此實體的範例`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。您可以重複此警示元件類型模式，為工作區中所需的所有資產建立警示元件。

透過 AWS IoT TwinMaker API 查詢和更新

建立警示元件之後，您可以透過 AWS IoT TwinMaker API 查詢警示狀態、閾值和更新警示閾值。

以下是查詢警示狀態的範例請求：

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

以下是查詢警示閾值的範例請求。

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

以下是更新警示閾值的範例請求：

```
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}"
                }
            ]
        }
    ]
}'
```

設定警示的 Grafana 儀表板

需要建立第二個啟用寫入的儀表板 IAM 角色 ，這是正常角色，但具有`iottwinmaker:BatchPutPropertyValues`將 動作新增至 TwinMaker 工作區的許可，如以下範例所示。

JSON

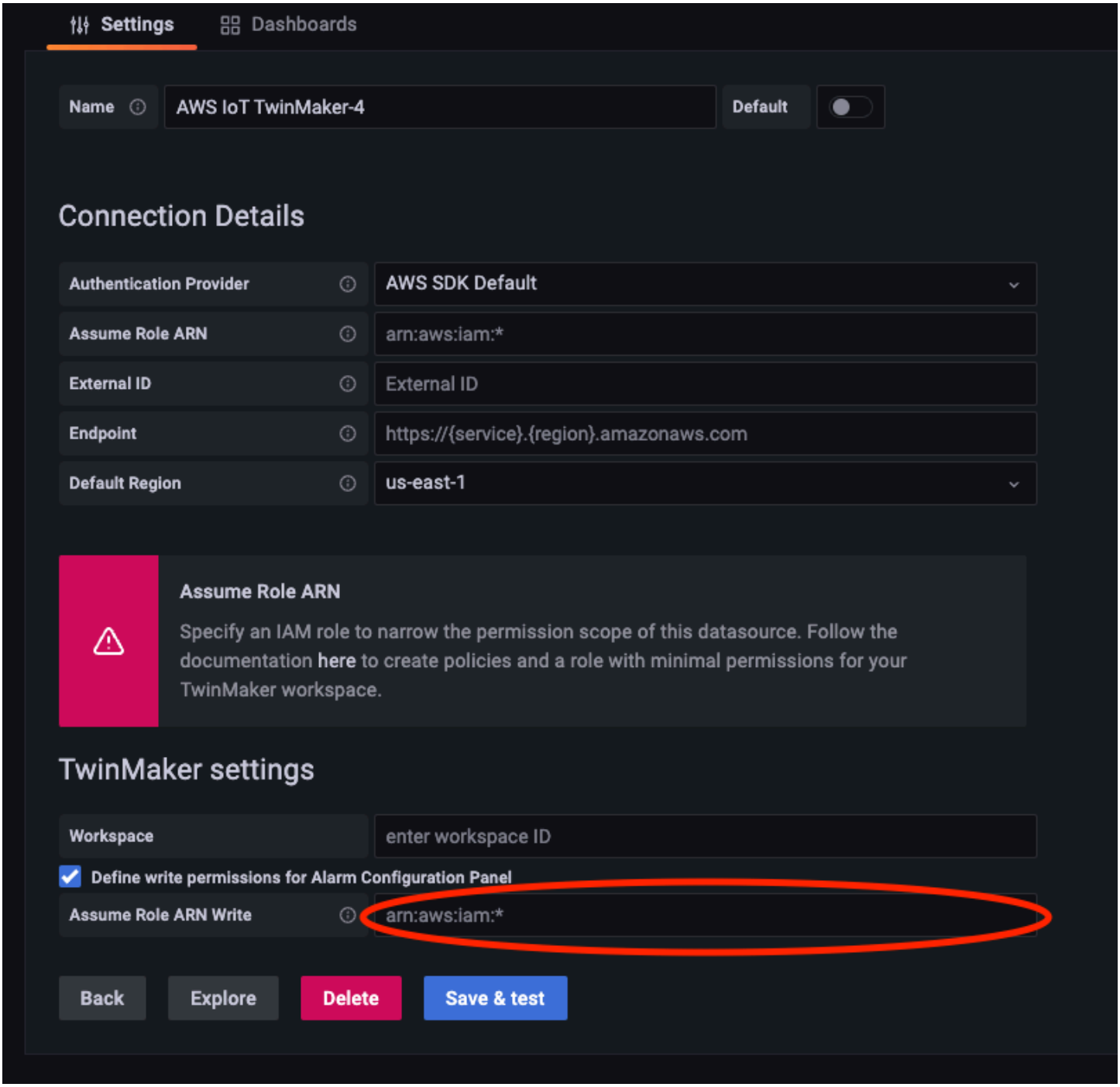
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iottwinmaker:Get*",
               "iottwinmaker:List*",
                "iottwinmaker:BatchPutPropertyValues"
            ],
            "Resource": [
            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

或者，您可以在 IAM 角色的結尾新增此陳述式：

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

資料來源需要使用您建立的儀表板寫入角色來設定其寫入 arn。

修改 IAM 角色後，請登入 Grafana 儀表板以擔任更新的角色。選取定義警示組態面板寫入許可的核取方塊，並在 arn 中複製寫入角色。



使用 Grafana 儀表板進行警示視覺化

使用下列程序將警示組態面板新增至儀表板並進行設定：：

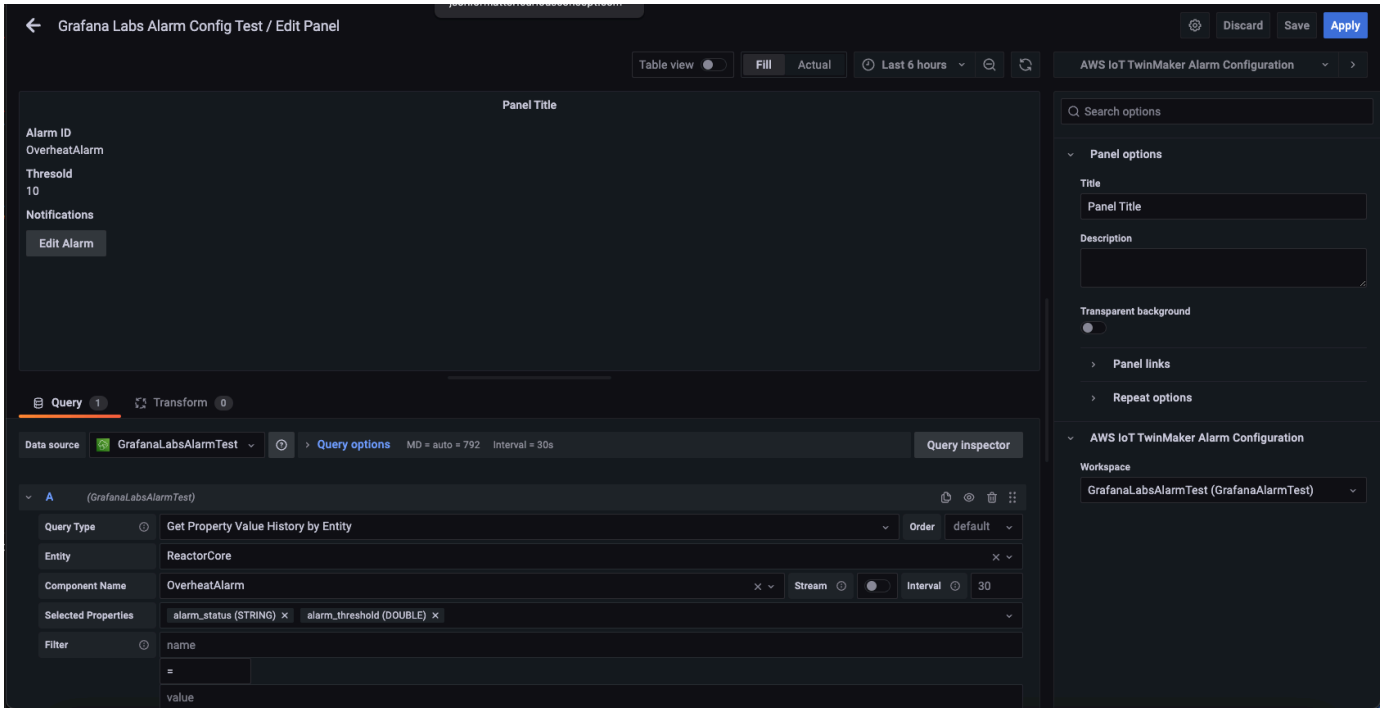
1. 在面板選項中選取工作區。
2. 在查詢組態中設定您的資料來源。

3. 使用下列查詢類型：Get Property Value History by Entity。
4. 選取您要新增警示的實體或實體變數。
5. 選取實體之後，請選取要套用屬性的元件或元件變數。
6. 針對 屬性，選擇： alarm_status和 alarm_threshold。

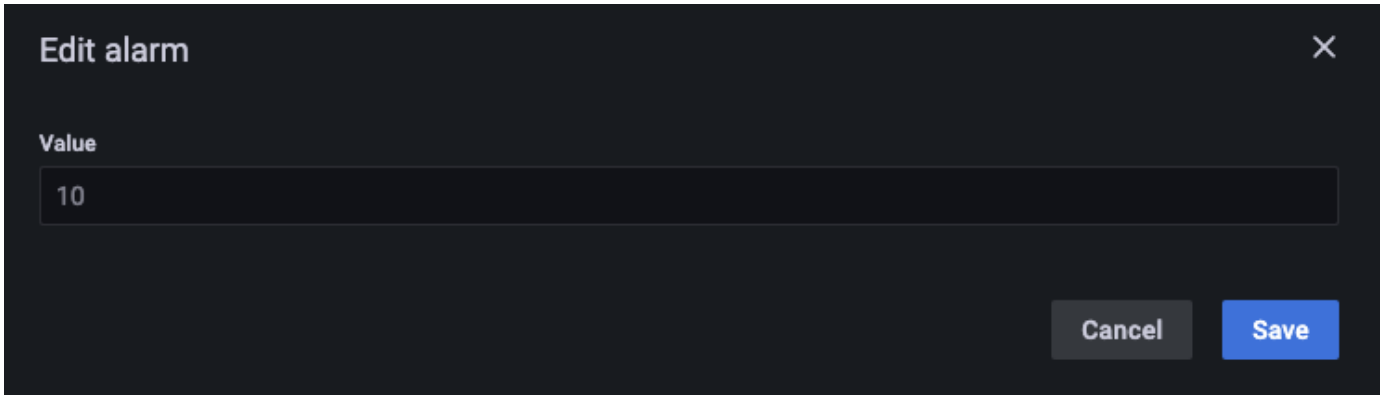
   連線後，您應該會看到警示 ID 的 ID 及其目前的閾值。

> **Note**
>
> 對於公開預覽，不會顯示通知。您應該檢閱警示狀態和閾值，以確保屬性已正確套用。

7. 應使用預設的遞增查詢順序，以便顯示最新的值。
8. 查詢的篩選條件區段可以保留空白。完整的組態如下所示：



9. 透過使用編輯警示按鈕，您可以開啟對話方塊來變更目前的警示閾值。
10. 選取儲存以設定新的閾值。



> **Note**
>
> 此面板只能與包含目前的即時時間範圍搭配使用。搭配過去結束和開始的時間範圍使用它，可能會在一律將警示閾值編輯為目前閾值時顯示非預期的值。

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

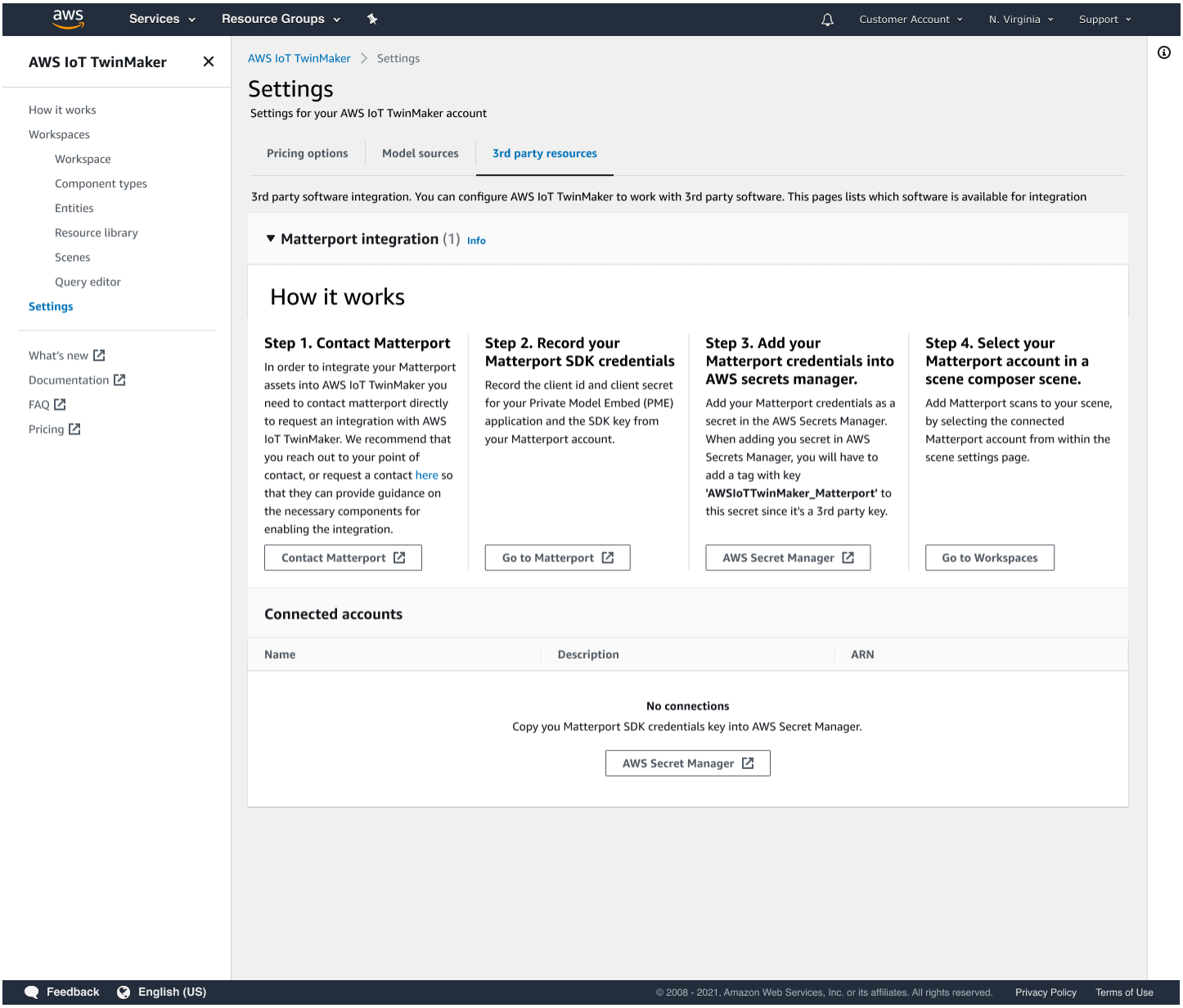
此整合可讓您執行下列作業：

- 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，在整合所需的不同資源之間瀏覽。



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。

6. 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。
7. 接下來，找到「應用程序集成管理」部分，您應該會看到列出您的 PME 應用程序。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 M atterport 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

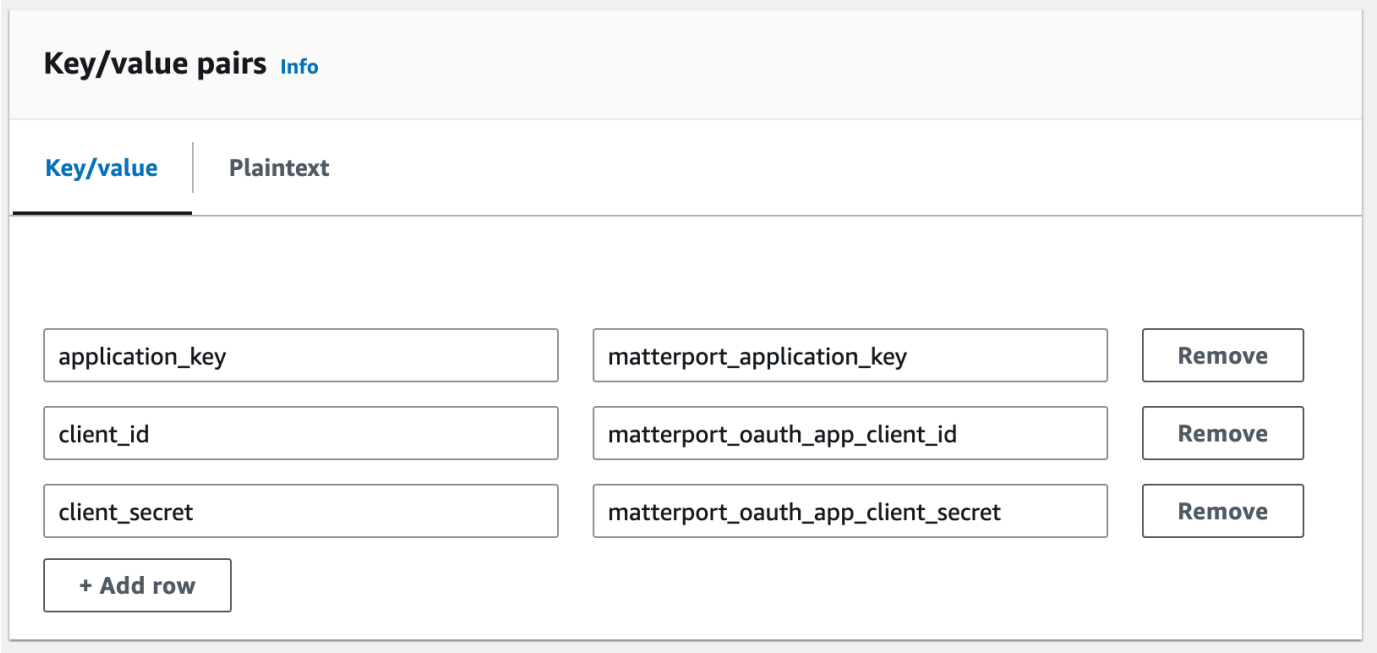
> **Note**
>
> 您需要從產生並記錄您的 Matterport 認證主題中的程序建立的用戶端密碼，才能繼續進行 Matterport 整合。

1. 登入 AWS Secrets Manager 主控台。
2. 導覽至「密碼」頁面，然後選取「儲存新密碼」。
3. 對於「密碼」類型，選取「其他類型的密碼」。
4. 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：。

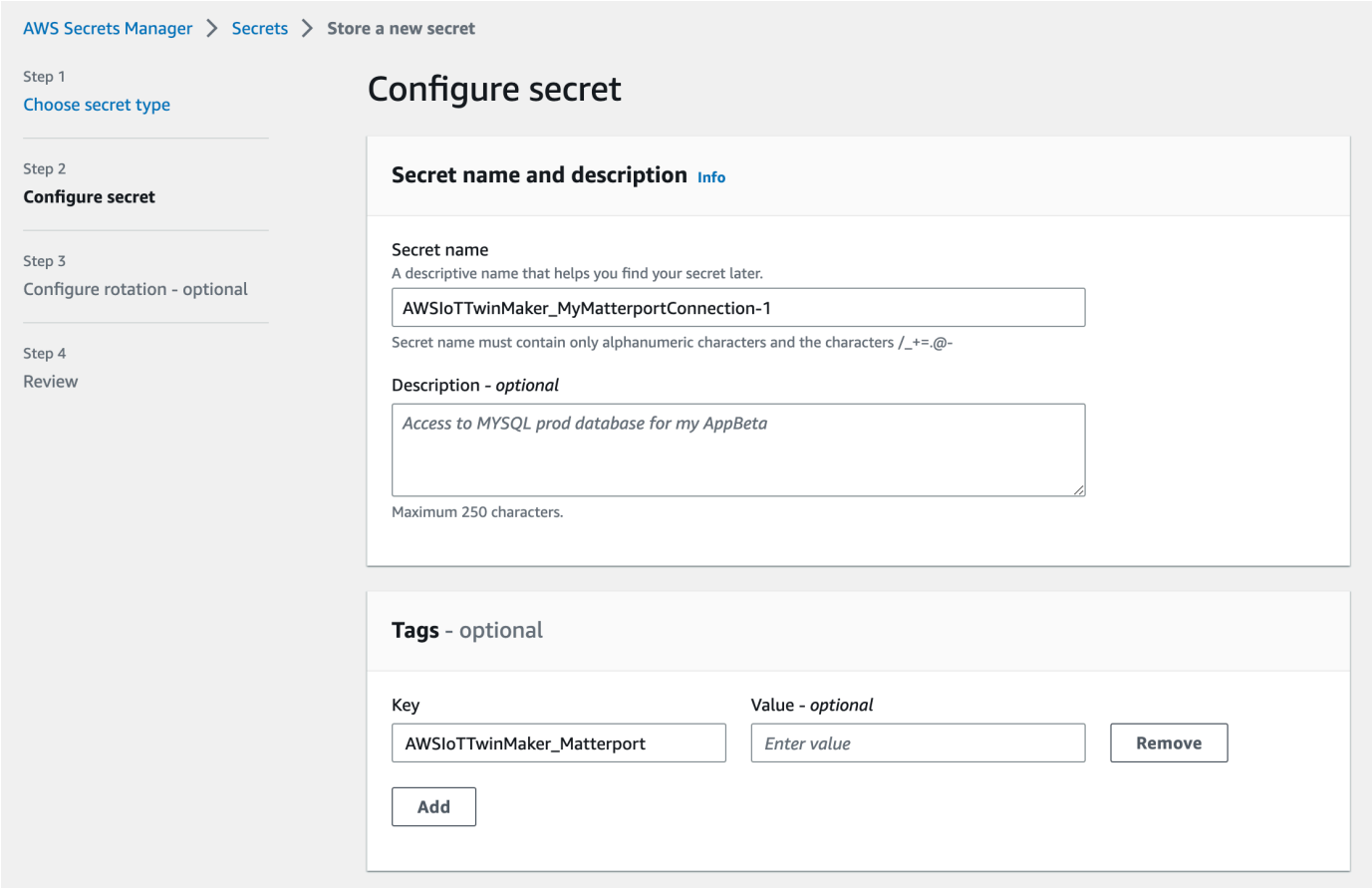
- <your Matterport credentials>創建一個鍵-值對，鍵：client_id和值：。
- <your Matterport credentials>創建一個鍵-值對，鍵：client_secret和值：。

完成後，您應該擁有類似下列範例的組態：



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



> **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。

9. 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。
10. 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？

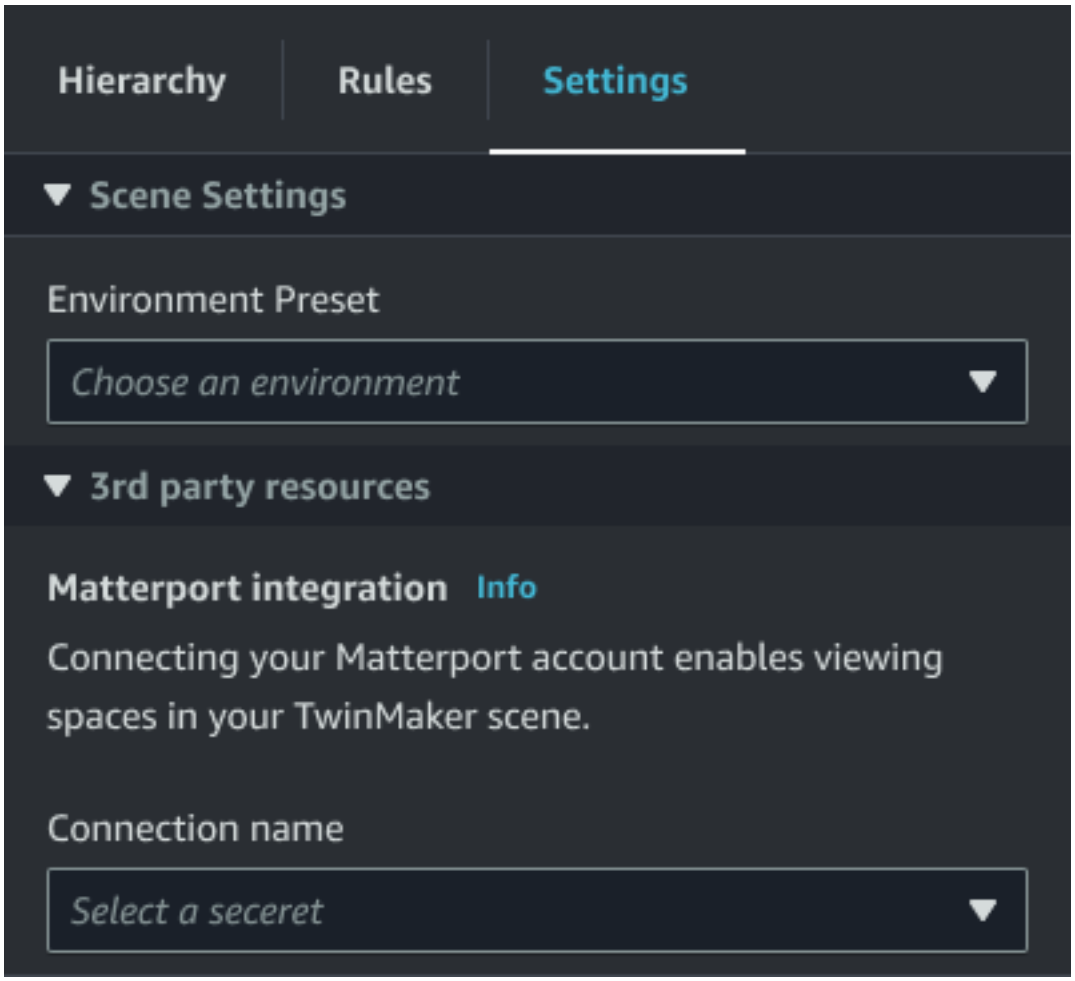
- 旋轉 AWS Secrets Manager 秘密

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

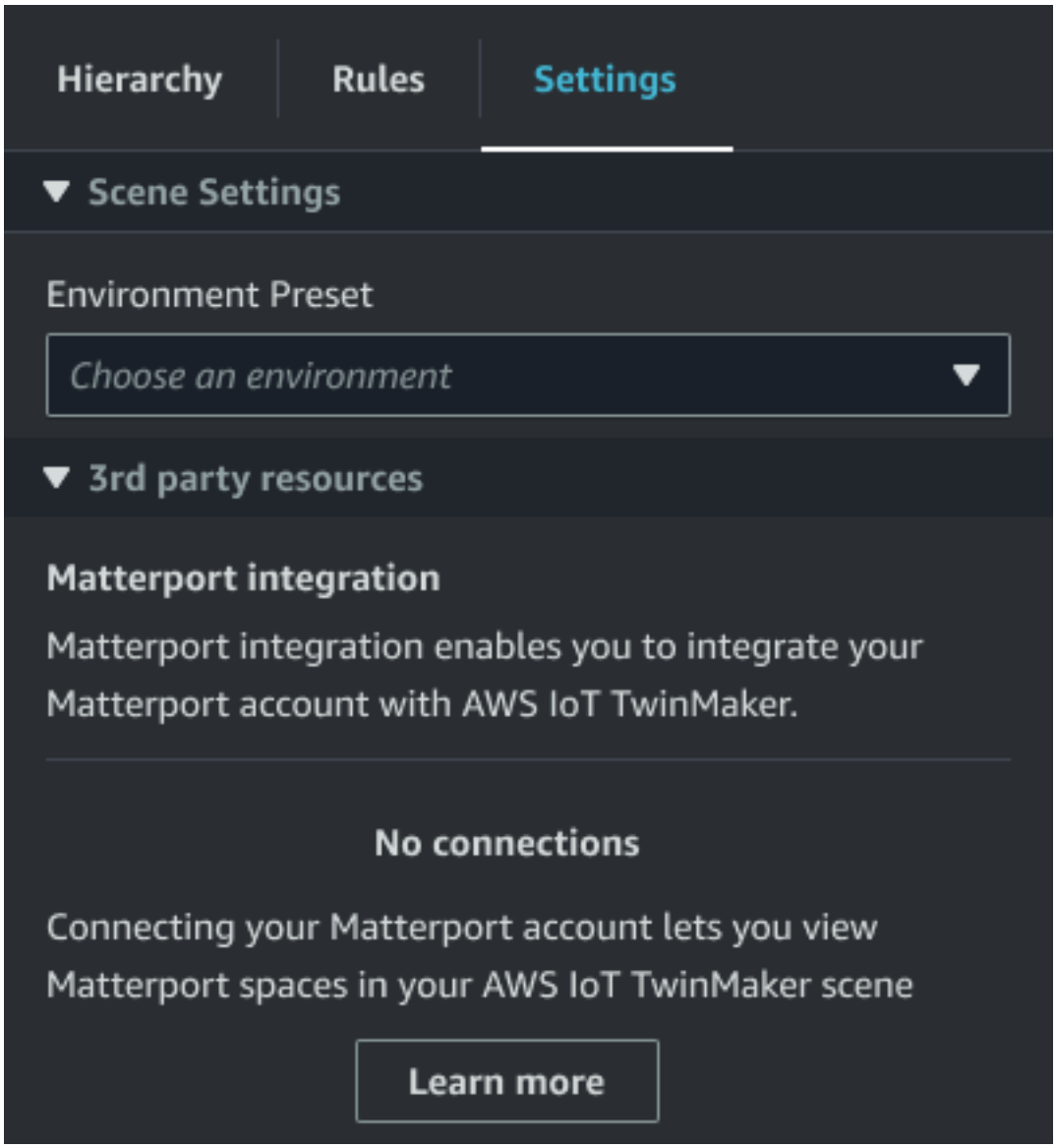
將物件連接埠空間匯入場景 AWS IoT TwinMaker

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：

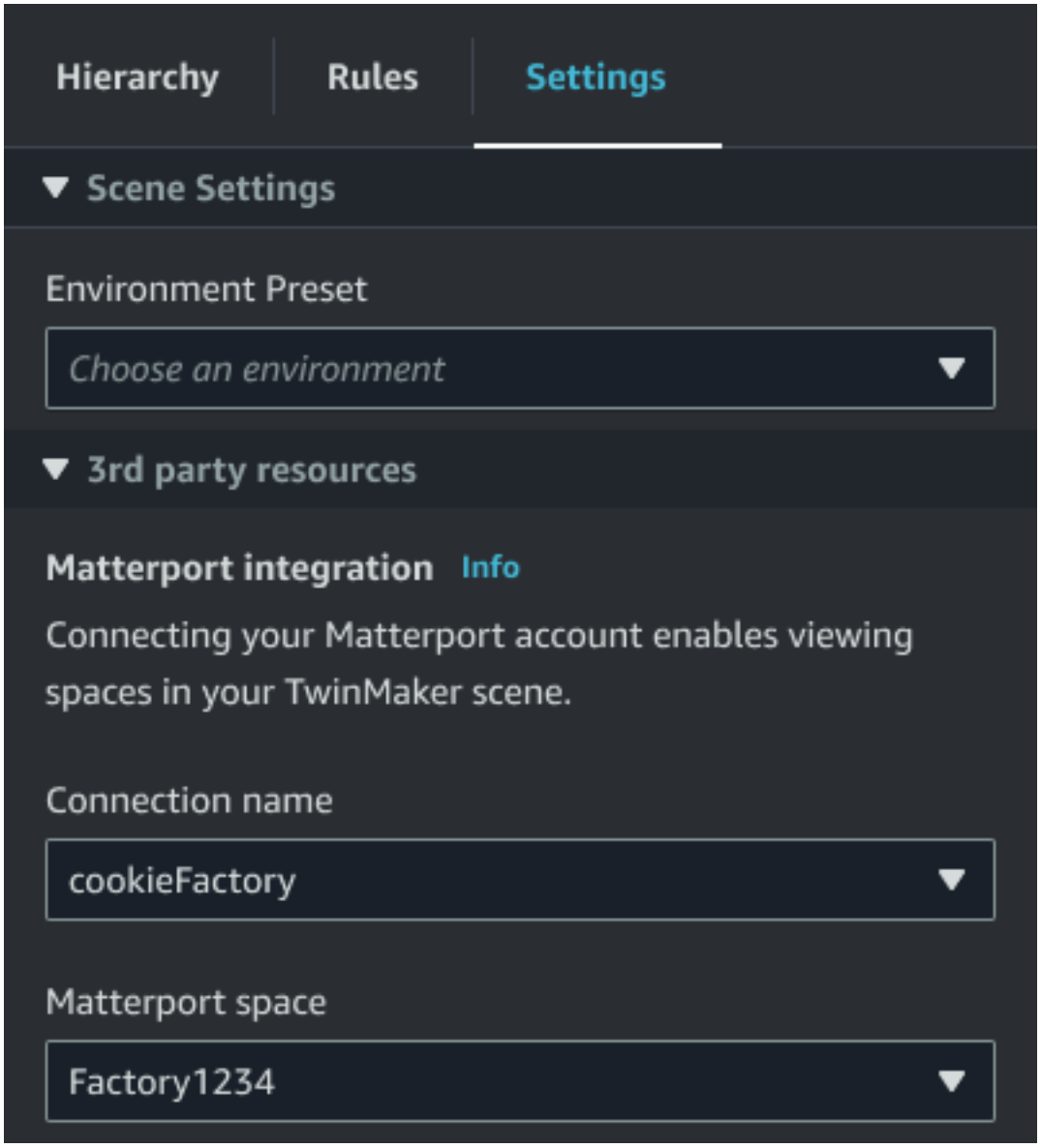
1. 登入 AWS IoT TwinMaker 主控台。
2. 建立或開啟您要在其中使用 Matterport 空間的現有 AWS IoT TwinMaker 場景。
3. 場景開啟後，導覽至「設定」標籤。
4. 在 [設定] 的 [協力廠商資源] 下，找到 [連線名稱]，然後輸入您在程序中建立的密碼將您的馬特港認證存儲在 AWS Secrets Manager。



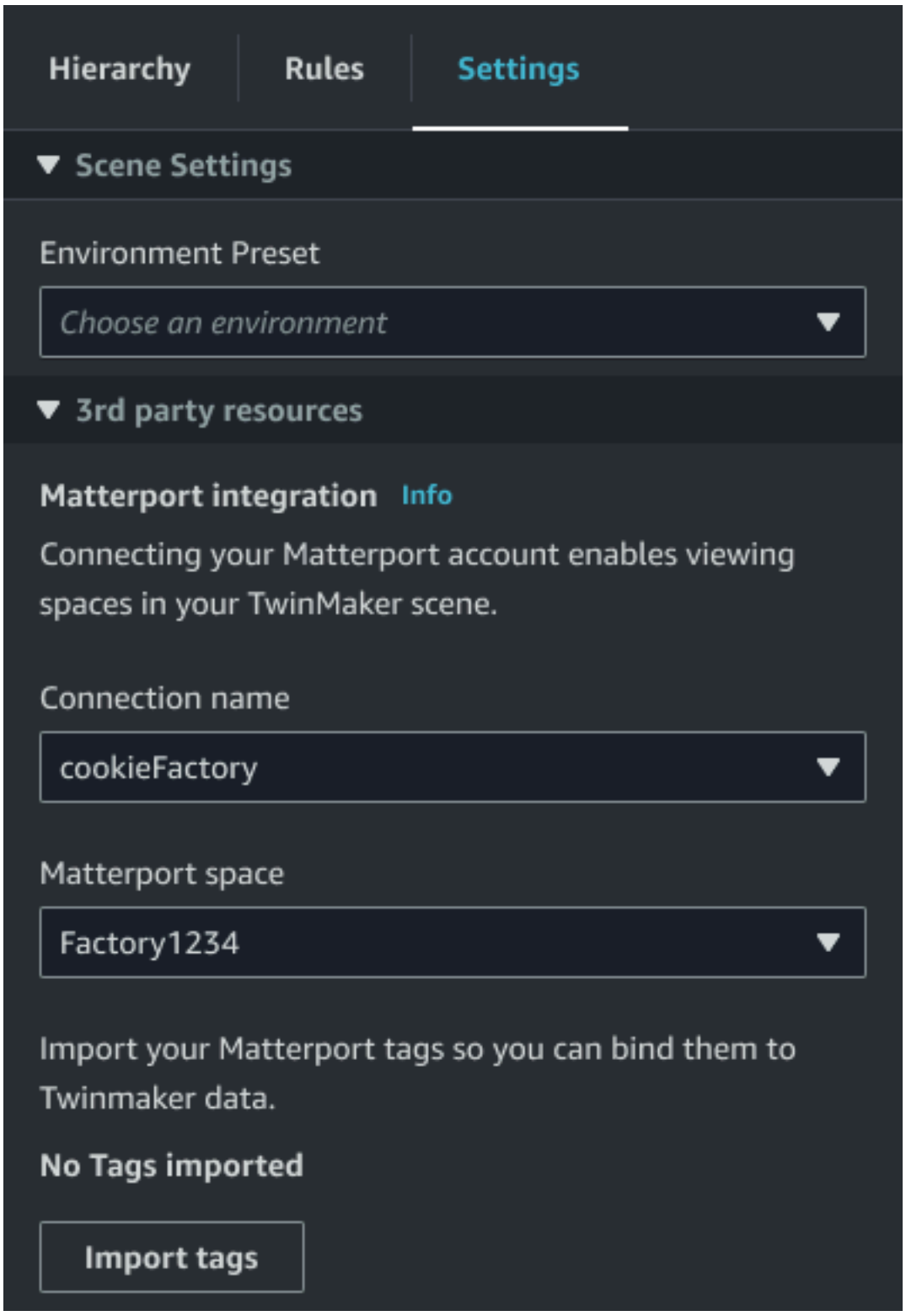
> **Note**
>
> 如果您看到指出「沒有連線」的訊息，請瀏覽至AWS IoT TwinMaker 主控台設定頁面以開始 Matterport 整合程序。



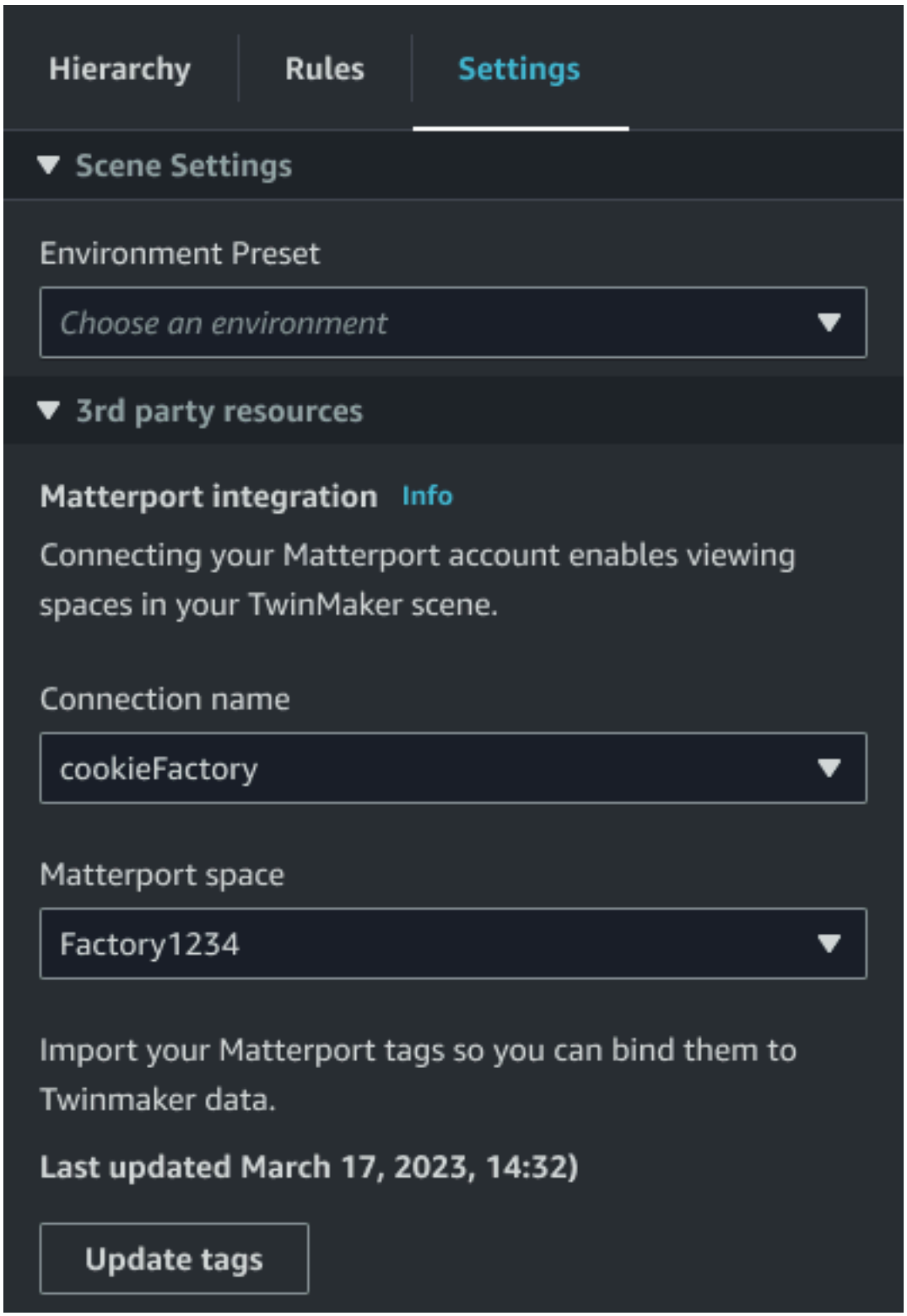
5. 接下來，在 Matterport 空間下拉式清單中選擇您想要在場景中使用的 Matter port 空間。



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 應用程序。
- 要了解有關使用 AWS IoT 應用程序包的更多信息，請訪問AWS IoT 應用程序包 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 影片整合

攝影機是數位孿生模擬的好機會。您可以使用 AWS IoT TwinMaker 來模擬相機的位置和實體條件。在中 AWS IoT TwinMaker 為您的現場攝影機建立實體，並使用影片元件將即時影片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 可以透過兩種方式從邊緣裝置擷取影片。您可以使用 Kinesis 影片串流的邊緣連接器從邊緣裝置串流影片，也可以在邊緣裝置上儲存影片，並使用 MQTT 訊息啟動影片上傳。使用此元件可從您的裝置串流影片資料，以搭配 AWS IoT 服務使用。若要產生所需的資源並部署 Kinesis Video Streams 的邊緣連接器，請參閱 GitHub 上 Kinesis Video Streams 的邊緣連接器入門。如需 AWS IoT Greengrass 元件的詳細資訊，請參閱 Kinesis Video Streams 邊緣連接器上的 AWS IoT Greengrass 文件。

建立所需的 AWS IoT SiteWise 模型並設定 Kinesis Video Streams Greengrass 元件後，您可以在 AWS IoT TwinMaker 主控台中將邊緣的影片串流或錄製到數位分身應用程式。您也可以在 Grafana 儀表板中檢視裝置的即時串流和中繼資料。如需整合 Grafana 和 的詳細資訊 AWS IoT TwinMaker，請參閱 AWS IoT TwinMaker Grafana 儀表板整合。

使用 Kinesis 影片串流的邊緣連接器在 中串流影片 AWS IoT TwinMaker

使用 Kinesis 影片串流的邊緣連接器，您可以將影片和資料串流到 AWS IoT TwinMaker 場景中的實體。您可以使用影片元件來執行此操作。若要建立影片元件以在場景中使用，請完成下列程序。

先決條件

在 AWS IoT TwinMaker 場景中建立影片元件之前，請確定您已完成下列先決條件。

- 為 Kinesis 影片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 影片串流的邊緣連接器入門。
- 已在您的裝置上部署 Kinesis 影片串流邊緣連接器 AWS IoT Greengrass 。如需部署 Kinesis 影片串流邊緣連接器元件的詳細資訊，請參閱部署 README。

建立 AWS IoT TwinMaker 場景的影片元件

完成下列步驟，為您的場景建立 Kinesis 影片串流元件的邊緣連接器。

1. 在 AWS IoT TwinMaker 主控台中，開啟您要新增視訊元件的場景。
2. 場景開啟後，選擇現有實體或建立您要新增元件的實體，然後選擇新增元件。
3. 在新增元件窗格中，輸入元件的名稱，然後在類型中選擇 com.amazon.iotsitewise.connector.edgevideo。
4. 選取您建立的 AWS IoT SiteWise 攝影機模型名稱，以選擇資產模型。此名稱應該具有下列格式：`EdgeConnectorForKVSCameraModel-0abc`，其中結尾的字母和數字字串與您自己的資產名稱相符。
5. 針對資產，選擇您要從中串流視訊的 AWS IoT SiteWise 攝影機資產。此時會出現一個小視窗，顯示目前視訊串流的預覽。

> **Note**
>
> 若要測試您的影片串流，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，查看播放器中出現的影片。

6. 若要將影片元件新增至您的實體，請選擇新增元件。

將影片和中繼資料從 Kinesis 影片串流新增至 Grafana 儀表板

在 AWS IoT TwinMaker 場景中為實體建立影片元件之後，您可以在 Grafana 中設定影片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 TwinMaker Grafana 儀表板整合。

> **Important**
>
> 若要在 Grafana 儀表板中檢視影片，您必須確定 Grafana 資料來源具有適當的 IAM 許可。若要建立所需的角色和政策，請參閱 建立儀表板 IAM 角色。

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。

> **Note**
>
> 對於 Grafana v10.4， AWS IoT TwinMaker 影片播放器位於 Widget 下。選取新增 >> 小工具。

3. 從面板清單中，選擇AWS IoT TwinMaker 影片播放器面板。
4. 在AWS IoT TwinMaker 影片播放器面板中，使用您要串流影片的 Kinesis 影片串流名稱，輸入 KinesisVideoStreamName 的串流名稱。

> **Note**
>
> 若要將中繼資料串流到 Grafana 影片面板，您必須先使用影片串流元件建立實體。

5. 選用：若要將中繼資料從 AWS IoT SiteWise 資產串流到影片播放器，請在實體中選擇您在 AWS IoT TwinMaker 場景中建立的 AWS IoT TwinMaker 實體。針對元件名稱，選擇您為場景中的 AWS IoT TwinMaker 實體建立的視訊元件。

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 F link 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閘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 CloudWatch

AWS IoT TwinMaker將下列各節中列出的量度和維度發佈至`AWS/IoTTwinMaker`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	此測量結果報告建立元件類型是否成功。 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。 指標值可以是0成功或1失敗。 尺寸： ComponentTypeId、 WorkspaceId。 單位：計數
ComponentTypeUpdateFailure	此測量結果報告元件類型更新是否成功。 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。 指標值可以是0成功或1失敗。 尺寸： ComponentTypeId、 WorkspaceId。

指標	描述
	單位：計數
EntityCreationFailure	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。 指標值可以是0成功或1失敗。 尺寸： EntityName、 EntityId、 WorkspaceId。 單位：計數
EntityUpdateFailure	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。 指標值可以是0成功或1失敗。 尺寸： EntityName、 EntityId、 WorkspaceId。 單位：計數
EntityDeletionFailure	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。 指標值可以是0成功或1失敗。 尺寸： EntityName、 EntityId、 WorkspaceId。 單位：計數

> **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 主控台中的 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

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

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

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

本文件可協助您了解如何在使用 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 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS IoT TwinMaker 資源。

主題

- 中的資料保護 AWS IoT TwinMaker
- 的 Identity and Access Management AWS IoT TwinMaker
- AWS IoT TwinMaker 和介面 VPC 端點 (AWS PrivateLink)
- 的合規驗證 AWS IoT TwinMaker
- 中的彈性 AWS IoT TwinMaker
- 中的基礎設施安全 AWS IoT TwinMaker

中的資料保護 AWS IoT TwinMaker

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

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

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

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

靜態加密

AWS IoT TwinMaker 如果您選擇， 會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。服務為您建立的儲存貯體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，我們建議您啟用預設伺服器端加密。如需 Amazon S3 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

傳送到 的所有資料 AWS IoT TwinMaker 都會使用 HTTPS 通訊協定透過 TLS 連線傳送，因此根據預設，它在傳輸時是安全的。

> **Note**
>
> 與 Amazon S3 儲存貯體 AWS IoT TwinMaker 互動時，建議您在 Amazon S3 儲存貯體地址上使用 HTTPS 做為控制，以強制執行傳輸中的加密。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) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行驗證 （ 登入 ） 和授權 （ 具有許可 ） 來使用 AWS IoT TwinMaker 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- AWS IoT TwinMaker 如何使用 IAM
- 的身分型政策範例 AWS IoT TwinMaker
- 對 AWS IoT TwinMaker 身分和存取進行故障診斷
- 使用 AWS IoT TwinMaker的服務連結角色
- AWS 的 受管政策 AWS IoT TwinMaker

目標對象

使用方式 AWS Identity and Access Management (IAM) 會有所不同，具體取決於您在其中執行的工作 AWS IoT TwinMaker。

服務使用者 – 如果您使用 AWS IoT TwinMaker 服務來執行任務，您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS IoT TwinMaker 功能來執行工作時，您可能需要額外的許可。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 對 AWS IoT TwinMaker 身分和存取進行故障診斷。

服務管理員 – 如果您負責公司 AWS IoT TwinMaker 的資源，您可能擁有 的完整存取權 AWS IoT TwinMaker。您的任務是判斷服務使用者應存取哪些 AWS IoT TwinMaker 功能和資源。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。若要進一步了解貴公司如何搭配 使用 IAM AWS IoT TwinMaker，請參閱 AWS IoT TwinMaker 如何使用 IAM。

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。若要檢視您可以在 IAM 中使用的以 AWS IoT TwinMaker 身分為基礎的政策範例，請參閱 的身分型政策範例 AWS IoT TwinMaker。

使用身分驗證

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

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

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

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的 憑證以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的適用於 API 請求的AWS Signature 第 4 版。

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

AWS 帳戶 根使用者

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

聯合身分

最佳實務是， 要求人類使用者，包括需要管理員存取權的使用者，使用 聯合身分提供者 AWS 服務 來使用臨時憑證來存取 。

聯合身分是來自您的企業使用者目錄、Web 身分提供者、 AWS Directory Service、 Identity Center 目錄或任何使用透過身分來源提供的登入資料 AWS 服務 存取的使用者。當聯合身分存取時 AWS 帳戶，它們會擔任 角色，而角色會提供臨時登入資料。

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，也可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群組，以便

在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

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

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

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

IAM 角色

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

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱《IAM 使用者指南》中的為第三方身分提供者 (聯合) 建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以將政策直接連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有在服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。
  - 服務連結角色 – 服務連結角色是一種連結至 的服務角色。 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

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

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

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

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 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。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

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

資源型政策

資源型政策是連接到資源的 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。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

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

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

其他政策類型

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

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政

策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。

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

多種政策類型

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

AWS IoT TwinMaker 如何使用 IAM

在您使用 IAM 管理對 的存取之前 AWS IoT TwinMaker，請先了解可使用哪些 IAM 功能 AWS IoT TwinMaker。

您可以搭配 使用的 IAM 功能 AWS IoT TwinMaker

IAM 功能	AWS IoT TwinMaker 支援
身分型政策	是
資源型政策	否
政策動作	是

IAM 功能	AWS IoT TwinMaker 支援
政策資源	是
政策條件索引鍵	是
ACL	否
ABAC(政策中的標籤)	部分
臨時憑證	是
主體許可	是
服務角色	是
服務連結角色	否

若要全面了解 AWS IoT TwinMaker 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱AWS IAM Identity Center 《 使用者指南》中的與 AWS IAM 搭配使用的 服務。

的身分型政策 AWS IoT TwinMaker

支援身分型政策：是

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

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

的身分型政策範例 AWS IoT TwinMaker

若要檢視 AWS IoT TwinMaker 身分型政策的範例，請參閱 的身分型政策範例 AWS IoT TwinMaker。

中的資源型政策 AWS IoT TwinMaker

支援資源型政策：否

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

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

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。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS IoT TwinMaker 動作清單，請參閱《服務授權參考》中的 定義的動作 AWS IoT TwinMaker。

中的政策動作在動作之前 AWS IoT TwinMaker 使用以下字首：

```
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 AWS IoT TwinMaker 身分型政策的範例，請參閱 的身分型政策範例 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 ARNs，請參閱《服務授權參考》中的 定義的資源 AWS IoT TwinMaker。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 AWS IoT TwinMaker 身分型政策的範例，請參閱 的身分型政策範例 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內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

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

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

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

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

若要查看 AWS IoT TwinMaker 條件索引鍵的清單，請參閱《服務授權參考》中的 的條件索引鍵 AWS IoT TwinMaker。若要了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 AWS IoT TwinMaker 身分型政策的範例，請參閱 的身分型政策範例 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。

使用 的屬性型存取控制 (ABAC) 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。您可以將標籤連接到 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。

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

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

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

搭配 使用暫時登入資料 AWS IoT TwinMaker

支援臨時憑證：是

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

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

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

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

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。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。

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

的服務連結角色 AWS IoT TwinMaker

支援服務連結角色：否

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

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

的身分型政策範例 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 來執行任務。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。

如需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 (主控台)。

如需 定義的動作和資源類型的詳細資訊 AWS IoT TwinMaker，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的適用於 的動作、資源和條件索引鍵 AWS IoT TwinMaker。

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

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 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 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。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。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。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。如果透過特定 等使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 AWS CloudFormation。如需詳細資訊，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，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_configure-api-require.html中的透過 MFA 的安全 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，您必須擁有一組最低許可。這些許可必須允許您列出和檢視中 AWS IoT TwinMaker 資源的詳細資訊 AWS 帳戶。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。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。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。

為了確保使用者和角色仍然可以使用 AWS IoT TwinMaker 主控台，請將 AWS IoT TwinMaker `ConsoleAccess`或 `ReadOnly` AWS 受管政策連接到實體。如需詳細資訊，請參閱AWS IAM Identity Center 《 使用者指南》中的新增許可給使用者。

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。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "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": "*"
        }
    ]
}
```

對 AWS IoT TwinMaker 身分和存取進行故障診斷

使用以下資訊來協助您診斷和修正使用 AWS IoT TwinMaker 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 中執行動作 AWS IoT TwinMaker
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 以外的人員 AWS 帳戶 存取我的 AWS IoT TwinMaker 資源

我無權在 中執行動作 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 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

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 可讓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 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 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：

- 若要了解 是否 AWS IoT TwinMaker 支援這些功能，請參閱 AWS IoT TwinMaker 如何使用 IAM。
- 若要了解如何提供您擁有 AWS 帳戶 的資源存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 中為 IAM 使用者提供存取權。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。

- 如需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 中的跨帳戶資源存取。

使用 AWS IoT TwinMaker的服務連結角色

AWS IoT TwinMaker use AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 的唯一 IAM 角色類型 AWS IoT TwinMaker。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 AWS 服務所需的所有許可。

服務連結角色可讓您更 AWS IoT TwinMaker 輕鬆地設定，因為您不必手動新增必要的許可。 AWS IoT TwinMaker 會定義其服務連結角色的許可，除非另有定義，否則只能 AWS IoT TwinMaker 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。這可保護您的 AWS IoT TwinMaker 資源，因為您不會不小心移除存取資源的許可。

如需有關支援服務連結角色的其他 服務的資訊，請參閱AWS 使用 IAM 的服務，並在服務連結角色欄中尋找具有是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

的服務連結角色許可 AWS IoT TwinMaker

AWS IoT TwinMaker 使用名為 AWSServiceRoleForIoTTwinMaker 的服務連結角色 – 允許 AWS IoT TwinMaker 呼叫其他 AWS 服務，並代表您同步其資源。

AWSServiceRoleForIoTTwinMaker 服務連結角色信任下列服務擔任該角色：

- `iottwinmaker.amazonaws.com`

名為 AWSIoTTwinMakerServiceRolePolicy 的角色許可政策允許 對指定的資源 AWS IoT TwinMaker 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。

為 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、 AWS CLI或 AWS API 中同步 AWS IoT SiteWise 資產和資產模型（資產同步）時， 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。當您同步 AWS IoT SiteWise 資產和資產模型（資產同步）時， 會再次為您 AWS IoT TwinMaker 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，使用服務名稱建立`iottwinmaker.amazonaws.com`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 的服務連結角色 AWS IoT TwinMaker

AWS IoT TwinMaker 不允許您編輯 AWSServiceRoleForIoTTwinMaker 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」的編輯服務連結角色。

刪除 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，您必須先清除任何仍在使用服務連結角色的 serviceLinked-workspace，才能手動刪除角色。

> **Note**
>
> 如果 AWS IoT TwinMaker 服務在您嘗試刪除資源時使用角色，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 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForIoTTwinMaker 服務連結角色。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。

AWS IoT TwinMaker 服務連結角色支援的 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 的 受管政策 AWS IoT TwinMaker

若要新增許可給使用者、群組和角色，使用 AWS 受管政策比自行撰寫政策更容易。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 受管政策。您無法變更 AWS 受管政策中的許可。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 受管政策中移除許可，因此政策更新不會破壞您現有的許可。

此外， AWS 支援跨多個 服務之任務函數的受管政策。例如，ReadOnlyAccess AWS 受管政策提供所有 AWS 服務和資源的唯讀存取權。當服務啟動新功能時， 會為新操作和資源 AWS 新增唯讀許可。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 受管政策：AWSIoTTwinMakerServiceRolePolicy

您無法將 AWSIoTTwinMakerServiceRolePolicy 連接至 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，請參閱的服務連結角色許可 AWS IoT TwinMaker。

名為 AWSIoTTwinMakerServiceRolePolicy 的角色許可政策允許 對指定的資源 AWS IoT TwinMaker 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

JSON

```
{
    "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"
                    ]
                }
            }
        }
    ]
}
```

AWS IoT TwinMakerAWS 受管政策的更新

檢視自此服務開始追蹤這些變更以來， AWS 受管政策更新的詳細資訊。如需自動收到有關此頁面變更的提醒，請前往 文件歷史記錄頁面上訂閱 RSS 摘要。

變更	描述	日期
AWSIoTTwinMakerServiceRolePolicy – 新增政策	AWS IoT TwinMaker 新增名為 AWSIoTTwinMakerServiceRolePolicy 的角色許可政策，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, iottwinma`	2023 年 11 月 17 日

變更	描述	日期
	ker:ListComponentTypes 如需詳細資訊，請參閱的服務連結角色許可 AWS IoT TwinMaker。	
開始追蹤變更	已開始追蹤其 AWS 受管政策的變更。	2022 年 5 月 11 日

AWS IoT TwinMaker 和介面 VPC 端點 (AWS PrivateLink)

您可以在虛擬私有雲端 (VPC) 與 之間建立私有連線， AWS IoT TwinMaker 方法是建立介面 VPC 端點。介面端點採用 技術AWS PrivateLink，您可以使用 來私下存取 AWS IoT TwinMaker APIs，無需網際網路閘道、網路位址轉譯 (NAT) 裝置、VPN 連線或 AWS Direct Connect 連線。 透過其介面端點 AWS IoT TwinMaker 支援 IPv4 和 IPv6 （雙堆疊）。VPC 中的執行個體不需要公有 IP 地址即可與 AWS IoT TwinMaker APIs通訊。VPC 與 之間的流量 AWS IoT TwinMaker 不會離開 Amazon 網路。

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》中的界面 VPC 端點 (AWS PrivateLink)。

AWS IoT TwinMaker VPC 端點的考量事項

設定介面 VPC 端點之前 AWS IoT TwinMaker，請檢閱《Amazon VPC 使用者指南》中的介面端點屬性和限制。

AWS IoT TwinMaker 支援從您的 VPC 呼叫其所有 API 動作。

- 對於資料平面 API 操作，請使用下列端點：

  ```
  data.iottwinmaker.region.amazonaws.com
  ```

  資料平面 API 操作包括下列項目：

  - GetPropertyValue
  - GetPropertyValueHistory

  - BatchPutPropertyValues
- 對於控制平面 API 操作，請使用下列端點：

```
api.iottwinmaker.region.amazonaws.com
```

  支援的控制平面 API 操作包括下列項目：

  - CreateComponentType
  - CreateEntity
  - CreateScene
  - CreateWorkspace
  - DeleteComponentType
  - DeleteEntity
  - DeleteScene
  - DeleteWorkspace
  - GetComponentType
  - GetEntity
  - GetScene
  - GetWorkspace
  - ListComponentTypes
  - ListComponentTypes
  - ListEntities
  - ListScenes
  - ListTagsForResource
  - ListWorkspaces
  - TagResource
  - UntagResource
  - UpdateComponentType
  - 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 () 來建立 AWS IoT TwinMaker 服務的 VPC 端點AWS CLI。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的建立介面端點。

為 建立使用以下服務名稱 AWS IoT TwinMaker 的 VPC 端點。

- 對於資料平面 API 操作，請使用下列服務名稱：

```
com.amazonaws.region.iottwinmaker.data
```

- 對於控制平面 API 操作，請使用下列服務名稱：

```
com.amazonaws.region.iottwinmaker.api
```

如果您為端點啟用私有 DNS，您可以使用 AWS IoT TwinMaker 區域的預設 DNS 名稱向 提出 API 請求，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。

下列區域支援AWS IoT TwinMaker PrivateLink：

- us-east-1

  下列可用區域支援 ControlPlane 服務：`use1-az1`、 `use1-az2`和 `use1-az6`。

  下列可用區域支援 DataPlane 服務：`use1-az1`、 `use1-az2`和 `use1-az4`。
- us-west-2

  下列可用區域支援 ControlPlane 和 DataPlane 服務：`usw2-az1`、 `usw2-az2`和 `usw2-az3`。
- 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 資源 - AWS Resource Access Manager 的可用區域 IDs。

AWS IoT TwinMaker 透過界面 VPC 端點存取

當您建立界面端點時， 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。預設會啟用私有 DNS 選項。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，您可以透過下列其中一個 VPC 端點向 AWS IoT TwinMaker 提出 API 請求。

- 對於資料平面 API 操作，請使用下列端點。將 AWS *##*取代為您的區域。

```
data.iottwinmaker.region.amazonaws.com
```

- 對於控制平面 API 操作，請使用下列端點。將 AWS *##*取代為您的區域。

```
api.iottwinmaker.region.amazonaws.com
```

如果您停用端點的私有 DNS，則必須執行下列動作，才能透過端點存取 AWS IoT TwinMaker ：

- 在 API 請求中指定 VPC 端點 URL。
  - 對於資料平面 API 操作，請使用下列端點 URL。將 *vpc-endpoint-id* 和*##*取代為您的 VPC 端點 ID 和區域。

```
vpc-endpoint-id.data.iottwinmaker.region.vpce.amazonaws.com
```

  - 對於控制平面 API 操作，請使用下列端點 URL。將 *vpc-endpoint-id* 和*##*取代為您的 VPC 端點 ID 和區域。

```
vpc-endpoint-id.api.iottwinmaker.region.vpce.amazonaws.com
```

- 停用主機字首注入。當您呼叫每個 API 操作時， AWS CLI and AWS SDKs會在服務端點前面加上各種主機字首。當您指定 VPC 端點 AWS IoT TwinMaker 時，這會導致 AWS CLI AWS SDKs 產生無效的 URLs。

> **Important**
>
> 您無法在 AWS CLI 或 中停用主機字首注入 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。

如需如何在 SDKs AWS 中停用主機字首注入的詳細資訊，請參閱每個 SDK 的下列文件章節：

- 適用於 C++ 的 AWS SDK
- 適用於 Go 的 AWS SDK
- 適用於 Go 的 AWS SDK v2
- 適用於 Java 的 AWS SDK
- AWS SDK for Java 2.x
- 適用於 JavaScript 的 AWS SDK
- 適用於 .NET 的 AWS SDK
- 適用於 PHP 的 AWS SDK
- 適用於 Python (Boto3) 的 AWS SDK
- 適用於 Ruby 的 AWS SDK

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。

為 建立 VPC 端點政策 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： AWS IoT TwinMaker 動作的 VPC 端點政策

以下是 端點政策的範例 AWS IoT TwinMaker。連接到端點時，此政策會授予 AWS 所有資源`iottwinmakeradmin`帳戶中 IAM 使用者的所列 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"
         ]
        }
    ]
}
```

的合規驗證 AWS IoT TwinMaker

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

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

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

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

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

中的彈性 AWS IoT TwinMaker

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

如需 AWS 區域 和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

除了 AWS 全球基礎設施之外， AWS IoT TwinMaker 還提供數種功能，以協助支援您的資料彈性和備份需求。

中的基礎設施安全 AWS IoT TwinMaker

作為受管服務， AWS IoT TwinMaker 受到 Amazon Web Services：安全程序概觀白皮書中所述的 AWS 全球網路安全程序的保護。

您可以使用 AWS 發佈的 API 呼叫， AWS IoT TwinMaker 透過網路存取 。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 限流限制的資訊，請參閱 AWS IoT TwinMaker API 限流限制。

有關 AWS IoT TwinMaker 端點的其他資訊

若要以程式設計方式連線至 AWS IoT TwinMaker，請使用 端點。如果您使用 HTTP 用戶端，您需要在控制平面和資料平面 APIs 前加上字首，如下所示。不過，不需要將字首新增至 AWS SDK 和 AWS Command Line Interface 命令，因為它們會自動新增必要的字首。

- 使用控制平面 APIs 的 `api` 字首。例如 `api.iottwinmaker.us-west-1.amazonaws.com`。
- 使用資料平面 APIs 的 `data` 字首。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
新服務連結角色和新的 IAM 政策	AWS IoT TwinMaker新增服務連結角色，稱為 AWSServiceRoleForIoTTwinMaker. 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 AWSIoTTwinMakerServiceRolePolicy 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
初始版本	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。