



開發人員指南

# AWS IoT 1-Click



# AWS IoT 1-Click: 開發人員指南

Copyright © 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，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

AWS IoT 1-Click 是什麼？ .....	1
AWS IoT 1-Click .....	1
AWS IoT 1-Click 的運作方式 .....	3
AWS IoT 1-Click 裝置 .....	3
領取裝置 .....	4
專案、範本和置放 .....	4
開始使用 AWS IoT 1-Click 主控台 .....	7
領取裝置 .....	7
建立專案 .....	7
範例：Meeting Room Satisfaction Project (會議室滿意度專案) .....	8
AWS IoT 1-Click 行動應用程式 .....	11
AWS IoT 1-Click 程式設計模型 .....	12
AWS IoT 1-Click 回呼事件 .....	13
AWS IoT 1-Click 事件 .....	13
AWS IoT 1-Click 運作 Health 活動 .....	14
裝置方法 .....	15
使用 CloudWatch 指標監控使用量 .....	16
使用 AWS CloudTrail 記錄 AWS IoT 1-Click API 呼叫 .....	17
AWS IoT 1-Click 的資訊 .....	17
範例：AWS IoT 1-Click 日誌檔項目 .....	18
AWS CloudFormation 整合 .....	21
AWS IoT 1-Click 的身分驗證與存取控制 .....	22
AWS IoT 1-Click 資源與操作 .....	22
AWS IoT 1-Click 使用以身分為基礎的政策 (IAM 政策) .....	22
AWS IoT 1-Click 的 AWS 受管 (預先定義) 政策 .....	23
標籤 AWS IoT 1-Click 資源 .....	27
標籤基本概念 .....	27
標籤限制 .....	28
AWS IoT Enterprise Button 使用者指南 .....	29
AWS AWS IoT 1-Click 使用 AWS IoT 1-Click .....	31
AWS IoT 1-Click 附錄 .....	49
AWS IoT 1-Click 支援的裝置 .....	49
AWS IoT 1-Click 服務限制 .....	50
文件歷史記錄 .....	52

---

AWS 詞彙表 .....	53
.....	liv

# AWS IoT 1-Click 是什麼？

AWS IoT 1-Click 能讓企業客戶可以輕鬆地將簡單的 IoT 裝置整合到其工作流程中，而無需製造裝置、寫入韌體或將其設定為安全連線。我們的製造合作夥伴建立可以安全連接到 AWS IoT。這些裝置可以觸發 [AWS Lambda](#) 函數是用 Java、Python 和 C# 等語言撰寫的。Lambda 函數可以自行執行商業邏輯，或在 AWS 雲端或現場部署的置放位置觸發動作。

AWS IoT 1-Click 旨在簡化客戶的物聯網，盡可能擷取與裝置硬體和韌體相關的細節。這可讓您將 AWS IoT 1-Click 裝置視為託管在 AWS 雲端中的軟體元件。如同使用任何其他軟體元件，這些裝置符合良好定義的介面。AWS IoT 1-Click 具有依裝置類型定義的介面。您可以使用這些介面來建立和作為應用程式的基礎。

AWS IoT 1-Click，您可以依函數、置放位置或其他準則將裝置分組。這種裝置的邏輯分組稱為「projectAWS IoT 1-Click」。您可以使用專案將裝置群組與 Lambda 函數相關聯，以執行所需的動作。

專案包含用於指定使用哪種裝置的範本、它們叫用的 Lambda 函數以及為這些裝置定義哪些可選屬性 (如位置或函數的相關前後資料)。

建立專案並定義範本後，可以在專案中新增配置 - 每個配置都遵循該範本，並根據序號和屬性值 (金鑰對/值對) 指定實際裝置，使其對特定位置或該特定配置的功能有意義。

## AWS IoT 1-Click

### 取得

指使用 AWS IoT 一鍵主控台、AWS IoT 1-Click 行動應用程式或 AWS IoT 1-Click API 將 AWS IoT 一鍵裝置與 AWS 帳戶建立關聯的程序。

### 申請代碼

用於一次領取多個 AT&T LTE-M 按鈕的值 (即批量)。您也可以使用裝置 ID 來領取裝置。請參閱裝置 ID 項目。

### 裝置

實體裝置，例如 AWS IoT Enterprise Button 或 AT&T LTE-M 按鈕。

### 裝置屬性

以金鑰對/值對的形式與特定裝置關聯的預設或自訂資料。預設屬性會從置放位置衍生。請參閱置放位置項目。

## 裝置 ID

所有裝置都有裝置 ID，例如裝置序號 (DSN)。裝置識別碼可用來使用 AWS IoT 1-Click 一下來註冊 AWS IoT 1 按一下裝置。領取程式碼與裝置識別碼不同。請參閱領取程式碼項目。

## 配置

代表裝置的一或多個範本群組 (例如，包含兩個範本化按鈕的房間)。若要填入置放位置，您可以使用 AWS IoT 1-Click 主控台或 AWS IoT 1-Click 行動應用程式來選擇範本化裝置。

## 配置名稱

置放位置的名稱，通常包括地理置放位置或物件 ID (例如 Room 217、North Dumpster 或 Container 314)。

## Project

包含零個或多個展示置放位置的具名群組 (包含範本化裝置)。

## 專案名稱

一組置放位置的描述性名稱，例如「Meeting Room Satisfaction (會議室滿意度)」或「Charter Container Pickup (章程 Container Pickup)」。

## Template

用於為一組裝置提供預設行為和預設屬性。實體裝置使用特定的範本來獲得該範本的屬性 — 其 Lambda 函數和預設裝置屬性。也就是說，範本定義置放位置中的裝置類別的行為和預設屬性。一個專案可以有多個範本。

## 取消申請

AWS IoT 1-Click 裝置與 AWS 帳戶取消關聯的程序。例如，想要借 AWS IoT 1-Click 裝置的人應該先解除裝置與 AWS 帳戶的關聯，以便新使用者可以將裝置與自己的 AWS 帳戶建立關聯。

# AWS IoT 1-Click 的運作方式

以下是 AWS IoT 1-Click 的工作流程：

1. 選擇一組支援的裝置。
2. 將 AWS Lambda 函數與裝置建立關聯，以觸發動作。您可以使用自己的 Lambda 函數或服務提供的其中一個預先定義函數。
3. 實際部署您的裝置，並使用 AWS IoT 1-Click 主控台、AWS IoT 1-Click 行動應用程式或 AWS IoT 一鍵一鍵 API 來啟用這些裝置。
4. 使用現成 AWS IoT 1-Click 報告或建立自己的報告，取得裝置狀態和使用情況的相關資訊。

## AWS IoT 1-Click 裝置

AWS IoT 1-Click 支援的裝置：

- 是現成的。客戶不需要設計或製造它們。
- 新增到 AWS IoT 1-Click 帳戶，只要使用[宣告功能](#)。
- 在製造時預先佈建了憑證並設定為安全連線到 AWS IoT。您不需要花時間為 AWS IoT 1-Click 裝置安裝憑證。
- AWS IoT 1-Click 裝置type以 AWS IoT 1-Click 定義的標準格式發出事件。例如，所有 AWS IoT 1-Click 裝置button類型具有相同的事件格式，無論製造商為何。
- 具有裝置類型和產品類型。裝置類型會指出裝置發出的事件格式，以及它支援的裝置方法。如需詳細資訊，請參閱[AWS IoT 1-Click 程式設計模型](#)。產品類型提供製造商和品牌詳細資料。例如，如果裝置類型是button，則產品類型可能是 AT&T LTE-M 按鈕。

### Important

AWS IoT 1-Click 支援的裝置在工廠設定為連接到特定[AWS 區域](#)。這些稱為裝置區域。裝置到裝置區域，以確保裝置安全地連線到 AWS IoT，而無需額外的使用者輸入。基於這個原因，您無法變更裝置區域。

AWS IoT 1-Click 裝置發出的事件一律透過預先設定的裝置區域進行路由，以便可以存取同一 AWS 區域中與裝置相關的 Amazon CloudWatch Logs 指標和 AWS Click 指標。裝置區域也是進行裝置計費的地方。

裝置、範本和專案資料儲存在與您帳戶相關聯的 AWS 區域中。此區域可能與裝置區域不同。

如需 AWS IoT 1-Click 支援裝置的相關資訊，包括如何購買和[要求](#)他們，請參閱[AWS IoT 1-Click 附錄](#)。

## 領取裝置

AWS IoT 1-Click 裝置未出廠時，它們與 AWS 客戶帳戶沒有關聯。客戶必須經過領取程序，才能使用其帳戶中的裝置。有兩種方式領取裝置：

- 使用索賠代碼：如果您收到索賠代碼（格式為 C-xxx），您可以將其輸入到 AWS IoT 1-Click 主控台或 AWS IoT 1-Click 行動應用程式，以領取與單一訂單有關的裝置。並非所有裝置（包括 AWS IoT Enterprise Button）都可以使用領取程式碼領取。
- 使用裝置識別碼：您可以使用裝置 ID（裝置序號，也稱為 DSN）透過 AWS IoT 1-Click 行動應用程式領取裝置。所有 AWS IoT 1-Click 裝置都可以使用裝置 ID 來領取。

如需如何宣告裝置的詳細資訊，請參閱 [AWS IoT 1-Click 附錄](#) 和 [領取裝置](#)。

## 專案、範本和置放

裝置可以按功能、位置或任何其他條件進行組織。這種裝置的邏輯分組稱為「專案」。您可以使用專案將裝置群組與 Lambda 函數相關聯。

專案包含用於指定使用哪種裝置的範本，它們所叫用且以名稱歸類保存前後資料的 Lambda 函數，如置放位置或函數。

建立專案並定義範本後，您就可以在專案中新增置放位置。置放位置會遵循範本，並依序列號和屬性值來指定裝置，這些裝置對該置放的特定位置或功能有意義。

以下是一些範例以說明專案和配置的使用情況：

範例 1：

在 SalesPersonNotification 專案中，10 個客戶會收到一個按鈕，他們可以按下按鈕聯絡銷售人員。有 10 個置放位置，每個客戶一個。每個置放位置都具有 CustomerName（例如 Jones 先生）、SalesPersonPhoneNumber（例如，1-555-555-1234）和按鈕序號（例如，G030PM12345678）的值。裝置範本 NotificationButton 會包含在置放位置中。系統會為每個置放位置定義 CustomerName 和 SalesPersonPhoneNumber 屬性。當客戶按一下按鈕時，AWS IoT 1-Click 會叫用 SendSMSLambda 與 CustomerName 和 SalesPersonPhoneNumber 與該按鈕相關聯的值。簡訊會根據這些值發送。



- 置放範本：
  - 由於每個客戶都得到一個按鈕來通知銷售人員，因此建立一個名為 `NotificationButton` 的裝置範本。
  - 裝置範本 (包含在配置中) 指定 `NotificationButton`，點擊時，將調用 `SendSMSLambdaLambda` 函數。
  - 每個配置定義了稱為 `CustomerName` 和 `SalesPersonPhoneNumber` 的屬性。
- 置放位置：建立 10 個置放位置，每個客戶一個置放位置。每個置放位置都有特定的值 `CustomerName` (例如，「Jones 先生」、`SalesPersonPhoneNumber` (例如 1-555-555-1234) 和按鈕序號 (例如，G030PM12345678)。
- 操作：當客戶按一下按鈕時，AWS IoT 1-Click 會調用 `SendSMSLambda` 與 `CustomerName` 和 `SalesPersonPhoneNumber` 值-並根據這些值發送簡訊。

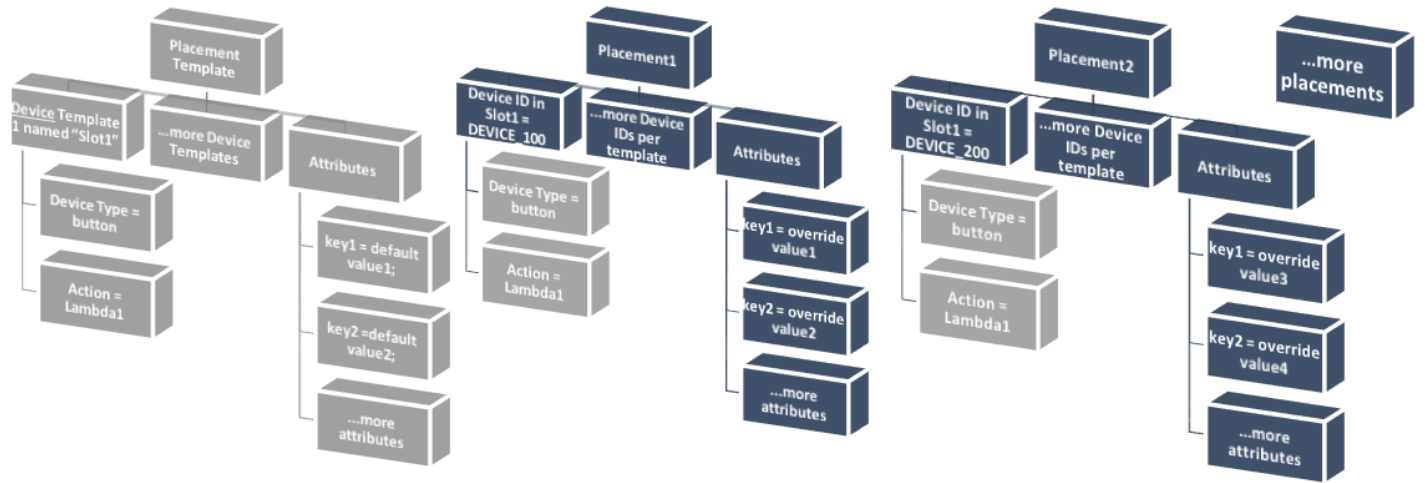
## 範例 2：

在 `MeetingRoomFeedback` 專案中，透過在 50 間會議室內按下「拇指向上」和「拇指向下」按鈕來追蹤使用者滿意度。有兩個裝置範本：`ThumbsUp` 及 `ThumbsDown`。按一下「拇指向上」按鈕時，便會調用 `PostiveFeedbackLambda` 函數。按一下「向下拇指」按鈕時，便會調用 `NegativeFeedbackLambda`。`AMeetingRoomNumber` 屬性定義為保存每個置放位置的會議室號碼。建立 50 個裝置置放位置，每個會議室一個。每個置放位置包含對應特定會議室號碼的 `MeetingRoomNumber` 金鑰組合 (例如，1001) 和兩個按鈕識別它們的唯一序號 (例如，G030PM12345678 和 G030PM23456789)。在會議室中按一下按鈕時，AWS IoT 1-Click 會調用 `PositiveFeedbackLambda` 函數或 `NegativeFeedbackLambda` 函數 `MeetingRoomNumber` 值。然後，您可以處理和製表進行意見回饋。

- 專案名稱：`MeetingRoomFeedback`
- 置放範本：
  - 由於每個會議室都有兩個按鈕，因此建立了兩個裝置範本，分別名為 `ThumbsUp` 和 `ThumbsDown`。
  - 裝置範本指定 `ThumbsUp` 按鈕在按一下時將調用 `PostiveFeedbackLambda`，並且 `ThumbsDown` 按鈕在按一下時將調用 `NegativeFeedbackLambda`。
  - 稱為 `MeetingRoomNumber` 的屬性定義為保存每個配置的會議室號碼。
- 置放位置：建立 50 個裝置的置放位置，每個會議室一個置放位置。每個配置包含 `MeetingRoomNumber` 金鑰設定為特定會議室號碼對 (例如，1001) 和兩個按鈕視為他們的唯一序號 (例如，G030PM12345678 和 G030PM23456789)。

- 操作：在會議室中按一下按鈕時，AWS IoT 1-Click 會叫用PositiveFeedbackLambda函數或NegativeFeedbackLambda函數MeetingRoomNumber值-並且可以處理和製表進行意見回饋。

下圖顯示了這些概念：



如需詳細資訊，請參閱 [開始使用 AWS IoT 1-Click 主控台](#)。

# 開始使用 AWS IoT 1-Click 主控台

下列主題說明如何進行常見的 AWS IoT 1-Click 任務。

## 主題

- [領取裝置](#)
- [建立專案](#)

## 領取裝置

下列程序說明如何申請一或多個 AWS IoT 1-Click 支援的裝置。

1. 登入 AWS 帳戶。如果您尚未擁有 AWS 帳戶，請開啟<https://aws.amazon.com/>中，選擇建立 AWS 帳戶，然後遵循線上指示進行。
2. 從 AWS 管理主控台中，搜尋「1-Click」，然後選擇 AWS IoT 1-Click。
3. 如果您使用的是一個或多個 AWS IoT 企業按鈕，請安裝適用於 iOS 或 Android 的 AWS IoT 1-Click 行動應用程式，然後將按鈕連接到您的區域 Wi-Fi 網路。AWS IoT 1-Click 行動應用程式可從加入頁 AWS IoT 1-Click。LTE-M 按鈕不需要執行此步驟，因為它使用行動網路。
4. 選擇加入，然後選擇宣告裝置。
5. 輸入一或多個[裝置 ID](#)(例如裝置序號) 或[宣告代碼](#)，以逗號分隔，然後選擇取得。如果取得按鈕無法使用，請再次勾選您輸入的所有值。
6. 按下裝置上的按鈕，然後選擇完成。系統應該會顯示所有已知裝置的清單。

## 建立專案

下列程序說明如何為您的 AWS IoT 1-Click 支援裝置建立 AWS IoT 1-Click 專案。

1. 登入您的 AWS 帳戶，並開啟 AWS IoT 1-Click 主控台。
2. 選擇加入，然後選擇建立專案。
3. 輸入專案的名稱和選擇性描述，然後選擇下一頁。
4. 若要定義置放位置的一或多個範本，請在設計裝置範本中，選擇啟動。
5. 若要定義任何按鈕裝置的範本，請選擇所有按鈕類型。
6. 針對 Device template name (裝置範本名稱)，輸入範本的描述性名稱。在 Action (動作) 下方，選擇 Send SMS (傳送簡訊) 或 Send email (傳送電子郵件)。您可以使用使用 Lambda 函數的自訂動

作選項，然後選擇一個您自己的 Lambda 函數。根據您的選擇，輸入電話號碼、電子郵件地址或 Lambda 函數名稱。如需建立 Lambda 函數的詳細資訊，請參閱[AWS Lambda 開發人員指南](#)。

7. UNTER新增另一個裝置範本 (如果每個配置需要多個裝置)選擇Add。
8. 輸入屬性金鑰值對。如有必要，您可以輸入其他金鑰值對。
9. 選擇 Create project (建立專案)。

下一節：[範例：Meeting Room Satisfaction Project \(會議室滿意度專案\)](#)，提供如何使用 AWS IoT 1-Click 主控台來建立專案的實例。

## 範例：Meeting Room Satisfaction Project (會議室滿意度專案)

下列範例可協助您瞭解 AWS IoT 1-Click 的概念。

- 專案追蹤 50 個會議室 (和相關 AV 設備) 的滿意度之建立和命名 MeetingRoomSat。
- 每個會議室將會收到兩個裝置 (按鈕)，一個實體標示為「滿意」，而另一個標示為「不滿意」。由於每個會議室有兩個按鈕，因此將會建立兩個範本，一個名為 Satisfied 和另一個名為 Unsatisfied。
- 所以此Satisfied範本已設定為 Lambda 用名為SatLambda。
- 所以此Unsatisfied範本已設定為 Lambda 用名為UnsatLambda。
- 對於這兩種範本，建立名為 MeetingRoomNum (金鑰) 的屬性 (金鑰對/值組)，其值為 TBD (當兩個按鈕都實體放置在會議室時，TBD 值將變更為會議室號碼)。
- 建立 50 個配置，每個名稱對應到一個會議室。每個配置有兩個相關聯的範本 (例如，Satisfied 和 Unsatisfied)。
- 兩個按鈕都是實體標示，並放入在一個會議室。然後，使用 AWS IoT 1-Click 行動應用程式或 AWS IoT 1-Click 主控台和按鈕的序號，「滿意」和「不滿意」標示按鈕與 50 個配置中的一個關聯。此程序一直持續到所有剩餘配置都部署好。
- 當在會議室中按一下會議室按鈕時，AWS IoT 1-Click 會叫用SatLambda或UnsatLambda函數MeetingRoomNum值 — 並且可以處理和存放在雲端的意見回饋。
- 之後，可以將另一個範本新增到現有的專案中，以便現有的 50 個配置現在包含一個新按鈕的插槽，以指示每個浴室需要更多毛巾或其他洗漱用品。

下方提供使用 AWS IoT 1-Click 主控台建立專案的範例，該專案用於監控辦公大樓 (做為一組辦公大樓的一部分) 中的會議室滿意度。

為了監控會議室 (包括其音訊/視訊設備) 的滿意度，每個會議室都會放置兩個 AWS IoT 企業按鈕，一個標示為「滿意」，另一個標示為「不滿意」。這是一個試行專案，成果可以用於提高會議室客戶在校園內其他建築物的滿意度。

會議結束後，請鼓勵與會者按下「滿意」或「不滿意」按鈕，記錄他們對會議室及其設備的整體滿意度。然後，此資料會用來識別有非功能性 A/V 設備或其他問題的會議室。

AWS IoT 1-Click 主控台可用來設定此專案：

1. 從 AWS IoT 1-Click 主控台，選擇建立專案。
2. 對於專案名稱，請輸入 **MeetingRoomSatisfaction**。如需專案說明，請輸入 **Project used to track customer meeting room satisfaction, including A/V equipment**。選擇下一頁。
3. UNTER設計裝置範本選擇啟動，然後選擇所有按鈕類型。
4. 在 Device template name (裝置範本名稱) 中，輸入 **Satisfied**。這是用於標示為「滿意」的所有按鈕的範本。對於 Action (動作)，請選擇 Send email (傳送電子郵件)。

#### Note

如果會議室滿意度試行方案成功，您可以在 Action (動作) 下選擇 Custom action using a Lambda function (使用 Lambda 函數的自訂動作)。此自訂 Lambda 函數可以傳送電子郵件或將「滿意」按鈕資料儲存在 Amazon DynamoDB 表格中，以供日後分析。如需建立 Lambda 函數的資訊，請參閱[AWS Lambda 開發人員指南](#)。

5. UNTER新增另一個裝置範本 (如果每個配置需要多個裝置)選擇Add，然後選擇所有按鈕類型。在 Device template name (裝置範本名稱) 中，輸入 **Unsatisfied**。這是用於標示為「不滿意」的所有按鈕的範本。對於 Action (動作)，請選擇 Send email (傳送電子郵件)。
6. 對於 Required email default value (必要的電子郵件預設值)，輸入電子郵件地址。對於 Required subject default value (必要的主旨預設值)，輸入 **Meeting Room Feedback**。對於 Required body default value (必要的主體預設值)，輸入 **Either positive or negative meeting room feedback has been provided.**
7. 在 Attribute key (屬性金鑰) 中，輸入 **Building**。針對 Default Value (預設值)，輸入 **Headquarters**。會議室滿意度試執行正在公司總部大樓進行。如果試執行成功，則將部署到公司的其他建築物。因此，關於建立會議室設備將要提供的資訊給誰，了解這點是很重要的。
8. 在第二個金鑰對/值對中，對於Attribute key(屬性金鑰)，輸入 **Room**。針對 Default Value (預設值)，輸入 **TBD**。所以此TBD值將變更為會議室編號 (使用 AWS IoT 1-Click 行動應用程式或 AWS IoT 1-Click 主控台)。

## 9. 選擇 Create project (建立專案)。

使用 AWS IoT 1-Click 行動應用程式，在會議室中放置「滿意」按鈕時，滿意範本會變成與它相關聯，並且**TBD**值會取代為會議室號碼。在會議室中放置「不滿意」按鈕時，也是如此。

# AWS IoT 1-Click 行動應用程式

AWS IoT 1-Click 行動應用程式可讓您：

- 使用類似於 AWS IoT 1-Click 的使用者界面，輕鬆設定和監控現場的 AWS IoT 1-Click 裝置。
- 設定 Wi-Fi 連線 AWS IoT 1-Click 裝置的 Wi-Fi 登入資料 (例如 AWS IoT Enterprise Button)。

AWS IoT 1-Click 的行動應用程式可用於 iPhone 和 Android 的行動裝置。如要下載應用程式，請前往 [應用程式商店](#) 或 [Google 播放](#)，然後按 1-Click 搜尋 AWS IoT。

## AWS IoT 1-Click 程式設計模型

若要使用 AWS IoT 1-Click 裝置建立應用程式，程式設計師會使用[AWS IoT 1-Click 裝置 API](#)與[AWS IoT 1-Click 專案 API](#)。裝置 API 與 AWS IoT 1-Click 裝置元件互動，並處理來自裝置的事件。這些事件包含啟用和停用裝置以及定義事件格式和其觸發的動作 (Lambda 函數)。裝置 API 與位於製造商註冊裝置所在區域的 AWS 元件緊密結合。這就是為什麼[AWS 裝置區域](#)可能與客戶使用該裝置的區域不同。專案 API 與 AWS IoT 1-Click 專案服務互動，並用來彙總管理 AWS IoT 1-Click 裝置，這使得它可以：

- 將裝置群組到專案中。
- 建立用於為專案中所有裝置設定動作的範本。
- 定義儲存與專案相關的關聯資料屬性。

透過使用裝置 API，您可以使用 AWS IoT 1-Click 程式設計模型對各個裝置進程式設計。在這種情況下，您將使用 AWS IoT 1-Click 裝置類型。API 定義了標準事件格式和為該類型的所有裝置形成程式設計界面的方法清單。為了叫用與特定裝置類型相關的方法，程式設計人員可以使用[InvokeDeviceMethod API](#) 並將裝置方法指定為參數。

例如，具有裝置類型「按鈕」的所有 AWS IoT 1-Click 裝置將發出與按一下關聯的事件，並且具有設定在按一下裝置時叫用的回呼函數的方法。如需按鈕介面的相關資訊，請參閱[依裝置類型分類的介面](#)。以下是設置此回呼函數的程式碼：

```
String methodParameters = mapper.writeValueAsString(
    SetOnClickCallbackRequestParameters.builder()
        .deviceId(deviceId)
        .callback(DeviceCallback.builder()
            .awsLambdaArn("arn:aws:lambda:us-
west-2:123456789012:MyButtonListener")
            .build())
        .build());
InvokeDeviceMethodRequest request = new InvokeDeviceMethodRequest()
    .withDeviceMethod(new DeviceMethod()
        .withDeviceType("button")
        .withMethodName("setOnClickCallback"))
    .withDeviceMethodParameters(methodParameters);
```



您可以使用專案 API 來進行裝置機群的程式設計。使用 API 時，必須先定義每個配置的外觀，包括每個配置的裝置範本和屬性。完成後，您將建立具有特定裝置 ID 的置放位置。每個配置遵循相同的範本。以下是範例程式碼來做到這一點：

```
final Map<String, String> callbacks = new HashMap<>();
callbacks.put("onClickCallback", "arn:aws:lambda:us-
west-2:123456789012:MyButtonListener");
final DeviceTemplate item = DeviceTemplate.builder()
    .withDeviceType("button")
    .withCallbackOverrides(callbacks)
    .build();
final Map<String, DeviceTemplate> deviceTemplateMap = new HashMap<>();
deviceTemplateMap.put("MyDevice", item);

final Map<String, String> placementDefaultAttributes = new HashMap<>();
placementDefaultAttributes.put("location", "Seattle")

request = CreateProjectRequest.builder()
    .withProjectName("HelloWorld")
    .withDescription("My first project!")
    .withPlacementTemplate(PlacementTemplate.builder()
        .withDefaultAttributes(placementDefaultAttributes)
        .withDeviceTemplates(deviceTemplateMap)
        .build())
    .build();
projectsClient.createProject(request)
```

## AWS IoT 1-Click 回呼事件

AWS IoT 1-Click 可讓您透過註冊回呼來訂閱裝置事件。回呼範例是 AWS IoT 1-Click 客戶 (也就是 AWS IoT 1-Click 客戶) 擁有和實作的 AWS Lambda 函數。每當有可耗用的事件時，系統就會呼叫此回呼。如需事件及其承載的相關資訊，請參閱 [AWS IoT 1-Click 事件](#) 和 [AWS IoT 1-Click 運作 Health 活動](#) 章節。

### AWS IoT 1-Click 事件

button 類型的裝置每次被點選時都會發佈點選事件。您可以透過以下方式訂閱此事件：

- 呼叫設備上的設備 SetOnClickCallback 方法。

- 適當地設定相關聯的專案，如先前的建立專案程式碼範例所示。

在下列範例中，請注意只有在裝置有相關聯的置放位置時，才會出現 `placementInfo` 區段 如需詳細資訊，請參閱 [專案、範本和置放](#)。

```
{
  "deviceEvent": {
    "buttonClicked": {
      "clickType": "SINGLE",
      "reportedTime": "2018-05-04T23:26:33.747Z"
    }
  },
  "deviceInfo": {
    "attributes": {
      "key3": "value3",
      "key1": "value1",
      "key4": "value4"
    },
    "type": "button",
    "deviceId": " G030PMXXXXXXXXXX ",
    "remainingLife": 5.00
  },
  "placementInfo": {
    "projectName": "test",
    "placementName": "myPlacement",
    "attributes": {
      "location": "Seattle",
      "equipment": "printer"
    },
    "devices": {
      "myButton": " G030PMXXXXXXXXXX "
    }
  }
}
```

## AWS IoT 1-Click 運作 Health 活動

裝置會根據 AWS IoT 1-Click 服務計算的運作狀態參數來發佈運作狀態事件，但您可以設定其對應的閾值。以下範例代表剩餘壽命為 10% 之裝置 `G030PMXXXXXXXXXX` (注意 `"remainingLifeLowerThan":10` 鍵/值對) 的運作狀態事件 JSON 承載。

```

{
  "deviceEvent": {
    "deviceHealthMonitor": {
      "condition": {
        "remainingLifeLowerThan": 10
      }
    }
  },
  "deviceInfo": {
    "attributes": {
      "key2": "value2",
      "key1": "value1",
      "projectRegion": "us-west-2"
    },
    "type": "button",
    "deviceId": "G030PMXXXXXXXXXX",
    "remainingLife": 5.4
  }
}

```

## 裝置方法

AWS IoT 1-Click 裝置方法是支援特定裝置類型的 API，如下表所示。任何裝置支援的裝置方法完整清單，可藉由呼叫 [GetDeviceMethods](#) 取得。

裝置類型	方法名稱	描述
device	getDeviceHealthParameters	取得裝置的運作狀態參數，例如 remainingLife。
device	setDeviceHealthMonitorCallback	設定在裝置運作狀態參數低於閾值時呼叫回呼。
device	getDeviceHealthMonitorCallback	取得在裝置運作狀態參數低於閾值時呼叫的回呼設定。
button	setOnClickCallback	設定在按下按鈕時呼叫回呼。
button	getOnClickCallback	取得在按下按鈕時呼叫的回呼設定。

# 使用 Amazon CloudWatch 監控 AWS IoT 1-Click

AWS IoT 1-Click 會代表您自動監視裝置，並透過[Amazon CloudWatch](#)。這些指標在製造商註冊裝置的裝置區域中回報。如需裝置區域的詳細資訊，請參閱[AWS IoT 1-Click 的運作方式](#)。您可以在 Amazon CloudWatch 儀表板下的按一下命名空間。

Amazon CloudWatch Events 可讓您自動化 AWS 服務，並自動回應系統事件 (例如應用程式可用性的問題或資源的變動)。來自 AWS 服務的事件，會以接近即時的速度傳送到 CloudWatch 事件。您可編寫簡單的規則，來指示您在意的事件，以及當事件符合規則時所要自動執行的動作。您可觸發下列動作：

- 叫用 AWS Lambda 函數。
- 調用 Amazon EC2 運行命令。
- 將事件轉送至 Amazon Kinesis Data Streams。
- 啟動 AWS Step Functions 狀態機器。
- 通知 Amazon SNS 主題或 AWS SMS 佇列。

AWS IoT 1-Click 會追蹤並報告下列指標：

- **TotalEvents** 追蹤裝置發佈的事件數。可以透過裝置事件、專案、裝置類型或產品類型檢視和繪製此指標。
- **RemainingLife** 代表裝置剩餘壽命的近似百分比。AWS IoT 1-Click 會根據裝置的製造商分級回報此數字。例如，如果按鈕設計為可以保持約 2000 下滑鼠點擊數，並且已記錄 500 下滑鼠點擊數，則 RemainingLife 值報告為 75%。系統可以按專案、裝置類型或產品類型檢視和繪製 RemainingLife 指標。客戶可以使用 RemainingLife 指標設定在裝置低於特定閾值時觸發的警報。然後，客戶可以查詢 RemainingLife 的裝置，藉由使用 `GetDeviceHealthParameters` 方法來識別具有低 RemainingLife 值。
- **CallbackInvocationErrors** 指標在裝置發出事件時，追蹤叫用回呼 (Lambda 函數) 時出現的故障。可以透過叫用的回呼 (Lambda 函數 ARN 設定為回呼) 或專案來查看和繪製 CallbackInvocationErrors 指標。客戶可以設定 CallbackInvocationErrors 指標，當 AWS IoT 1-Click 無法將來自其裝置的事件路由到其設定的 Lambda 函數時，指標設定為通知。

如需詳細資訊，請參閱 [Amazon CloudWatch Events 使用者指南](#)。

# 使用 AWS CloudTrail 記錄 AWS IoT 1-Click API 呼叫

AWS IoT 1-Click 是與 AWS CloudTrail 整合的一項服務，可提供由使用者、角色或 AWS 服務在 AWS IoT 1-Click 中所採取動作的記錄。CloudTrail 將 AWS IoT 1-Click 的 API 呼叫擷取為事件。該呼叫的擷取包括從 AWS IoT 1-Click 主控台的呼叫，以及對 AWS IoT 1-Click 操作的程式碼呼叫。如果您建立線索，就可以將 CloudTrail 事件持續交付到 Amazon S3 儲存貯體，包括 AWS IoT 1-Click 的事件。如果您不設定追蹤記錄，仍然可以透過 CloudTrail 主控台內的 Event history (事件歷史記錄) 檢視最新的事件。您可以利用 CloudTrail 所收集的資訊來判斷向 AWS IoT 1-Click 發出的請求，以及發出請求的來源 IP 地址、人員、時間和其他詳細資訊。

若要進一步了解 CloudTrail，包括如何配置及啟用，請參閱 [AWS CloudTrail 使用者指南](#)。

## AWS IoT 1-Click 的資訊

CloudTrailAWS 當您建立帳戶時，系統會在您的帳戶中啟用。此外，AWS IoT 1-Click 發生支援的事件活動時，系統便會將該活動記錄至 CloudTrail 事件，並將其他 AWS 服務事件記錄到事件歷史記錄。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](#)。

若要持續記錄 AWS 帳戶中的事件 (包括 AWS IoT 1-Click 事件)，請建立線索。線索可讓 CloudTrail 將日誌檔案交付到 Amazon S3 儲存貯體。根據預設，當您在主控台建立追蹤記錄時，追蹤記錄會套用到所有 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並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 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [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](#)

AWS IoT 1-Click [裝置 API](#) 支援將下列動作記錄為 CloudTrail 日誌檔的事件：

- [ListDevices](#)
- [DescribeDevice](#)
- [GetDeviceMethods](#)
- [UpdateDeviceState](#)

- [InvokeDeviceMethod](#)

AWS IoT 1-Click [專案 API](#) 支援將下列動作記錄為 CloudTrail 日誌檔的事件：

- [CreateProject](#)
- [UpdateProject](#)
- [DescribeProject](#)
- [ListProjects](#)
- [DeleteProject](#)
- [CreatePlacement](#)
- [UpdatePlacement](#)
- [DescribePlacement](#)
- [ListPlacements](#)
- [DeletePlacement](#)
- [AssociateDeviceWithPlacement](#)
- [DisassociateDeviceFromPlacement](#)
- [GetDevicesInPlacement](#)

每一筆事件或記錄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 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 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登入資料。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail 使用者身分元素](#)。

## 範例：AWS IoT 1-Click 日誌檔項目

線索是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔案並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的是展示 DescribeDevice 動作的 CloudTrail 日誌項目。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2018-04-12T18:57:27Z",
  "eventSource": "iot1click.amazonaws.com",
  "eventName": "DescribeDevice",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "console.aws.amazon.com",
  "requestParameters": {
    "deviceId": "G030PM12345678"
  },
  "responseElements": null,
  "requestID": "573c5654-3e83-11e8-9eac-c999bd01134e",
  "eventID": "be323b62-082a-4352-929d-085d2a3249b0",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "recipientAccountId": "012345678910"
}
```

以下範例顯示的是展示 CreateProject 動作的 CloudTrail 日誌項目。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2018-04-12T20:31:02Z",
  "eventSource": "iot1click.amazonaws.com",
  "eventName": "CreateProject",
  "awsRegion": "us-west-2",
```

```
"sourceIPAddress": "127.0.0.1",
"userAgent": "console.aws.amazon.com",
"requestParameters": {
  "description": "",
  "placementTemplate": {
    "defaultAttributes": "****",
    "deviceTemplates": {
      "happyId": {
        "deviceType": "button",
        "callbackOverrides": {
          "onClickCallback": "arn:aws:lambda:us-
west-2:012345678910:function:rating_buttons_happy"
        }
      },
      "sadId": {
        "deviceType": "button",
        "callbackOverrides": {
          "onClickCallback": "arn:aws:lambda:us-
west-2:012345678910:function:rating_buttons_sad"
        }
      }
    }
  }
}
```



# AWS CloudFormation 整合

AWS IoT 1-Click CloudFormation 是與 AWS CloudFormation 整合的一種通用語言，可讓您在雲端環境中描述和佈建所有基礎結構資源 (例如 Amazon EC2、Auto Scaling、Amazon SNS 等)。AWS CloudFormation 可讓您使用簡單的文字檔，以自動化且安全的方式，為您的應用程式跨所有區域和帳戶建立所需的所有資源模型和佈建。這個檔案是雲端環境唯一的真實資料來源。如需詳細資訊，請參閱 [AWS CloudFormation 使用者指南](#) 以及 AWS IoT 1-Click 的 AWS 主題 (例如 [AWS::IoT1Click::Project](#)) 在 AWS CloudFormation 使用者指南。

# AWS IoT 1-Click 的身分驗證與存取控制

AWS IoT 1-Click 的存取需要登入資料。這些登入資料必須具備 AWS IoT 1-Click 專案或裝置等 AWS 資源的存取許可。下列各節提供如何使用 AWS Identity Access Management (IAM) 和 AWS IoT 1-Click 的詳細資訊，以協助您安全存取資源。

每項 AWS 資源均由某個 AWS 帳戶所持有，而建立或存取資源的許可則由許可政策管理。帳戶管理員可以將許可政策連接到 IAM 身分 (即使用者、群組與角色) 以及某些服務 (例如 AWS Lambda)，也支援將許可政策連接到資源。授予許可時，管理員決定誰能取得這些許可、可存取哪些資源、以及他們想要在這些資源上允許的特定動作。

## AWS IoT 1-Click 資源與操作

AWS IoT 1-Click 中的主資源是專案和裝置。在政策中，您使用 Amazon Resource Name (ARN) 來識別要套用政策的資源。這些資源都有與其相關的唯一 Amazon Resource Name (ARN)，如下表所示。

資源類型	ARN 格式
裝置	arn:aws:iot1click:region:account-id:devices/device-id
Project	arn:aws:iot1click:region:account-id:projects/project-name

AWS IoT 1-Click 實作 API，可搭配 AWS IoT 1-Click 資源使用。這些資源稱為在 IAM 中的動作。有關可用操作的清單，請參閱本主題尾端的表格。

## AWS IoT 1-Click 使用以身分為基礎的政策 (IAM 政策)

本主題提供身分類型政策範例，示範帳戶管理員如何將許可政策連接至 IAM 身分 (即使用者、群組和角色)，並藉此授予許可，以對 AWS IoT 1-Click 資源執行操作。

以下顯示許可政策範例。

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    "iot1click:CreateProject"
  ],
  "Resource": "*"
}
```

政策有一個陳述式-為授予 AWS IoT 1-Click 動作的許可 (iot1click:CreateProject) 在資源上使用應用程式的 Amazon Resource Name (ARN)。在這個案例中，ARN 指定萬用字元 (\*) 來表示為任何資源授予許可。

如需顯示所有 AWS IoT 1-Click 操作與其適用資源的資料表，請參閱[AWS IoT 1-Click 許可：動作、許可與資源參考](#)。

## AWS IoT 1-Click 的 AWS 受管 (預先定義) 政策

Amazon Web Services 透過提供獨立的 IAM 政策來解決許多常用案例，這些政策由 AWS 所建立與管理。這些 AWS 受管政策會授予常用案例所需的許可，讓您免於調查所需的許可。如需詳細資訊，請參閱《[IAM 使用者指南](#)》中的 AWS Managed Policies (AWS 受管政策)。

您可以將下列 AWS 受管政策連接到自己帳戶中的使用者，而這些政策為 AWS IoT 1-Click 特定並依使用案例情況加以分組：

- `AWSIoT1ClickFullAccess`：使用 AWS 管理主控台授予對 AWS IoT 1-Click 資源的完整存取許可。授予的許可包括管理裝置和專案的所有 AWS IoT 1-Click 動作。
- `AWSIoT1ClickReadOnlyAccess`：使用 AWS 管理主控台授予對 AWS IoT 1-Click 資源的唯讀存取許可。該存取可讓使用者能夠列出 AWS IoT 1-Click 裝置和專案，並檢閱專案的組態。

### Note

您可以透過登入 IAM 主控台 (<https://console.aws.amazon.com/iam/>) 並在該處搜尋特定政策名稱。

您也可以建立自己的自訂 IAM 政策，以允許 AWS IoT 1-Click 動作與資源的許可。您可以將這些自訂政策連接至需要這些許可的 IAM 使用者或群組。

## AWS IoT 1-Click 許可：動作、許可與資源參考

當您在 AWS 雲端設定存取控制並撰寫可連接至 IAM 身分的許可政策 (以身分為基礎的政策) 時，可使用下表做為參考。該表格列有各項 AWS IoT 1-Click API 操作、您可以授與執行動作許可的相應動作，以及您可以授與許可的 AWS 資源。您在政策的 Action 欄位中指定動作，然後在政策的 Resource 欄位中指定資源值。

您可以在 AWS IoT 1-Click 政策中使用全 AWS 條件金鑰來表示條件。如需全 AWS 金鑰的完整清單，請參閱[可用的金鑰](#)中的IAM 使用者指南。

### Note

若要指定動作，請使用後接 API 操作名稱的 `iot1click:` 字首 (例如，`iot1click:ListProjects`)。

IoT 1-Click 操作	所需的許可 (API 動作)	資源
ListDevices	iot1click:ListDevices	*
DescribeDevice	iot1click:DescribeDevice	arn:aws:iot1click:region:account-id:devices/device-id
GetDeviceMethods	iot1click:GetDeviceMethods	arn:aws:iot1click:region:account-id:devices/device-id
UpdateDeviceState	iot1click:UpdateDeviceState	arn:aws:iot1click:region:account-id:devices/device-id
InvokeDeviceMethod	iot1click:InvokeDeviceMethod	arn:aws:iot1click:region:account-id:devices/device-id
ListDeviceEvents	iot1click:ListDeviceEvents	arn:aws:iot1click:region:account-id:devices/device-id
InitializeDeviceClaim	iot1click:InitializeDeviceClaim	arn:aws:iot1click:region:account-id:devices/device-id

IoT 1-Click 操作	所需的許可 (API 動作)	資源
FinalizeDeviceClaim	iot1click:FinalizeDeviceClaim	arn:aws:iot1click:region:account-id:devices/device-id
UnclaimDevice	iot1click:UnclaimDevice	arn:aws:iot1click:region:account-id:devices/device-id
ClaimDeviceByClaimCode	iot1click:ClaimDeviceByClaimCode	*
CreateProject	iot1click:CreateProject	arn:aws:iot1click:region:account-id:projects/project-name
UpdateProject	iot1click:UpdateProject	arn:aws:iot1click:region:account-id:projects/project-name
DescribeProject	iot1click:DescribeProject	arn:aws:iot1click:region:account-id:projects/project-name
ListProjects	iot1click:ListProjects	*
DeleteProject	iot1click>DeleteProject	arn:aws:iot1click:region:account-id:projects/project-name
CreatePlacement	iot1click:CreatePlacement	arn:aws:iot1click:region:account-id:projects/project-name
UpdatePlacement	iot1click:UpdatePlacement	arn:aws:iot1click:region:account-id:projects/project-name
DescribePlacement	iot1click:DescribePlacement	arn:aws:iot1click:region:account-id:projects/project-name
ListPlacements	iot1click:ListPlacements	arn:aws:iot1click:region:account-id:projects/project-name
DeletePlacement	iot1click>DeletePlacement	arn:aws:iot1click:region:account-id:projects/project-name

IoT 1-Click 操作	所需的許可 (API 動作)	資源
AssociateDeviceWithPlacement	iot1click:AssociateDeviceWithPlacement	arn:aws:iot1click:region:account-id:projects/project-name
DissociateDeviceFromPlacement	iot1click:DissociateDeviceFromPlacement	arn:aws:iot1click:region:account-id:projects/project-name
GetDevicesInPlacement	iot1click:GetDevicesInPlacement	arn:aws:iot1click:region:account-id:projects/project-name

# 標籤 AWS IoT 1-Click 資源

為協助您管理 AWS IoT 1-Click 資源，您可以選擇性的將您自己的中繼資料，使用標籤指派給任何以 ARN 為基礎的資源。本章說明標籤並示範如何建立標籤。

## 標籤基本概念

標籤可讓您以不同的方式分類 AWS IoT 1-Click 資源，例如依據目的、擁有者或環境。當您有許多相同類型的資源時，這將會很有用，因為您可以依據先前指派的標籤，快速搜尋並識別特定的資源。每個標籤皆包含由您定義的一個「索引鍵」與選擇性的「值」。例如，您可以為特定管理員或帳戶擁有的多個按鈕定義一組標籤。您可以根據您新增的標籤搜尋和篩選資源。我們建議您為每種資源類型建立符合您需求的標籤金鑰。使用一致的標籤金鑰組可讓您更輕鬆的管理您的資源。如需更多資訊，請參閱 [AWS 標記策略](#)。

您也可以使用標籤來分類和追蹤您的成本。當您將標籤套用至資源，AWS 會以逗號分隔值 (CSV) 檔案格式產生一份成本分配報告，其中包含按標籤匯總的用量與成本。您可以套用代表業務類別 (例如成本中心、應用程式名稱或擁有者) 的標籤，來整理多個服務中的成本。如需用於成本配置的標籤詳細資訊，請參閱 [使用成本分配標籤](#) 中的 [AWS Billing and Cost Management 使用者指南](#)。

為了易用性，請使用 AWS 管理主控台內的標籤編輯器，此編輯器可讓您以集中、統一的方式建立和管理標籤。如需詳細資訊，請參閱「[使用標籤編輯器](#)」中的 [AWS 管理主控台入門](#)。

您也可以利用 AWS CLI 及 AWS IoT 1-Click 裝置及專案 API 來使用標籤。當您建立 AWS IoT 1-Click 專案和裝置時，您可以使用 tags 欄位中的命令：

- [CreateProject](#) (專案 API)
- [完成領取](#) (裝置 API)

您可以新增、修改或刪除現有資源的標籤，請使用下列命令：

AWS IoT 1-Click <a href="#">專案 API</a> (使用專案 ARN)	AWS IoT 1-Click <a href="#">裝置 API</a> (使用裝置 ARN)
<a href="#">TagResource</a>	<a href="#">標籤</a>
<a href="#">ListTagsForResource</a>	請參閱 <a href="#">TagResource</a> (POST)、 <a href="#">TagListTagsForResource</a> (GET) 和 <a href="#">UntagResource</a> (DELETE)。
<a href="#">UntagResource</a>	

您可以編輯標籤金鑰和值，並且可以隨時從資源移除標籤。您可以將標籤的值設為空白字串，但您無法將標籤的值設為 null。若您將與現有標籤具有相同鍵的標籤新增到該資源，則新值會覆寫舊值。如果您刪除資源，也會刪除與該資源相關聯的任何標籤。

## 標籤限制

以下基本限制適用於標籤：

- 每一資源最多標籤數 – 50
- 金鑰長度上限 — 127 個 UTF-8 Unicode 字元
- 值長度上限 — 255 個 UTF-8 Unicode 字元
- 標籤金鑰與值皆區分大小寫。
- 標籤名稱或值不可使用 aws: 字首，因為它已保留給 AWS 使用。您不可編輯或刪除具此字首的標籤名稱或值。具此字首的標籤，不算在受資源限制的標籤計數內。
- 如果您的標記結構描述是跨多項服務和資源使用，請記得其他服務可能會有字元使用限制。通常，允許使用的字元為：可用 UTF-8 表示的英文字母、空格和數字，加上以下特殊字元：+ - = . \_ : / @。



# AWS IoT Enterprise Button 使用者指南

AWS IoT Enterprise Button 是一種簡單且易於設定的 Wi-Fi 型按鈕。它是專為企業和開發人員設計，可輕鬆整合使用 AWS IoT 1-Click 的現有業務工作流程和系統。

AWS IoT Enterprise Button 支援三種類型的點按方式：

- 單一
- Double
- 長按

為了讓按鈕能夠正常運作，您必須使用 AWS IoT 1-Click 行動應用程式 (iOS 或 Android) 來設定按鈕的 Wi-Fi 連線 (iOS 或 Android)。在應用程式中，您可以登入 AWS 帳戶來設定按鈕的 Wi-Fi 連線，或按一下應用程式右上角的 Wi-Fi 圖示以略過登入。

透過行動應用程式或主控台設定並要求連線後，當發生按一下、按兩下或長按時，按鈕應該會閃爍綠燈。

如果您在設定按鈕之後懷疑有問題，此表格可協助您進行疑難排解。

顏色	狀態	建議
白色快速閃爍	正在連接 Wi-Fi、正在取得 IP 地址、正在連接 AWS IoT。	無
綠色恆亮	成功連接到 Wi-Fi 並已發佈訊息到 AWS IoT。	無
藍色快速閃爍	按鈕處於組態模式。	等待組態程序完成。
橘色恆亮	未設定 Wi-Fi。	使用 AWS IoT 1-Click 行動應用程式來設定 Wi-Fi。
紅色：短、短、短	連接到已設定的無線網路時發生問題。	檢查是否有變更任何網路設定，或是按鈕離 Wi-Fi 路由器過遠。

顏色	狀態	建議
紅色：短、短、長	從無線網路取得 IP 地址時發生錯誤。	請查看無線網路問題。
紅色：短、長、短	執行主機名稱查詢時發生錯誤。	請查看無線網路問題。
紅色：短、長、長	無法連接到 AWS IoT。	請查看無線網路問題。如果沒有網路問題，但問題仍然存在，請聯絡 <a href="#">AWS Support 中心</a> 並提供裝置序號 (DSN)。您會在按鈕的背面找到此資訊。
紅色：長、短、短	無法與伺服器建立安全連線。	使用 AWS IoT 1-Click iOS 或 Android 行動應用程式來檢查您是否擁有最新的韌體。
紅色：長、短、長	收到 HTTP 403 禁止錯誤。	請聯絡 <a href="#">AWS Support 中心</a> 並提供 DSN。您會在按鈕的背面找到此資訊。
紅色：15 秒後按下按鈕	按鈕重設。	您可以按下按鈕 15 秒，以重設 AWS IoT Enterprise Button Wi-Fi 組態。

# AWS AWS IoT 1-Click 使用 AWS IoT 1-Click

若要示範如何使用 AWS Command Line Interface (AWS CLI)，請考慮使用 AWS IoT 1-Click 簡化其垃圾箱取件服務的處置公司案例。

在此案例中，每個垃圾桶會與 AWS IoT Enterprise Button 配對。垃圾箱已滿時，客戶只需按下相關聯的按鈕，即可請求更換垃圾箱。

## Note

所有 AWS IoT Enterprise Button 裝置 ID 都以「G030PM」開頭。

垃圾處理公司會使用下列步驟來準備供客戶使用的 AWS IoT Enterprise Button。

### 準備 AWS IoT Enterprise Button 供客戶使用

1. AWS IoT Enterprise Button 1-Click 行動裝置應用程式的唯一方法是使用 AWS IoT 1-Click 行動裝置應用程式。若要安裝應用程式，請參閱 [AWS IoT 1-Click 行動應用程式](#)。安裝應用程式後，請勿按下 Login to AWS Account (登入 AWS 帳戶) (通常情況會是如此)。在本練習中，我們想要示範如何使用 AWS CLI。如果您按下 Login to AWS Account (登入 AWS 帳戶)，則系統會為您叫用 `initiate-device-claim` 和 `finalize-device-claim` 命令，我們希望使用 CLI「手動」執行此操作，如下列步驟所示。
2. 對於 AWS CLI 示範目的，而不是按登入 AWS 帳戶下，選擇右上角的小圓形 Wi-Fi 圖示。接下來，選擇 Configure Wi-Fi (設定 Wi-Fi)。掃描或輸入裝置 ID，然後依照行動應用程式的其餘指示操作。
3. 如果您未安裝 AWS CLI，請遵循 [安裝 AWS CLI](#)。若要列出可用的 AWS IoT 1-Click AWS CLI 命令，請執行下列兩個命令。

```
aws iot1click-projects help
```

```
aws iot1click-devices help
```

4. 若要將現在連接到 Wi-Fi 建立關聯的 AWS IoT Enterprise Button 與垃圾處理公司的 AWS 帳戶 Wi-Fi 建立關聯，請使用裝置上的裝置 ID 執行以下命令。

```
aws iot1click-devices initiate-device-claim --device-id G030PM0123456789  
{
```

```
"State": "CLAIM_INITIATED"
}
```

按下裝置上的按鈕。在白光暫時閃爍之後，您應該會看到綠燈恆亮約一秒鐘。如果沒有這樣做，請重複先前的 Wi-Fi 連線程序。

5. 在上一個步驟中看到綠燈恆亮後，請執行下列命令 (使用裝置的 ID 值)。

```
aws iot1click-devices finalize-device-claim --device-id G030PM0123456789
{
  "State": "CLAIMED"
}
```

所以此 "State": "CLAIMED" 回應表示裝置已使用 AWS IoT 1-Click 服務成功註冊裝置。

#### Note

如果裝置製造商提供以「C-」開頭的領取程式碼，您可以只使用 `aws iot1click-devices claim-devices-by-claim-code` 命令來領取一或多個裝置，如下列範例所示。

```
aws iot1click-devices claim-devices-by-claim-code --claim-code C-123EXAMPLE
{
  "Total": 9
  "ClaimCode": "C-123EXAMPLE"
}
```

在此範例中，"Total": 9 表示與領取程式碼相關聯的九個裝置 C-123EXAMPLE 已使用 AWS IoT 1-Click 服務成功領取領取。

6. 接下來，您可以建立名為的 JSON 文字檔，以準備為垃圾處理公司建立適當的 AWS IoT 1-Click 專案。 `create-project.json`。該檔案包含下列項目：

```
{
  "projectName": "SeattleDumpsters",
  "description": "All dumpsters in the Seattle region.",
  "placementTemplate": {
    "defaultAttributes": {
      "City": "Seattle"
    },
    "deviceTemplates": {
      "empty-dumpster-request": {
```

```

        "deviceType": "button"
      }
    }
  }
}

```

placementTemplate 和 deviceTemplates 金鑰值對屬性會套用到 SeattleDumpsters 專案部分的所有按鈕。若要建立此專案，請執行下列命令 (假設 create-project.json 位於 [目前的工作目錄](#))。

```
aws iot1click-projects create-project --cli-input-json file://create-project.json
```

要查看新建立的專案，請執行以下命令。

```
aws iot1click-projects list-projects
{
  "projects": [
    {
      "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
      "projectName": "SeattleDumpsters",
      "createdDate": 1563483100,
      "updatedDate": 1563483100,
      "tags": {}
    }
  ]
}

```

如需詳細資訊，請執行 describe-project 命令，如下所示。

```
aws iot1click-projects describe-project --project-name SeattleDumpsters
{
  "project": {
    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
    "projectName": "SeattleDumpsters",
    "description": "All dumpsters in the Seattle region.",
    "createdDate": 1563483100,
    "updatedDate": 1563483100,
    "placementTemplate": {
      "defaultAttributes": {

```

```

        "City": "Seattle"
      },
      "deviceTemplates": {
        "empty-dumpster-request": {
          "deviceType": "button",
          "callbackOverrides": {}
        }
      }
    },
    "tags": {}
  }
}

```

7. 在為西雅圖區域建立專案後，接下來您要針對特定垃圾箱建立置放位置這 (針對客戶 217)，如下所示。逸出的引號對 Windows 是必要的標記。

```

aws iot1click-projects create-placement --project-name SeattleDumpsters --
placement-name customer217 --attributes "{\"location\": \"1800 9th Ave Seattle, WA
98101\", \"phone\": \"206-123-4567\"}"

```

若要檢視新建立的置放位置，請執行下列命令。

```

aws iot1click-projects list-placements --project-name SeattleDumpsters
{
  "placements": [
    {
      "projectName": "SeattleDumpsters",
      "placementName": "customer217",
      "createdDate": 1563488454,
      "updatedDate": 1563488454
    }
  ]
}

```

如需詳細資訊，請執行 describe-placement 命令，如下所示。

```

aws iot1click-projects describe-placement --project-name SeattleDumpsters --
placement-name customer217
{
  "placement": {
    "projectName": "SeattleDumpsters",
    "placementName": "customer217",

```

```
    "attributes": {
      "phone": "206-123-4567",
      "location": "1800 9th Ave Seattle, WA 98101"
    },
    "createdDate": 1563488454,
    "updatedAt": 1563488454
  }
}
```

8. 雖然裝置現在已與處置公司的 AWS IoT 1-Click 帳戶相關聯，但裝置與置放位置無關聯。透過執行下列命令即可確認此部份：

```
aws iot1click-projects get-devices-in-placement --project-name SeattleDumpsters --
placement-name customer217
{
  "devices": {}
}
```

若要將裝置與置放位置相關聯，請執行下列命令。

```
aws iot1click-projects associate-device-with-placement --project-name
SeattleDumpsters --placement-name customer217 --device-template-name empty-
dumpster-request --device-id G030PM0123456789
```

若要確認前一個命令，請再次執行 `get-devices-in-placement`。

```
aws iot1click-projects get-devices-in-placement --project-name SeattleDumpsters --
placement-name customer217
{
  "devices": {
    "empty-dumpster-request": "G030PM0123456789"
  }
}
```

如需詳細資訊，請執行 `describe-device` 命令，如下所示 (請注意從 `iot1click-projects` 切換到 `iot1click-devices`)。

```
aws iot1click-devices describe-device --device-id G030PM0123456789
{
  "DeviceDescription": {
```

```

    "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789",

    "Attributes": {
      "projectRegion": "us-west-2",
      "projectName": "SeattleDumpsters",
      "placementName": "customer217",
      "deviceTemplateName": "empty-dumpster-request"
    },
    "DeviceId": "G030PM0123456789",
    "Enabled": false,
    "RemainingLife": 99.9,
    "Type": "button",
    "Tags": {}
  }
}

```

因為目前只有一個裝置，下列命令會產生類似的結果。

```

aws iot1click-devices list-devices --device-type button
{
  "Devices": [
    {
      "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/
G030PM0123456789",
      "Attributes": {
        "projectRegion": "us-west-2",
        "projectName": "SeattleDumpsters",
        "placementName": "customer217",
        "deviceTemplateName": "empty-dumpster-request"
      },
      "DeviceId": "G030PM0123456789",
      "Enabled": false,
      "RemainingLife": 99.9,
      "Type": "button",
      "Tags": {}
    }
  ]
}

```

9. 若要確認裝置運作正常，請執行下列命令。適當調整時間戳記，這會使用 [ISO 8061 格式](#)。

```

aws iot1click-devices list-device-events --device-id G030PM0123456789 --from-time-
stamp 2019-07-17T15:45:12.880Z --to-time-stamp 2019-07-19T15:45:12.880Z

```



```
{
  "Events": [
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\"clickType\": \"SINGLE\",
        \"reportedTime\": \"2019-07-18T23:47:55.015Z\", \"certificateId\":
        \"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",
        \"remainingLife\": 99.85000000000001, \"testMode\": false}"
    }
  ]
}
```

在這裡，我們看到一個單擊事件（`\"clickType\": \"SINGLE\"`）在 2019-07-18T 23:47:55.015Z 發生。現在按兩下設備（連續快速按兩下按鈕），然後再次執行該命令。現在請注意按兩下事件（`\"clickType\": \"DOUBLE\"`），如下所示。

```
aws iot1click-devices list-device-events --device-id G030PM0123456789 --from-time-
stamp 2019-07-17T15:45:12.880Z --to-time-stamp 2019-07-19T15:45:12.880Z
{
  "Events": [
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\"clickType\": \"SINGLE\",
        \"reportedTime\": \"2019-07-18T23:47:55.015Z\", \"certificateId\":
        \"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",
        \"remainingLife\": 99.85000000000001, \"testMode\": false}"
    },
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\"clickType\": \"DOUBLE\",
        \"reportedTime\": \"2019-07-19T00:14:41.353Z\", \"certificateId\":
```

```

    \"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",
    \"remainingLife\": 99.8, \"testMode\": false}
  }
]
}

```

10. 每個裝置類型都有一組可叫用的裝置方法。若要列出您裝置類型的可用方法，請執行 `get-device-methods` 命令，如下所示。

```

aws iot1click-devices get-device-methods --device-id G030PM0123456789
{
  "DeviceMethods": [
    {
      "MethodName": "getDeviceHealthParameters"
    },
    {
      "MethodName": "setDeviceHealthMonitorCallback"
    },
    {
      "MethodName": "getDeviceHealthMonitorCallback"
    },
    {
      "MethodName": "setOnClickCallback"
    },
    {
      "MethodName": "getOnClickCallback"
    }
  ]
}

```

若要叫用其中一個可用的方法，請使用 `invoke-device-method` 命令，如下所示。

```

aws iot1click-devices invoke-device-method --cli-input-json file://invoke-device-method.json
{
  "DeviceMethodResponse": "{\"remainingLife\": 99.8}"
}

```

在這裡，`invoke-device-method.json` 包含下列內容：

```

{
  "DeviceId": "G030PM0123456789",

```

```
"DeviceMethod": {
  "DeviceType": "device",
  "MethodName": "getDeviceHealthParameters"
}
}
```

### Note

get 方法 (例如 `getDeviceHealthParameters`) 預期不會有參數。因此，JSON 文件中的 `"DeviceMethodParameters": ""` 行不能被使用 (這樣做會導致這種情況: An error occurred (InvalidRequestException) when calling the InvokeDeviceMethod operation: A request parameter was invalid.)

11. 透過執行 `aws iot1click-devices list-devices --device-type button`，您可以看到 `Enabled` 的預設值是 `false`。下列命令會將此金鑰設定為 `true`。

```
aws iot1click-devices update-device-state --device-id G030PM0123456789 --enabled
```

要將其設回 `false`，請再次使用 `--no-enabled` 引數以執行前一個命令。

12. 如果客戶資訊有所變更，您可以更新裝置的置放資訊，如下所示 (請注意從 `iot1click-devices` 切換到 `iot1click-projects`)。執行下列命令以檢視 `customer217` 的目前資訊 (請參閱 `attributes`)。

```
aws iot1click-projects describe-placement --project-name SeattleDumpsters --
placement-name customer217
{
  "placement": {
    "projectName": "SeattleDumpsters",
    "placementName": "customer217",
    "attributes": {
      "phone": "206-123-4567",
      "location": "1800 9th Ave Seattle, WA 98101"
    },
    "createdDate": 1563488454,
    "updatedAt": 1563488454
  }
}
```

接下來，執行下列命令以更新客戶的電話和置放屬性。

```
aws iot1click-projects update-placement --cli-input-json file://update-
placement.json
```

在這裡，update-placement.json 包含下列內容：

```
{
  "projectName": "SeattleDumpsters",
  "placementName": "customer217",
  "attributes": {
    "phone": "206-266-1000",
    "location": "410 Terry Ave N Seattle, WA 98109"
  }
}
```

若要檢閱此更新，請再次執行 describe-placement，如圖所示。

```
aws iot1click-projects describe-placement --project-name SeattleDumpsters --
placement-name customer217
{
  "placement": {
    "projectName": "SeattleDumpsters",
    "placementName": "customer217",
    "attributes": {
      "phone": "206-266-1000",
      "location": "410 Terry Ave N Seattle, WA 98109"
    },
    "createdDate": 1563488454,
    "updatedAt": 1563572842
  }
}
```

- 若要更新專案資訊，請使用 update-project 命令。一個項目通常包含多個客戶置放位置。以下是現有的 SeattleDumpster 專案資訊。

```
aws iot1click-projects describe-project --project-name SeattleDumpsters
{
  "project": {
    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/
SeattleDumpsters",
    "projectName": "SeattleDumpsters",
    "description": "All dumpsters in the Seattle region.",
```

```

    "createdDate": 1563483100,
    "updatedAt": 1563483100,
    "placementTemplate": {
      "defaultAttributes": {
        "City": "Seattle"
      },
      "deviceTemplates": {
        "empty-dumpster-request": {
          "deviceType": "button",
          "callbackOverrides": {}
        }
      }
    },
    "tags": {}
  }
}

```

要更改「所有垃圾桶在西雅圖區域」為「所有垃圾桶 ( 庭院垃圾、回收箱和垃圾桶)」，請執行下列命令。

```

aws iot1click-projects update-project --project-name SeattleDumpsters --description
  "All dumpsters (yard waste, recycling, garbage) in the Seattle region."

```

您可以看到，"description" 金鑰值已更新為所有的 SeattleDumpsters 置放位置。

```

aws iot1click-projects describe-project --project-name SeattleDumpsters
{
  "project": {
    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
    "projectName": "SeattleDumpsters",
    "description": "All dumpsters (yard waste, recycling, garbage) in the Seattle region.",
    "createdDate": 1563483100,
    "updatedAt": 1563819039,
    "placementTemplate": {
      "defaultAttributes": {
        "City": "Seattle"
      },
      "deviceTemplates": {
        "empty-dumpster-request": {
          "deviceType": "button",

```

```

        "callbackOverrides": {}
      }
    },
    "tags": {}
  }
}

```

14. 您可以使用標籤將中繼資訊套用至專案資源 (iot1click-projects) 和置放資源 (iot1click-devices) , 如下所示。

```
aws iot1click-projects tag-resource --cli-input-json file://projects-tag-resource.json
```

在這裡 , `projects-tag-resource.json` 包含下列內容 :

```

{
  "resourceArn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
  "tags": {
    "Account": "45215",
    "Manager": "Tom Jones"
  }
}

```

若要列出專案資源的標籤 , 請執行下列動作。

```
aws iot1click-projects list-tags-for-resource --resource-arn "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters"
{
  "tags": {
    "Manager": "Tom Jones",
    "Account": "45215"
  }
}

```

若要查看專案標籤的關聯 , 請執行以下操作。

```
aws iot1click-projects describe-project --project-name SeattleDumpsters
{
  "project": {

```

```

    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/
SeattleDumpsters",
    "projectName": "SeattleDumpsters",
    "description": "All dumpsters (yard waste, recycling, garbage) in the
Seattle region.",
    "createdDate": 1563483100,
    "updatedDate": 1563819039,
    "placementTemplate": {
      "defaultAttributes": {
        "City": "Seattle"
      },
      "deviceTemplates": {
        "empty-dumpster-request": {
          "deviceType": "button",
          "callbackOverrides": {}
        }
      }
    },
    "tags": {
      "Manager": "Tom Jones",
      "Account": "45215"
    }
  }
}

```

若要探索裝置亞馬遜資源名稱 (ARN)，請執行下列命令。

```

aws iot1click-devices list-devices
{
  "Devices": [
    {
      "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/
G030PM0123456789",
      "Attributes": {
        "projectRegion": "us-west-2",
        "projectName": "SeattleDumpsters",
        "placementName": "customer217",
        "deviceTemplateName": "empty-dumpster-request"
      },
      "DeviceId": "G030PM0123456789",
      "Enabled": true,
      "RemainingLife": 99.7,
      "Type": "button",

```

```
        "Tags": {}
      }
    ]
  }
}
```

若要將標籤新增至上一個裝置，請執行下列動作。

```
aws iot1click-devices tag-resource --cli-input-json file://devices-tag-resource.json
```

在這裡，`devices-tag-resources.json` 包含以下內容 (請注意要求的 `ResourceArn` 和 `Tags` 大小寫)。

```
{
  "ResourceArn": "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789",
  "Tags": {
    "Driver": "John Smith",
    "Driver Phone": "206-123-4567"
  }
}
```

若要列出裝置資源的標籤，請執行下列動作。

```
aws iot1click-devices list-tags-for-resource --resource-arn "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789"
{
  "Tags": {
    "Driver Phone": "206-123-4567",
    "Driver": "John Smith"
  }
}
```

若要查看關聯的裝置標籤，請執行 `list-devices`。

```
aws iot1click-devices list-devices
{
  "Devices": [
    {
      "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789",
```



```

    "Attributes": {
      "projectRegion": "us-west-2",
      "projectName": "SeattleDumpsters",
      "placementName": "customer217",
      "deviceTemplateName": "empty-dumpster-request"
    },
    "DeviceId": "G030PM0123456789",
    "Enabled": true,
    "RemainingLife": 99.7,
    "Type": "button",
    "Tags": {
      "Driver Phone": "206-123-4567",
      "Driver": "John Smith"
    }
  }
]
}

```

15. 此時，您可以將動作與裝置按鈕建立關聯，例如觸發 AWS Lambda 函數或傳送 Amazon SNS 訊息。您可以使用 AWS IoT 1-Click 主控台 ([AWS IoT 1-Click 程式設計模型](#) 也是一個選項)。在適當的動作與裝置產生關聯之後，您可以將裝置移至客戶的置放位置，然後使用步驟 1 和 2 所述的相同程序，將裝置連接至他們的 Wi-Fi 網路。

## AWS IoT 1-Click 裝置

下列步驟說明如何反轉 (復原) 先前的步驟。

- 若要取消標記專案資源，請執行下列命令。

```
aws iot1click-projects untag-resource --resource-arn "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters" --tag-keys "Manager"
```

這會移除專案的 Manager 標籤，如下所示。

```
aws iot1click-projects describe-project --project-name SeattleDumpsters
{
  "project": {
    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
    "projectName": "SeattleDumpsters",
    "description": "All dumpsters (yard waste, recycling, garbage) in the Seattle region.",

```

```

    "createdDate": 1563483100,
    "updatedAt": 1563819039,
    "placementTemplate": {
      "defaultAttributes": {
        "City": "Seattle"
      },
      "deviceTemplates": {
        "empty-dumpster-request": {
          "deviceType": "button",
          "callbackOverrides": {}
        }
      }
    },
    "tags": {
      "Account": "45215"
    }
  }
}

```

- 若要取消標記裝置資源，請執行下列命令。

```
aws iot1click-devices untag-resource --resource-arn "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789" --tag-keys "Driver Phone" "Driver"
```

這會移除裝置的標籤，如下所示 (請注意空白清單 "Tags": {})

```
aws iot1click-devices list-devices
{
  "Devices": [
    {
      "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789",
      "Attributes": {
        "projectRegion": "us-west-2",
        "projectName": "SeattleDumpsters",
        "placementName": "customer217",
        "deviceTemplateName": "empty-dumpster-request"
      },
      "DeviceId": "G030PM0123456789",
      "Enabled": true,
      "RemainingLife": 99.7,
      "Type": "button",
      "Tags": {}
    }
  ]
}

```

```
    }  
  ]  
}
```

- 若要取消裝置與置放位置的關聯，請執行下列命令。

```
aws iot1click-projects disassociate-device-from-placement --project-name  
SeattleDumpsters --placement-name customer217 --device-template-name empty-  
dumpster-request
```

如下文所示，置放位置 `customer217` 不再具有與其相關聯的裝置。

```
aws iot1click-projects get-devices-in-placement --project-name SeattleDumpsters --  
placement-name customer217  
{  
  "devices": {}  
}
```

- 若要從專案中刪除置放位置，請執行下列命令。

```
aws iot1click-projects delete-placement --project-name SeattleDumpsters --  
placement-name customer217
```

如下所示，專案沒 `SeattleDumpsters` 有置放位置，因為置放位置 `customer217` 是中唯一的置放位置 `SeattleDumpsters`。

```
aws iot1click-projects list-placements --project-name SeattleDumpsters  
{  
  "placements": []  
}
```

- 若要刪除專案，請執行下列命令。

```
aws iot1click-projects delete-project --project-name SeattleDumpsters
```

如您所見，所有專案都會被移除，因為 `SeattleDumpsters` 是唯一與 AWS IoT 1-Click 鍵式帳戶相關聯的專案。

```
aws iot1click-projects list-projects  
{  
  "projects": []  
}
```

```
}
```

舉例來說，如果您想讓朋友使用他們的 AWS 帳戶試用您的裝置，您必須先從 AWS IoT 1-Click 帳戶取消領取裝置，如下所示。

```
aws iot1click-devices unclaim-device --device-id G030PM0123456789
{
  "State": "UNCLAIMED"
}
```

裝置現在可以與任何 AWS IoT 1-Click 帳戶一起使用。

## AWS IoT 1-Click 附錄

本節提供如下所示的 AWS IoT 1-Click 資訊。

### AWS IoT 1-Click 支援的裝置

產品	裝置類型	裝置 ID 字首	領取裝置	購買連結	裝置區域 <sup>†</sup>
AWS IoT IoT 的種子 IoT Button (美國、歐盟及日本)	按鈕	P5SJVQ (裝置 ID 的前 6 位數)	在 AWS IoT 1-Click 行動應用程式中，輸入裝置識別碼以設定 Wi-Fi 並宣告裝置。	<a href="#">Seed Studio Bazaar</a>	US West (Oregon)
伺服器 IoT 按鈕 (僅限美國)	按鈕	7VT4EQ (裝置 ID 的前 6 位數)	在 AWS IoT 1-Click 行動應用程式或 AWS IoT 1-Click 主控台中，輸入您在購買裝置時取得的索賠代碼。您也可以在此 AWS IoT 一鍵行動應用程式中輸入裝置識別碼來申請裝置。	<a href="#">社通行動產品 網上商店</a>	US West (Oregon)
SORCOM LTE-M Button (僅限日本)	按鈕	7MF6JK (裝置 ID 的前 6 位數)	在 AWS IoT 1-Click 行動應用程式中，輸入裝置 ID 來領取裝置。	<a href="#">SORACOM</a>	US West (Oregon)

產品	裝置類型	裝置 ID 字首	領取裝置	購買連結	裝置區域 <sup>†</sup>
AWS IoT Enterprise Button (美國、歐盟及日本)	按鈕	G030PM (裝置 ID 的前 6 位數)	在 AWS IoT 1-Click 行動應用程式中，輸入裝置 ID 以設定 Wi-Fi 並領取裝置。	已停產	US West (Oregon)
AT&T LTE-M 按鈕 (僅限美國)	按鈕	B9GHXT (裝置 ID 的前 6 位數)	在 AWS IoT 1-Click 行動應用程式或 AWS IoT 1-Click 主控台中，輸入購買裝置時所取得的索賠代碼。您也可以在此 AWS IoT 一鍵行動應用程式中輸入裝置識別碼來申請裝置。	已停產	US West (Oregon)

<sup>†</sup>如需關於區域的詳細資訊，請參閱[AWS IoT 1-Click 裝置](#)。

#### Note

AWS IoT 1-Click 不支援其裝置序號 (DSN) 以 G030JF、G030MD 和 G030PT 開始的 AWS IoT Button。若要了解如何將這些按鈕連接到 AWS IoT 雲端 (不使用 AWS IoT 1-Click)，請參閱[雲端可編程儀表板按鈕](#)。

## AWS IoT 1-Click 服務限制

- 每個置放範本最多可以有 5 個裝置範本。這對應於每個置放位置的 5 個裝置。

- 每個 AWS IoT 1-Click 專案最多可有 512 個 [AWS 區域](#) 每個帳戶帳戶。
- 每個 AWS IoT 1-Click 資源最多可有 50 個標籤。標籤是可用於管理資源的鍵/值對 (中繼資料)。如需更多資訊，請參閱 [AWS 標記策略](#)。

## 開發人員指南的文件歷史記錄

下表說明此版本 AWS IoT 1-Click 的說明文件。

- API 版本：最新
- 最新文件更新時間：2018 年 10 月 22 日

變更	描述	日期
發行版本	該文件的初始版本。	2018 年 5 月 14 日
編輯精選	編輯精選的改進功能。	2018 年 5 月 31 日
編輯精選	已更新支援的裝置表格。	2018 年 10 月 22 日



# AWS 詞彙表

For the latest AWS terminology, see the [AWS glossary](#) in the AWS General Reference.

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。