開發人員指南

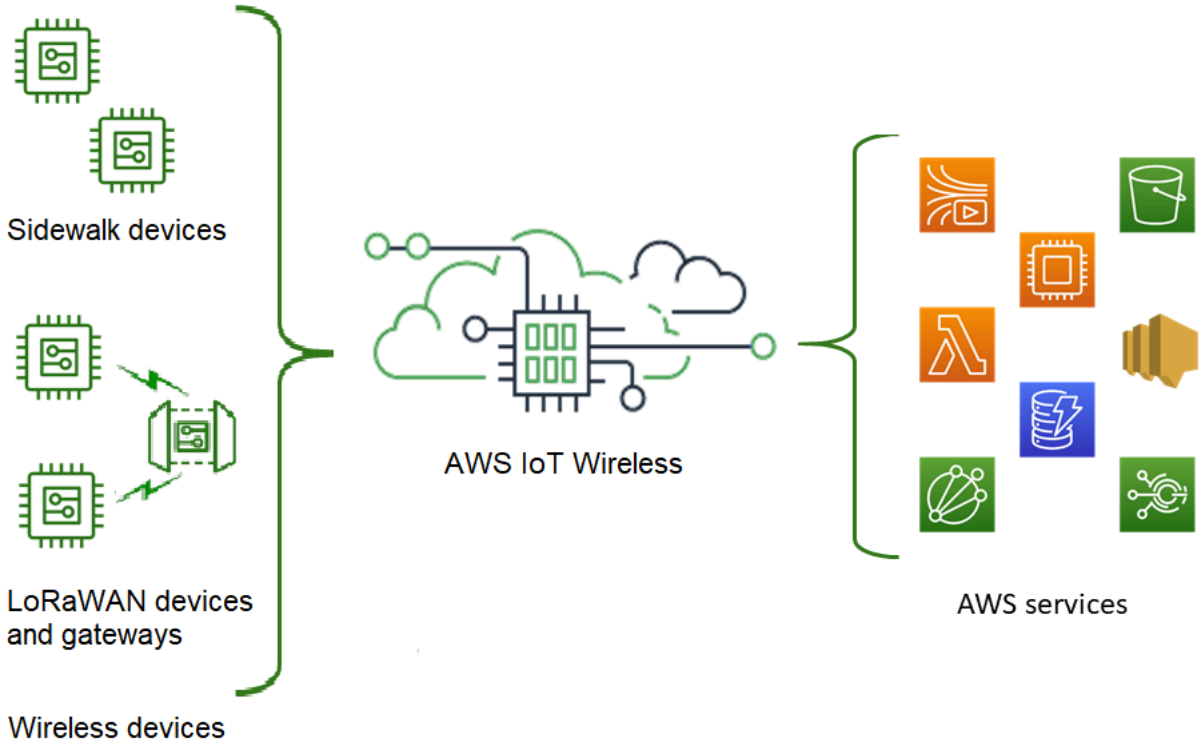
AWS IoT Wireless

AWS IoT Wireless: 開發人員指南

Table of Contents

什麼是 AWS IoT Wireless？

AWS IoT Wireless 提供雲端服務，可將您的無線裝置連線至其他裝置和 AWS 雲端 服務。透過將裝置連限制 AWS IoT Wireless，即可將裝置整合到 AWS IoT 型解決方案中。使用 AWS IoT Wireless 可同時將 LoRaWAN 和 Sidewalk 裝置加入 AWS IoT。這些無線裝置使用低功耗廣域網路 (LPWAN) 通訊協定與 AWS IoT 進行通訊。



AWS IoT Wireless 的功能

AWS IoT Wireless 提供以下功能：

加入 LoRaWAN 和 Sidewalk 裝置

您可以將 LoRaWAN 和 Sidewalk 裝置加入 AWS IoT Wireless。

- AWS IoT Core for LoRaWAN

  若要將您的 LoRaWAN 裝置和閘道加入 AWS IoT Wireless，可使用 AWS IoT Core for LoRaWAN。這是全受管 LoRaWAN 網路伺服器 (LNS)，因此您不需要設定和操作私有 LNS。AWS IoT Core for LoRaWAN 使用組態與更新伺服器 (CUPS) 和無線韌體更新 (FUOTA) 功能來提供閘道管理。如需詳細資訊，請參閱什麼是 AWS IoT Core for LoRaWAN？。

- 適用於 Amazon Sidewalk 的 AWS IoT Core

  若要將 Sidewalk 裝置加入 AWS IoT Wireless，您可以使用適用於 Amazon Sidewalk 的 AWS IoT Core 所提供的功能。Amazon Sidewalk 是一種共用網絡，可連線如 Amazon Echo、Ring 安全攝影機、戶外燈光等裝置，並且可支援您社群中的其他 Sidewalk 裝置。如需詳細資訊，請參閱什麼是適用於 Amazon Sidewalk 的 AWS IoT Core？。

與 AWS IoT Core 的整合

您可以將 AWS IoT Wireless 整合提供的下列功能與 AWS IoT Core 搭配使用：

- 將裝置與 AWS IoT 物件建立關聯

  您可以將無線裝置和閘道與 AWS IoT 物件建立關聯，這樣做有助於將裝置的表示形式儲存在雲端上。使用 AWS IoT 中的物件可讓您更輕鬆地搜尋和管理裝置，以及存取其他 AWS IoT Core 功能。如需詳細資訊，請參閱《AWS IoT Core 開發人員指南》中的使用 AWS IoT 管理裝置。
- 使用 AWS IoT 規則路由訊息

  您可以使用 AWS IoT 的規則功能與其他 AWS 服務 和應用程式進行互動。從您的裝置傳送到雲端的上行訊息可以路由至這些服務和其他應用程式。如需詳細資訊，請參閱《AWS IoT Core 開發人員指南》中的 AWS IoT 規則。

AWS IoT Wireless 新手須知

如果您是第一次使用 AWS IoT Wireless，建議您從閱讀下列各節開始著手：

- 什麼是 AWS IoT Core for LoRaWAN？

  本節概述了 LoRaWAN 技術以及 AWS IoT Core for LoRaWAN 的運作方式。另外還提供了各種資源來幫助您深入了解。
- 什麼是適用於 Amazon Sidewalk 的 AWS IoT Core？

  本節概述了 Amazon Sidewalk 技術，以及適用於 Amazon Sidewalk 的 AWS IoT Core 的運作方式。另外還提供了各種資源來幫助您深入了解。
- 適用於 AWS IoT Core 的 Amazon Sidewalk 入門

  請閱讀本節，以了解如何使用適用於 Amazon Sidewalk 的 AWS IoT Core，以及如何加入您的 Amazon Sidewalk 裝置。

- 將閘道和裝置連接至 AWS IoT Core for LoRaWAN

  接著您可以深入了解如何使用主控台和 API 加入您的 LoRaWAN 裝置。

相關服務

- Amazon CloudWatch

  將 LoRaWAN 或 Sidewalk 裝置加入 AWS IoT Wireless 後，您可以使用 Amazon CloudWatch 即時記錄和監控您的無線裝置和閘道。若要監控 LoRaWAN 裝置和閘道，您也可以使用網路分析器，它能縮短設定連線並開始接收追蹤訊息所需的時間。
- AWS IoT Core

  您還可以使用 AWS IoT Core 整合來連線到可從規則引擎存取的 AWS 服務。如需詳細資訊，請參閱規則引擎使用的 AWS 服務。

存取 AWS IoT Wireless

您可以使用主控台、API 或 CLI 來加入 LoRaWAN 和 Sidewalk 裝置。

- 使用 AWS IoT 主控台

  若要加入您的無線裝置，可使用 AWS Management Console 的 AWS IoT Wireless 頁面。
- 使用 AWS IoT Wireless API

  您可以使用 AWS IoT Wireless API 同時加入 Sidewalk 和 LoRaWAN 裝置。AWS SDK 支援用於建置 AWS IoT Core 的 AWS IoT Wireless API。如需詳細資訊，請參閱 AWS 開發套件與工具組。
- 使用 AWS CLI

  您可以使用 AWS CLI 執行命令來加入和管理您的 LoRaWAN 和 Amazon Sidewalk 裝置。如需詳細資訊，請參閱 AWS IoT Wireless CLI 參考。

AWS IoT Wireless 入門

透過註冊 AWS 帳戶 並按照步驟建立 IAM 使用者，即可開始使用 AWS IoT Wireless。註冊完成後，您就可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI 加入 Sidewalk 和 LoRaWAN 裝置及閘道。加入裝置時，請考慮如何描述和標記資源，以協助您更輕鬆地識別這些資源。

下列主題說明如何開始使用 AWS IoT Wireless。

主題

設定 AWS IoT Wireless

註冊 AWS 時，您的 AWS 帳戶 帳戶會自動註冊 AWS 中的所有服務，包括 AWS IoT Wireless。您只需支付實際使用服務的費用。

請執行下節中的步驟來設定 AWS IoT Wireless：

主題

設定您的 AWS 帳戶

首次使用 AWS IoT Core for LoRaWAN 或適用於 Amazon Sidewalk 的 AWS IoT Core 之前，請先完成下列任務來設定您的 AWS 帳戶：

主題

註冊 AWS 帳戶。

如果您還沒有 AWS 帳戶，請完成以下步驟建立新帳戶。

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，並在電話鍵盤輸入驗證碼。

   註冊 AWS 帳戶時，會建立 AWS 帳戶根使用者。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為最佳安全實務，將管理存取權指派給管理使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

建立 IAM 使用者

若要建立管理員使用者，請選擇下列其中一個選項。

選擇一種管理管理員的方式	到	By	您也可以
在 IAM Identity Center (建議)	使用短期憑證存取 AWS。 這與安全性最佳實務一致。有關最佳實務的資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。	請遵循 AWS IAM Identity Center 使用者指南的入門中的說明。	請參閱 AWS Command Line Interface 使用者指南中的設定 AWS CLI 以使用 AWS IAM Identity Center 設定程式設計存取。
在 IAM 中 (不建議使用)	使用長期憑證存取 AWS。	請遵循 IAM 使用者指南中建立您的第一個 IAM 管理員使用者和使用者群組的說明。	請參閱 IAM 使用者指南中的管理 IAM 使用者的存取金鑰，設定程式設計存取。

以 IAM 使用者身分登入

建立 IAM 使用者後，您可以使用 IAM 使用者名稱和密碼登入 AWS。

以 IAM 使用者身分登入之前，您可以在 IAM 主控台中驗證 IAM 使用者的登入連結。在 IAM 儀表板的「IAM 使用者登入連結」下，您可以看到 AWS 帳戶 的登入連結。您登入連結的 URL 包含 AWS 帳戶 ID，不含破折號 (-)。

如果您不希望登入連結的 URL 包含 AWS 帳戶 ID，則可以建立帳戶別名。如需詳細資訊，請參閱《IAM 使用者指南》中的建立、刪除和列出 AWS 帳戶 別名。

以 IAM 使用者身分登入

1. 登出 AWS Management Console。
2. 輸入您的登入連結，包括您的 AWS 帳戶 ID (但不包括破折號) 或您的 AWS 帳戶 別名。

```
https://aws_account_id_or_alias.signin.aws.amazon.com/console
```

3. 輸入您剛才建立的 IAM 使用者名稱和密碼。

登入時，導覽列會顯示*#your_user_name @ your_aws_account_id#*。

安裝 Python 和 AWS CLI

在您連線 LoRaWAN 或 Sidewalk 終端裝置之前，必須先安裝 Python 並設定 AWS CLI。

> **Important**
>
> 若要執行佈建和註冊 Sidewalk 終端裝置的完整加入工作流程，您還必須設定 Sidewalk 閘道和 HDK。如需指示，請參閱《Amazon Sidewalk 文件》中的設定硬體開發套件 (HDK) 和設定 Sidewalk 閘道。

主題

- 安裝 Python 和 Python3-pip
- 設定 AWS CLI

安裝 Python 和 Python3-pip

若要如下一節所述使用 AWS CLI 和 boto3，您必須使用 Python 3.6 或更新版本。如果您想使用 AWS IoT 主控台加入終端裝置，您可以略過本節並繼續設定 AWS 帳戶。要檢查您是否已經安裝了

Python 和 Python3-pip，請執行以下命令。如果執行這些命令傳回版本，這表示已正確安裝 Python 和 Python3-pip。

```
python3 -V
pip3 --version
```

如果此命令傳回錯誤，原因可能是沒有安裝 Python，或您的作業系統呼叫 Python v3.x 可執行檔作為 Python3。在這種情況下，在執行命令時請以 `python3` 取代 `python` 的所有執行個體。如果仍然產生錯誤，請下載並執行 Python 安裝程式，或根據您的作業系統安裝 Python，如下所述。

Windows

在您的 Windows 電腦上，從 Python 網站下載 Python 然後執行安裝程式，在您的電腦上安裝 Python。

Linux

在 Ubuntu 機器上，執行下列 `sudo` 命令來安裝 Python。

```
sudo apt install python3
sudo apt install python3-pip
```

macOS

在你的 Mac 機器上，使用 Homebrew 來安裝 Python。Homebrew 也會安裝 pip，然後指向已安裝的 Python3 版本。

```
$ brew install python
```

設定 AWS CLI

以下步驟說明如何設定 AWS CLI 和 boto3 (適用於 Python 的 AWS SDK)。您必須先註冊 AWS 帳戶並建立管理使用者，然後再依照這些步驟進行。如需指示，請參閱 設定 AWS IoT Wireless。

1. 安裝及設定 AWS CLI

   您可以使用 AWS CLI，以程式設計方式將您的 Sidewalk 終端裝置加入適用於 Amazon Sidewalk 的 AWS IoT Core。如果您想使用 AWS IoT 主控台來加入裝置，可以跳過本節。開啟 AWS IoT Core 主控台，然後繼續下一節，開始將裝置連接到適用於 Amazon Sidewalk 的 AWS IoT Core。如需有關設定 AWS CLI 的指示，請參閱安裝和設定 AWS CLI。

2. 安裝 boto3 (適用於 Python 的 AWS 開發套件)

   下列命令顯示如何安裝 boto3 (適用於 Python 的 AWS 開發套件) 和 AWS CLI。您也會安裝 botocore，這是執行 boto3 所需的。如需詳細說明，請參閱《Boto3 文件指南》中的安裝 Boto3。

   > **Note**
   >
   > `awscli` 版本 `1.26.6` 需要 3.10 或更高版本的 PyYAML 版本，但不得高於 5.5。

   ```
   python3 -m pip install botocore-version-py3-none-any.whl
   python3 -m pip install boto3-version-py3-none-any.whl
   ```

3. 設定您的憑證和預設區域

   在 `~/.aws/credentials` 和 `~/.aws/config` 檔案中設定您的憑證和預設區域。boto3 程式庫使用這些憑證來識別您的 AWS 帳戶 並授權 API 呼叫。如需組態指示，請參閱：

   - 《Boto3 文件指南》中的組態
   - 《AWS CLI 文件指南》中的組態和憑證檔案設定

描述您的 AWS IoT Wireless 資源

在您開始加入 LoRaWAN 或 Sidewalk 裝置之前，請考慮裝置、閘道和目的地的命名慣例。AWS IoT Wireless 會提供數個選項來協助您識別您建立的資源。如果在建立 AWS IoT Wireless 資源時給與唯一 ID，則此 ID 不具描述性，也不能在建立資源之後加以變更。您也可以指派名稱、新增描述，以及將標籤和標籤值連接至大部分的 AWS IoT Wireless 資源，讓您更方便選取、識別和管理您的資源。

- 資源名稱和描述

  對於裝置、閘道和設定檔，資源名稱是選用欄位，您可以在建立資源之後變更此欄位。名稱會出現在資源中樞頁面上顯示的清單中。

  對於目的地，您會提供在 AWS 帳戶和 AWS 區域 中唯一的名稱。在建立目的地資源之後，您無法修改目的地名稱。

  雖然名稱最多可有 256 個字元，但資源中樞中的顯示空間是有限的。如果可能，請確定名稱的區別部分出現在前 20 到 30 個字元中。

- 資源標籤

標籤是可以附加至 AWS 資源的中繼資料的鍵值對。您可以同時選擇標籤鍵及其對應值。

閘道、目的地和設定檔最多可有 50 個標籤附加至其中。裝置不支援標籤。

資源名稱和描述

名稱的 AWS IoT Wireless 資源支援

資源	名稱欄位支援
目的地	名稱是資源的唯一 ID，無法變更。
無線裝置	名稱是資源的選用描述項，可以變更。
LoRaWAN 閘道	名稱是資源的選用描述項，可以變更。
設定檔	名稱是資源的選用描述項，可以變更。

名稱欄位會出現在資源中樞清單中；不過，空間有限，因此只能看見名稱的前 15-30 個字元。選取資源的名稱時，請考慮您想要它們如何識別資源，以及它們在主控台中的顯示方式。

描述

目的地、裝置和閘道資源也支援描述欄位，此欄位最多可接受 2,048 個字元。描述欄位只會出現在個別資源的詳細資料頁面中。儘管描述欄位可以保存大量資訊，但因為它只會出現在資源的詳細資料頁面中，所以在多個資源的內容中進行掃描並不方便。

資源標籤

AWS 標籤的 AWS IoT Wireless 資源支援

資源	AWS 標籤支援
目的地	最多 50 個 AWS 標籤可以新增至資源。

資源	AWS 標籤支援
無線裝置	此資源不支援 AWS 標籤。
LoRaWAN 閘道	最多 50 個 AWS 標籤可以新增至資源。
設定檔	最多 50 個 AWS 標籤可以新增至資源。

標籤是充當中繼資料的字組或詞組，您可以用來辨識和組織 AWS 資源。您可以將標籤鍵視為資訊類別，並將標籤值視為該類別中的特定值。例如，您可能有一個標籤值 color，然後為一些資源提供該標籤值 blue，以及為其他資源提供標籤值 red。透過此方式，您可以使用 AWS 主控台中的標籤編輯器，來尋找 color 標籤值為 blue 的資源。

如需 AWS IoT Wireless 中進行標記的相關資訊，請參閱 標記您的 AWS IoT Wireless 資源。

如需關於標記和標記策略的詳細資訊，請參閱標籤編輯器。

AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWAN 是全受管 LoRaWAN 網路伺服器 (LNS)，可使用組態與更新伺服器 (CUPS) 和無線韌體更新 (FUOTA) 功能來提供閘道管理。您可以將私有 LNS 取代為 AWS IoT Core for LoRaWAN，並將長距離廣域網路 (LoRaWAN) 裝置和閘道連線到 AWS IoT Core。如此一來，您就可以降低維護、營運和額外負荷成本以及設定時間。

> **Note**
>
> AWS IoT Core for LoRaWAN 僅支援 IPv4 地址格式。它不支援 IPv6 或雙堆疊組態 (IPv4 和 IPv6)。如需詳細資訊，請參閱支援 IPv6 的 AWS 服務。

簡介

LoRaWAN 裝置是長距離、低功耗、電池供電型裝置，使用 LoRaWAN 通訊協定在免授權的無線電頻譜中操作。LoRaWAN 是以 LoRa 為基礎的低功耗廣域網路 (LPWAN) 通訊協定。LoRa 是實體層通訊協定，支持裝置之間的低功耗廣域通訊。

若要將 LoRaWAN 裝置連接至 AWS IoT，您必須使用 LoRaWAN 閘道。閘道可充當將裝置連線至 AWS IoT Core for LoRaWAN 和交換訊息的橋接。AWS IoT Core for LoRaWAN 會使用 AWS IoT 規則引擎，將訊息從您的 LoRaWAN 裝置路由至其他 AWS IoT 服務。

為了減少開發工作並快速將裝置加入 AWS IoT Core for LoRaWAN，建議您使用 LoRaWAN 認證的終端裝置。如需詳細資訊，請參閱 AWS IoT Core for LoRaWAN 產品概觀頁面。如需取得裝置的 LoRaWAN 認證相關資訊，請參閱認證 LoRaWAN 產品。

存取 AWS IoT Core for LoRaWAN

您可以使用主控台或 AWS IoT Wireless API，快速將 LoRaWAN 裝置和閘道加入 AWS IoT Core for LoRaWAN。

使用主控台

若要使用 AWS Management Console 加入 LoRaWAN 裝置和閘道，請登入 AWS Management Console，再導覽至 AWS IoT 主控台中的 AWS IoT Core for LoRaWAN 頁面。然後使用簡介區段將閘

道和裝置新增至 AWS IoT Core for LoRaWAN。如需詳細資訊，請參閱使用主控台將您的裝置和閘道加入至 AWS IoT Core for LoRaWAN。

使用 API 或 CLI

您可以使用 AWS IoT Wireless API 加入 LoRaWAN 和 Sidewalk 裝置。AWS SDK 支援用於建置 AWS IoT Core for LoRaWAN 的 AWS IoT Wireless API。如需詳細資訊，請參閱 AWS 開發套件與工具組。

您可以使用 AWS CLI 執行命令，用於加入和管理 LoRaWAN 閘道和裝置。如需詳細資訊，請參閱 AWS IoT Wireless CLI 參考。

AWS IoT Core for LoRaWAN 區域與端點

AWS IoT Core for LoRaWAN 為專屬於您 AWS 區域 的控制平面和資料平面 API 端點提供支援。專屬於您 AWS 帳戶 和 AWS 區域 的資料平面 API 端點。如需有關 AWS IoT Core for LoRaWAN 端點的詳細資訊，請參閱《AWS 一般參考》中的 AWS IoT Core for LoRaWAN 端點。

為了在您的裝置與 AWS IoT 之間進行更安全的通訊，您可以透過虛擬私有雲端 (VPC) 內的 AWS PrivateLink 將裝置連線至 AWS IoT Core for LoRaWAN，而不要透過公有網路連線。如需詳細資訊，請參閱AWS IoT Core for LoRaWAN 和介面 VPC 端點 (AWS PrivateLink)。

AWS IoT Core for LoRaWAN 擁有的配額可用於在裝置之間傳輸資料，且具有適用於 AWS IoT Wireless API 操作的最大 TPS。如需詳細資訊，請參閱《AWS 一般參考》中的 AWS IoT Core for LoRaWAN 配額。

AWS IoT Core for LoRaWAN 定價

如果您是新客戶，當您註冊 AWS 時，可以透過 AWS 免費方案免費開始使用 AWS IoT Core for LoRaWAN。使用 AWS IoT Core for LoRaWAN 時，您只需按實際用量付費。如需一般產品概觀和定價的詳細資訊，請參閱 AWS IoT Core 定價。

什麼是 AWS IoT Core for LoRaWAN？

AWS IoT Core for LoRaWAN 可取代私有 LoRaWAN 網路伺服器 (LNS)，方法是將 LoRaWAN 裝置和閘道連線到 AWS。您可以使用 AWS IoT 規則引擎路由從 LoRaWAN 裝置收到的訊息，這些訊息會在 LoRaWAN 裝置中格式化並傳送至其他 AWS IoT 服務。AWS IoT Core for LoRaWAN 會使用 X.509 憑證來保護與 AWS IoT 的裝置通訊。

AWS IoT Core for LoRaWAN 會管理服務和裝置政策，而 AWS IoT Core 需要這些政策才能與 LoRaWAN 閘道和裝置通訊。AWS IoT Core for LoRaWAN 也會管理描述 AWS IoT 規則的目的地，而這些規則會將裝置資料傳送至其他服務。

AWS IoT Core for LoRaWAN 的功能

AWS IoT Core for LoRaWAN 可讓您：

- 將 LoRaWAN 裝置和閘道加入並連接到 AWS IoT，無需設定及管理私有 LNS。
- 將 LoRaWAN 裝置連接至符合由 LoRa Alliance 標準化的 1.0.x 或 1.1 LoRaWAN 規格的裝置。這些裝置可以在 A 類、B 類或 C 類模式下運作。
- 使用支援 LoRa Basics Station 2.0.4 版或更新版本的 LoRaWAN 閘道。所有符合 AWS IoT Core for LoRaWAN 資格的閘道都會執行相容版本的 LoRa Basics Station。
- 使用公開提供的 LoRaWAN 網路將您的 LoRaWAN 裝置連線到雲端，這樣可以縮短部署時間，並且不需要管理私有 LoRaWAN 網路，進而節省時間和成本。
- 使用 AWS IoT Core for LoRaWAN 的調適型資料速率監控訊號強度、頻寬和分散因素，並視需要最佳化資料速率。您也可以使用網路分析器來即時監控您的 LoRaWAN 資源。
- 使用 CUPS 服務更新 LoRaWAN 閘道的韌體，使用無線韌體更新 (FUOTA) 更新 LoRaWAN 裝置的韌體。

下列主題將提供有關 LoRaWAN 技術和 AWS IoT Core for LoRaWAN 的詳細資訊。

主題

什麼是 LoRaWAN？

LoRa Alliance 將 LoRaWAN 描述為「低功耗、廣域 (LPWA) 網路通訊協定，旨在以無線方式將電池供電的「物件」連接到區域、國家或全球網路中的網際網路，並將重要物聯網 (IoT) 需求設為目標，例如雙向通訊、端對端安全性、行動性和本地化服務。。

LoRa 和 LoRaWAN

LoRaWAN 通訊協定是在 LoRa 上運作的低功耗廣域網路 (LPWAN) 通訊協定。

LoRaWAN 已是公認的低功耗廣域網路國際標準。如需詳細資訊，請參閱 LoRaWAN 正式認定為 ITU 國際標準。LoRaWAN 是開放規格，因此任何人都可以設定和操作 LoRa 網路。

LoRa 是一種無線音頻技術，於免授權的無線電頻譜中運作。LoRa 是一種實體層通訊協定，使用分散頻譜調變，並支援以窄頻寬的成本進行長距離通訊。它使用窄頻波形搭配中央頻率來傳送資料，這使得它堅固不受干擾。

LoRaWAN 技術的特性

- 在視線範圍內長達 10 英里的遠距離通訊。
- 高達 10 年的電池續航力。為了延長電池壽命，您可以在 A 類或 B 類模式下操作裝置，但這需要增加下行延遲。
- 裝置和維護的成本低廉。
- 免授權的無線電頻譜，但適用於特定區域的法規。
- 低功耗，但有限的承載大小為 51 個位元組到 241 個位元組，視資料速率而定。資料速率可以是 0.3 千位元/秒 至 27 千位元/秒的資料速率，最大承載大小為 222。

LoRaWAN 通訊協定版本

LoRa Alliance 使用 LoRaWAN 規格文件指定 LoRaWAN 通訊協定。為了將區域特定的法規納入考量，LoRa Alliance 同時發布了區域參數文件。如需詳細資訊，請參閱 LoRaWAN 區域參數和規格。

LoRaWAN 的初始版本為 1.0。其他發行版本包括 1.0.1、1.0.2、1.0.3、1.0.4 和 1.1。1.0.1-1.0.4 版通常稱為 1.0.x。

進一步了解 LoRaWAN

下列連結包含 LoRaWAN 技術和 LoRa Basics Station 的實用資訊，LoRa Basics Station 是在 LoRaWAN 閘道上執行的軟體，用於將終端裝置連線至 AWS IoT Core for LoRaWAN。

- LoRaWAN 已認定為 ITU 國際標準

  LoRaWAN 已獲得 ITU 正式認定為低功耗廣域網路國際標準。此標準的正式名稱為 Recommendation ITU-T Y.4480「低功耗廣域無線網路通訊協定」。
- The Things Fundamentals on LoRaWAN

  The Things Fundamentals on LoRaWAN 包含一段簡介視訊，內容涵蓋 LoRaWAN 的基礎原理，以及一系列章節，可協助您了解 LoRa 和 LoRaWAN。

- 什麼是 LoRaWAN

  LoRa Alliance 提供 LoRa 和 LoRaWAN 的技術概觀，包括不同區域的 LoRaWAN 規格摘要。

- LoRa Basics Station

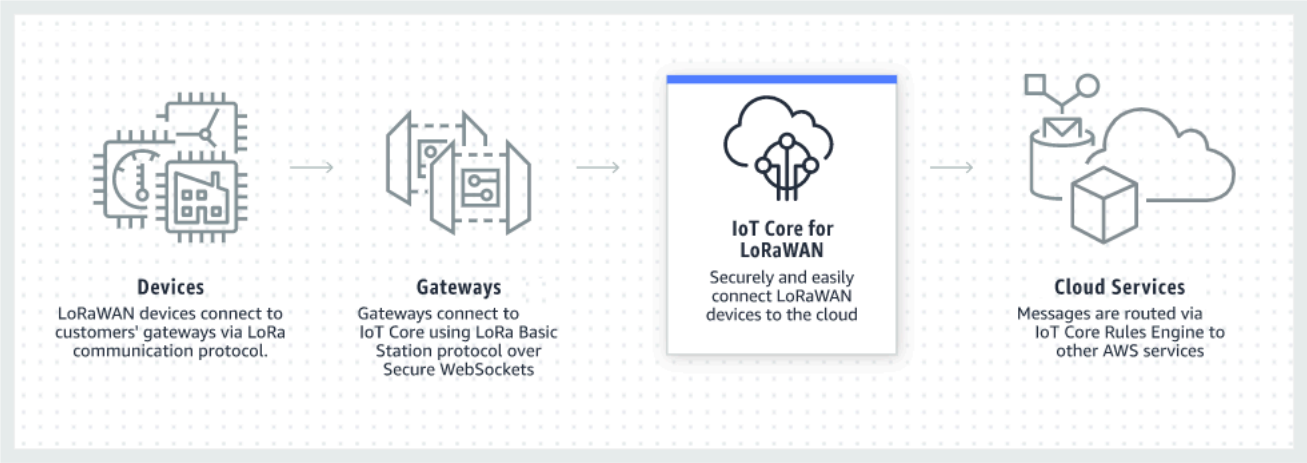
  Semtech Corporation 提供關於閘道和終端節點的 LoRa 基礎知識的實用概念。LoRa Basics Station 是在 LoRaWAN 閘道上執行的開放原始碼軟體，透過 Semtech Corporation 的 GitHub 儲存庫進行維護和分配。您也可以了解 LNS 和 CUPS 通訊協定，這些通訊協定描述如何交換 LoRaWAN 資料以及執行組態更新。

- LoRaWAN 區域參數和規格

  RP002-1.0.2 文件包含對 LoRaWAN Layer 2 規格所有版本的支援。文件中包括有關 LoRaWAN 規格和區域參數，以及不同 LoRaWAN 版本的資訊。

AWS IoT Core for LoRaWAN 的運作方式

LoRaWAN 網路架構部署在星形拓撲中，其中閘道會在終端裝置與 LoRaWAN 網路伺服器 (LNS) 之間轉送資訊。以下顯示 LoRaWAN 裝置如何與 AWS IoT Core for LoRaWAN 互動。此外也示範了 AWS IoT Core for LoRaWAN 如何充當 LNS，並與 AWS 雲端 中的其他 AWS 服務 進行通訊。



LoRaWAN 裝置會透過 LoRaWAN 閘道與 AWS IoT Core 通訊。AWS IoT Core for LoRaWAN 會管理服務和裝置政策，而 AWS IoT Core 需要這些政策才能管理 LoRaWAN 閘道和裝置，並與其進行通訊。AWS IoT Core for LoRaWAN 也會管理描述 AWS IoT 規則的目的地，而這些規則會將裝置資料傳送至其他服務。

開始使用 AWS IoT Core for LoRaWAN。

下列步驟說明如何使用 AWS IoT Core for LoRaWAN 開始進行的概觀。

1. 選取您需要的無線裝置和 LoRaWAN 閘道。

   AWS Partner Device Catalog 包含有資格與 AWS IoT Core for LoRaWAN 搭配使用的閘道和開發人員套件。如需詳細資訊，請參閱使用來自 AWS Partner Device Catalog 的合格閘道。

2. 將您的無線裝置和 LoRaWAN 閘道新增至 AWS IoT Core for LoRaWAN。

   將閘道和裝置連接至 AWS IoT Core for LoRaWAN 提供如何描述您的資源，以及將您的無線裝置和 LoRaWAN 閘道新增至 AWS IoT Core for LoRaWAN 的相關資訊。您也會了解如何設定其他 AWS IoT Core for LoRaWAN 資源，而您需要這些資源，才能管理這些裝置並將其資料傳送至 AWS 服務。

3. 完成您的 AWS IoT Core for LoRaWAN 解決方案。

   從我們的範例 AWS IoT Core for LoRaWAN 解決方案開始著手，並使其成為您的解決方案。

AWS IoT Core for LoRaWAN 資源

下列資源將協助您深入了解 AWS IoT Core for LoRaWAN 以及如何開始進行。

- AWS IoT Core for LoRaWAN 入門

  下列影片介紹了 AWS IoT Core for LoRaWAN 的運作方式，以及逐步解說從 AWS Management Console 新增 LoRaWAN 閘道的程序。

- AWS IoT Core for LoRaWAN 研討會

  此研討會涵蓋了 LoRaWAN 技術的基礎原理，以及其搭配 AWS IoT Core for LoRaWAN 的實作方式。您也可以使用此研討會逐步瀏覽實驗室，這些實驗室展示如何將您的閘道和裝置連線至 AWS IoT Core for LoRaWAN，以建置範例 IoT 解決方案。

- 使用 AWS IoT 實作低功耗廣域網路 (LPWAN) 解決方案

  這份文件為您提供了決策架構，幫助您決定 LPWAN 是否為適合您 IoT 使用案例的正確選擇，此外還提供 LPWAN 連線技術及其功能的概觀，以及實作指引。

將閘道和裝置連接至 AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWAN 可協助您連線和管理無線 LoRaWAN (低功耗長距離廣域網路) 裝置，讓您不需開發和操作 LNS。長距離 WAN (LoRaWAN) 裝置和閘道可以使用 AWS IoT Core for LoRaWAN 連接到 AWS IoT Core。

裝置、閘道、設定檔和目的地的命名慣例

開始使用 AWS IoT Core for LoRaWAN 並建立資源之前，請考慮裝置、閘道和目的地的命名慣例。

AWS IoT Core for LoRaWAN 會將唯一 ID 指派給您為無線裝置、閘道和設定檔建立的資源；不過，您也可以為資源提供更具描述性的名稱，讓您更容易識別這些資源。在將裝置、閘道、設定檔和目的地新增至 AWS IoT Core for LoRaWAN 之前，請考慮您將如何命名它們，以便更容易管理它們。

您也可以將標籤新增至您建立的資源。在新增 LoRaWAN 裝置之前，請考慮您可能如何使用標籤來識別和管理 AWS IoT Core for LoRaWAN 資源。在您新增標籤之後，可以對其進行修改。

如需有關命名和標記的詳細資訊，請參閱 描述您的 AWS IoT Wireless 資源。

將裝置資料映射至服務資料

來自 LoRaWAN 無線裝置的資料通常會進行編碼以最佳化頻寬。這些編碼的訊息到達 AWS IoT Core for LoRaWAN，該訊息所採用的格式可能不容易被其他 AWS 服務使用。AWS IoT Core for LoRaWAN 會使用 AWS IoT 規則，其可以使用 AWS Lambda 函式來處理裝置訊息，並將其解碼為其他 AWS 服務可以使用的格式。

若要轉換裝置資料並將其傳送至其他 AWS 服務，您需要知道：

- 無線裝置傳送的資料格式和內容。
- 您要將資料傳送至哪一個服務。
- 服務需要的格式。

使用該資訊，您可以建立 AWS IoT 規則，執行轉換並將轉換後的資料傳送至將使用它的 AWS 服務。

使用主控台將您的裝置和閘道加入至 AWS IoT Core for LoRaWAN

您可以使用主控台界面或 API 來新增 LoRaWAN 閘道和裝置。如果您使用的是 AWS IoT Core for LoRaWAN，我們建議您使用主控台。當一次管理幾個 AWS IoT Core for LoRaWAN 資源時，主控台界面最實用。當管理大量 AWS IoT Core for LoRaWAN 資源時，請考慮使用 AWS IoT Wireless API 建立更多的自動化解決方案。

您在設定 AWS IoT Core for LoRaWAN 資源時輸入的大部分資料都是由裝置廠商提供，且特定於其支援的 LoRaWAN 規格。下列主題描述如何描述您的 AWS IoT Core for LoRaWAN 資源，以及使用主控台或 API 來新增閘道和裝置。

> **Note**
>
> 如果您使用公有網路將 LoRaWAN 裝置連線至雲端，則可以略過加入您的閘道。如需詳細資訊，請參閱管理來自公有 LoRaWAN 裝置網路的 LoRaWAN 流量 (Everynet)。

主題

- 將閘道加入 AWS IoT Core for LoRaWAN
- 將裝置加入 AWS IoT Core for LoRaWAN

將閘道加入 AWS IoT Core for LoRaWAN

如果您是第一次使用 AWS IoT Core for LoRaWAN，則可以使用主控台來新增您的第一個 LoRaWAN 閘道和裝置。

> **Note**
>
> 如果您使用公有網路將 LoRaWAN 裝置連線至雲端，則可以略過加入您的閘道。如需詳細資訊，請參閱管理來自公有 LoRaWAN 裝置網路的 LoRaWAN 流量 (Everynet)。

在加入您的閘道之前

在將閘道加入至 AWS IoT Core for LoRaWAN 之前，建議您：

- 使用符合與 AWS IoT Core for LoRaWAN 搭配使用資格的閘道。這些閘道無需任何其他組態設定即可連線至 AWS IoT Core，並且執行 2.0.4 版或更新版本的 LoRa Basics Station 軟體。如需詳細資訊，請參閱使用 AWS IoT Wireless 管理閘道。
- 請考慮所建立資源的命名慣例，以便您可以更輕鬆地管理它們。如需詳細資訊，請參閱描述您的 AWS IoT Wireless 資源。
- 請事先準備好每個閘道專屬的組態參數，讓資料輸入至主控台更順暢。AWS IoT 與閘道進行通訊和管理閘道所需的無線閘道組態參數，包括閘道的 EUI 及其 LoRa 頻帶。

為了將閘道加入至 AWS IoT Core for LoRaWAN，請執行下列動作：

- 考慮頻帶選擇並新增必要的 IAM 角色
- 將閘道新增至 AWS IoT Core for LoRaWAN

- 連接您的 LoRaWAN 閘道並驗證其連線狀態

考慮頻帶選擇並新增必要的 IAM 角色

將閘道新增至 AWS IoT Core for LoRaWAN 之前，我們建議您考慮閘道將在其中運作的頻帶，並新增必要的 IAM 角色，以將閘道連接至 AWS IoT Core for LoRaWAN。

> **Note**
>
> 如果您要使用主控台新增閘道，請按一下主控台中的 Create role (建立角色) 以建立必要的 IAM 角色，讓您可以略過這些步驟。只有在使用 CLI 建立閘道時，才需要執行這些步驟。

考慮為閘道和裝置連線選取 LoRa 頻帶

AWS IoT Core for LoRaWAN 支援 EU863-870、US902-928、AU915 和 AS923-1 頻帶，您可以使用這些頻帶來連接實際存在於國家/地區的閘道和裝置，而這些國家/地區支援這些頻帶的頻率範圍和特性。EU863-870 和 US902-928 分別是歐洲和北美洲常用的頻帶。AS923-1 是澳洲、紐西蘭、日本和新加坡等國家常用的頻帶。AU915 是澳大利亞和阿根廷等國家常用的頻帶。如需有關您的地區或國家使用哪個頻帶的詳細資訊，請參閱 LoRaWAN® 區域參數。

LoRa Alliance 發佈了可從 LoRa Alliance 網站下載的 LoRaWAN 規格和區域參數文件。LoRa Alliance 區域參數可協助公司決定在其地區或國家使用哪個頻段。AWS IoT Core for LoRaWAN 的頻帶實作遵循區域參數規格文件中的建議。這些區域參數會分組成一組無線電參數，以及適用於工業、科學和醫療 (ISM) 頻帶的頻率配置。我們建議您與合規團隊合作，以確保您符合任何適用的法規需求。

新增 IAM 角色以允許組態與更新伺服器 (CUPS) 管理閘道憑證

此程序描述如何新增將允許組態與更新伺服器 (CUPS) 管理閘道憑證的 IAM 角色。請確定您在 LoRaWAN 閘道嘗試與 AWS IoT Core for LoRaWAN 連接之前執行此程序；不過，您只需執行此程序一次。

新增 IAM 角色以允許組態與更新伺服器 (CUPS) 管理閘道憑證

1. 開啟 IAM 主控台的角色中樞，然後選擇 Create role (建立角色)。
2. 如果您認為可能已新增 IoTWirelessGatewayCertManagerRole 角色，請在搜尋列中，輸入 **`IoTWirelessGatewayCertManagerRole`**。

   如果在搜尋結果中看到 IoTWirelessGatewayCertManagerRole 角色，則您擁有必要的 IAM 角色。您現在可以離開此程序了。

如果搜尋結果是空的，表示您沒有必要的 IAM 角色。繼續此程序以新增角色。

3. 在 Select type of trusted entity (選取信任的實體類型) 中，選擇 Another AWS 帳戶 (另一個 AWS 帳戶 帳戶)。
4. 在 Account ID (帳戶 ID) 中，請輸入您的 AWS 帳戶 ID，然後選擇 Next: Permissions (下一步：許可)。
5. 在搜尋方塊中，輸入 **`AWSIoTWirelessGatewayCertManager`**。
6. 在搜尋結果清單中，選取名為 AWSIoTWirelessGatewayCertManager 的政策。
7. 選擇 Next: Tags (下一步：標籤)，然後選擇 Next: Review (下一步：檢閱)。
8. 在 Role name (角色名稱) 中，輸入 **`IoTWirelessGatewayCertManagerRole`**，然後選擇 Create role (建立角色)。
9. 若要編輯新角色，請在確認訊息中選擇 IoTWirelessGatewayCertManagerRole。
10. 在 Summary (摘要) 上，選擇 Trust relationships (信任關係) 標籤，然後選擇 Edit trust relationship (編輯信任關係)。
11. 在 Policy Document (政策文件) 中，變更 `Principal` 屬性以看起來像此範例。

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

在您變更 `Principal` 屬性之後，完整政策文件應該看起來像此範例。

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

12. 若要儲存您的變更，請選擇 Update Trust Policy (更新信任政策)。

您現在已建立 IoTWirelessGatewayCertManagerRole。您不需要再次執行此動作。

如果您在新增閘道時執行了此程序，則可以關閉此視窗和 IAM 主控台，並返回 AWS IoT 主控台以完成新增閘道。

將閘道新增至 AWS IoT Core for LoRaWAN

您可以使用主控台或 CLI，將閘道新增至 AWS IoT Core for LoRaWAN。

新增閘道之前，我們建議您考慮 將閘道加入 AWS IoT Core for LoRaWAN 的在加入您的閘道之前一節中提到的因素。

如果您是第一次新增閘道，我們建議您使用主控台。如果您想要改用 CLI 新增閘道，則必須已建立必要的 IAM 角色，以便閘道可以與 AWS IoT Core for LoRaWAN 連接。如需如何建立角色的相關資訊，請參閱 新增 IAM 角色以允許組態與更新伺服器 (CUPS) 管理閘道憑證。

使用主控台新增閘道

導覽至 AWS IoT 主控台的 AWS IoT Core for LoRaWAN Intro (簡介) 頁面，並選擇 Get started (開始使用)，然後選擇 Add gateway (新增閘道)。如果已新增閘道，請選擇 View gateway (檢視閘道)，以檢視您已新增的閘道。如果您想要新增更多閘道，請選擇 Add gateway (新增閘道)。

1. 提供閘道詳細資訊和頻帶資訊

   使用 Gateway details (閘道詳細資訊) 區段，以提供裝置組態資料的相關資訊，例如閘道的 EUI 和頻帶組態。

   - 閘道的 EUI

     個別閘道裝置的 EUI (延伸唯一識別碼)。EUI 是 16 位數的英數代碼，例如 `c0ee40ffff29df10`，可唯一識別 LoRaWAN 網路中的閘道。此資訊專屬於您的閘道型號，而且您可以在閘道裝置或其使用者手冊中找到該資訊。

     > **Note**
     >
     > 閘道的 EUI 與您可能看到列印在閘道裝置上的 Wi-Fi MAC 位址不同。EUI 遵循 EUI-64 標準，可唯一識別您的閘道，因此無法在其他 AWS 帳戶 和 Region 中重複使用。

   - 頻帶 (RFRegion)

閘道的頻帶。您可以選擇 US915、EU868、AU915 或 AS923-1，取決於閘道支援的項目，以及閘道實際連線的國家或地區。如需頻帶的相關資訊，請參閱 考慮為閘道和裝置連線選取 LoRa 頻帶。

2. 指定您的無線閘道組態資料 (選用)

   這些是選用欄位，您可以使用它們來提供關於閘道及其組態的其他資訊。

   - 閘道的名稱、描述和標籤

     這些選用欄位中的資訊來自於您如何組織和描述無線系統中的元素。您可以將 Name (名稱) 指派給閘道、使用Description (描述) 欄位來提供閘道的相關資訊，以及使用 Tags (標籤) 來新增關於閘道的中繼資料的鍵值對。如需有關命名和描述資源的詳細資訊，請參閱 描述您的 AWS IoT Wireless 資源。

   - 使用子頻帶和篩選條件的 LoRaWAN 組態

     您也可以選擇性地指定 LoRaWAN 組態資料，例如您想要使用的子頻帶，以及可以控制流量的篩選條件。針對本教學課程，您可以略過這些欄位。如需詳細資訊，請參閱設定閘道的子頻帶和篩選功能。

3. 將 AWS IoT 物件與閘道建立關聯

   指定是否要建立 AWS IoT 物件，並將其與閘道建立關聯。AWS IoT 中的物件可讓您更輕鬆地搜尋和管理您的裝置。將物件與閘道建立關聯，可讓閘道存取其他 AWS IoT Core 功能。

4. 建立並下載閘道憑證

   若要驗證您的閘道，使其能夠安全地與 AWS IoT 通訊，您的 LoRaWAN 閘道必須提供私有金鑰和憑證給 AWS IoT Core for LoRaWAN。建立 Gateway certificate (閘道憑證)，以便 AWS IoT 可以使用 X.509 標準來驗證閘道的身分。

   按一下 Create certificate (建立憑證) 按鈕並下載憑證檔案。稍後您將使用它們來設定閘道。

5. 複製 CUPS 和 LNS 端點並下載憑證

   建立與 AWS IoT Core for LoRaWAN 的連線時，您的 LoRaWAN 閘道必須連接到 CUPS 或 LNS 端點。建議您使用 CUPS 端點，因為它也可以提供組態管理。若要驗證 AWS IoT Core for LoRaWAN 端點，您的閘道會針對每個 CUPS 和 LNS 端點使用信任憑證，

   按一下 Copy (複製) 按鈕來複製 CUPS 和 LNS 端點。您稍後需要此資訊來設定您的閘道。然後按一下 Download server trust certificates (下載伺服器信任憑證) 按鈕，來下載 CUPS 和 LNS 端點的信任憑證。

6. 建立 IAM 角色以取得閘道許可

您需要新增一個允許組態與更新伺服器 (CUPS) 管理閘道憑證的 IAM 角色。

> **Note**
>
> 在此步驟中，您會建立 IoTWirelessGatewayCertManager 角色。如果您已建立此角色，則可略過此步驟。您必須先執行此動作，然後 LoRaWAN 閘道才會嘗試與 AWS IoT Core for LoRaWAN 連接；不過，您只需執行此動作一次。

若要為您的帳戶建立 IoTWirelessGatewayCertManager IAM 角色，請按一下 Create role (建立角色) 按鈕。如果角色已存在，請從下拉式清單中選取該角色。

按一下 Submit (提交) 以完成閘道建立。

使用 API 新增閘道

如果您是第一次使用 API 或 CLI 新增閘道，則必須新增 IoTWirelessGatewayCertManager IAM 角色，以便閘道可以與 AWS IoT Core for LoRaWAN 連接。如需如何建立角色的相關資訊，請參閱下列 新增 IAM 角色以允許組態與更新伺服器 (CUPS) 管理閘道憑證 一節。

下方清單描述 API 動作，其會執行與新增、更新或刪除 LoRaWAN 閘道相關聯的任務。

AWS IoT Core for LoRaWAN 閘道的 AWS IoT Wireless API 動作

- CreateWirelessGateway
- GetWirelessGateway
- ListWirelessGateways
- UpdateWirelessGateway
- DeleteWirelessGateway

如需可用來建立和管理 AWS IoT Core for LoRaWAN 資源的動作和資料類型完整清單，請參閱 AWS IoT Wireless API 參考。

如何使用 AWS CLI 來新增閘道

您可以使用 AWS CLI 來建立無線閘道，方法為使用 create-wireless-gateway 命令。以下範例會建立無線 LoRaWAN 裝置閘道。您也可以提供 `input.json` 檔案，其中包含其他詳細資訊，例如閘道憑證和佈建憑證。

> **Note**
>
> 您也可以使用 API 中對應於此處顯示的 CLI 命令的方法，在 AWS API 中執行此程序。

```
aws iotwireless create-wireless-gateway \
    --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
    --name "myFirstLoRaWANGateway" \
    --description "Using my first LoRaWAN gateway"
    --cli-input-json input.json
```

如需您可以使用哪些 CLI 的相關資訊，請參閱 AWS CLI 參考

連接您的 LoRaWAN 閘道並驗證其連線狀態

在檢查閘道連線狀態之前，您必須已新增閘道，並將其連接至 AWS IoT Core for LoRaWAN。如需如何新增閘道的相關資訊，請參閱 將閘道新增至 AWS IoT Core for LoRaWAN。

將您的閘道連接到 AWS IoT Core for LoRaWAN

在新增了閘道之後，請連接至閘道的組態界面，以輸入組態資訊和信任憑證。

在將閘道資訊新增至 AWS IoT Core for LoRaWAN 之後，請將一些 AWS IoT Core for LoRaWAN 資訊新增至閘道裝置。閘道廠商提供的文件應該描述將憑證檔案上傳至閘道，以及設定閘道裝置與 AWS IoT Core for LoRaWAN 通訊的程序。

有資格與 AWS IoT Core for LoRaWAN 搭配使用的閘道

如需如何設定 LoRaWAN 閘道的指示，請參閱 AWS IoT Core for LoRaWAN 研討會的設定閘道裝置一節。在這裡，您可以找到閘道連接指示的相關資訊，這些閘道有資格與 AWS IoT Core for LoRaWAN 搭配使用。

支援 CUPS 通訊協定的閘道

以下指示展示如何連接支援 CUPS 通訊協定的閘道。

1. 上傳您在新增閘道時取得的下列檔案。

- 閘道裝置憑證和私有金鑰檔案。
- CUPS 端點的信任憑證檔案，即 `cups.trust`。

2. 指定您先前取得的 CUPS 端點 URL。端點的格式為 *`prefix`*`.cups.lorawan.`*`region`*`.amazonaws.com:443`。

如需有關如何取得此資訊的詳細資訊，請參閱 將閘道新增至 AWS IoT Core for LoRaWAN。

支援 LNS 通訊協定的閘道

以下指示展示如何連接支援 LNS 通訊協定的閘道。

1. 上傳您在新增閘道時取得的下列檔案。
   - 閘道裝置憑證和私有金鑰檔案。
   - LNS 端點的信任憑證檔案，即 `lns.trust`。
2. 指定您先前取得的 LNS 端點 URL。端點的格式為 https://*`prefix`*`.lns.lorawan.`*`region`*`.amazonaws.com:443`。

如需有關如何取得此資訊的詳細資訊，請參閱 將閘道新增至 AWS IoT Core for LoRaWAN。

在將閘道連接至 AWS IoT Core for LoRaWAN 之後，您可以使用主控台或 API 來檢查連線狀態，以及取得何時收到上次上行的相關資訊。

使用主控台檢查閘道連線狀態

若要使用主控台檢查連線狀態，請導覽至 AWS IoT 主控台的 Gateways (閘道) 頁面，然後選擇您已新增的閘道。在閘道詳細資訊頁面的 LoRaWAN specific details (LoRaWAN 特定詳細資訊) 區段中，您會看到連線狀態，以及上次收到上行的日期和時間。

使用 API 檢查閘道連線狀態

若要使用 API 檢查連線狀態，請使用 `GetWirelessGatewayStatistics` API。此 API 沒有要求主體，而且只包含回應主體，其中顯示閘道是否已連線，以及上次收到上行的時間。

```
HTTP/1.1 200
Content-type: application/json

{
  “ConnectionStatus”: “Connected”,
```

```
  “LastUplinkReceivedAt”: “2021-03-24T23:13:08.476015749Z”,
  “WirelessGatewayId”: “30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
}
```

將裝置加入 AWS IoT Core for LoRaWAN

在將閘道加入至 AWS IoT Core for LoRaWAN，並驗證其連線狀態之後，您就可以加入自己的無線裝置。如需如何加入閘道的相關資訊，請參閱 將閘道加入 AWS IoT Core for LoRaWAN。

LoRaWAN 裝置使用 LoRaWAN 通訊協定，與雲端託管的應用程式交換資料。AWS IoT Core for LoRaWAN 支援符合 LoRa Alliance 所標準化之 1.0.x 或 1.1 LoRaWAN 規格的裝置。

LoRaWAN 裝置通常包含一或多個感應器 和 動作者。這些裝置會透過 LoRaWAN 閘道將上行遙測資料傳送至 AWS IoT Core for LoRaWAN。雲端託管的應用程式可以控制感應器，方法為透過 LoRaWAN 閘道將下行命令傳送至 LoRaWAN 裝置。

在加入您的無線裝置之前

在將無線裝置加入至 AWS IoT Core for LoRaWAN 之前，您需要事先備妥以下資訊：

- LoRaWAN 規格與無線裝置組態

  事先準備好要輸入的每個閘道專屬組態參數，可讓資料輸入至主控台更順暢。您需要輸入的特定參數取決於裝置使用的 LoRaWAN 規格。如需其規格和組態參數的完整清單，請參閱每個裝置的文件。

- 裝置名稱和描述 (選用)

  這些選用欄位中的資訊來自於您如何組織和描述無線系統中的元素。如需有關命名和描述資源的詳細資訊，請參閱 描述您的 AWS IoT Wireless 資源。

- 裝置和服務設定檔

  準備好一些無線裝置組態參數，供許多裝置共用，並可儲存在 AWS IoT Core for LoRaWAN 中，作為裝置和服務設定檔。組態參數可以在裝置的文件中或裝置本身上找到。您會想要識別符合裝置組態參數的裝置設定檔，或在必要時建立一個設定檔，然後再新增裝置。如需詳細資訊，請參閱將設定檔新增至 AWS IoT Core for LoRaWAN。

- AWS IoT Core for LoRaWAN destination

  每個裝置都必須指派給目的地，該目的地將處理其要傳送至 AWS IoT 和其他服務的訊息。處理和傳送裝置訊息的 AWS IoT 規則是特定於裝置的訊息格式。若要處理來自裝置的訊息並將它們傳送至正確的服務，請識別您將建立以與裝置訊息搭配使用的目的地，並將它指派給裝置。

將您的無線裝置加入至 AWS IoT Core for LoRaWAN

- 將您的無線裝置新增至 AWS IoT Core for LoRaWAN
- 將設定檔新增至 AWS IoT Core for LoRaWAN
- 新增目的地至 AWS IoT Core for LoRaWAN
- 建立規則來處理 LoRaWAN 裝置訊息
- 連接您的 LoRaWAN 裝置並驗證其連線狀態

將您的無線裝置新增至 AWS IoT Core for LoRaWAN

如果您是第一次新增無線裝置，我們建議您使用主控台。導覽至 AWS IoT 主控台的 AWS IoT Core for LoRaWAN Intro (簡介) 頁面、選擇 Get started (開始使用)，然後選擇 Add device (新增裝置)。如果已新增裝置，請選擇 View device (檢視裝置)，以檢視您已新增的閘道。如果您想要新增更多裝置，請選擇 Add device (新增裝置)。

或者，您也可以從 AWS IoT 主控台的 Devices (裝置)頁面新增無線裝置。

使用主控台將您的無線裝置規格新增至 AWS IoT Core for LoRaWAN

根據您的啟用方法和 LoRaWAN 版本，選擇 Wireless device specification (無線裝置規格)。一旦選取，就會使用 AWS 擁有並為您管理的金鑰來加密您的資料。

OTAA 和 ABP 啟用模式

在您的 LoRaWAN 裝置可以傳送上行資料之前，您必須先完成稱為「啟用」或「聯結」的程序。若要啟用您的裝置，您可以使用 OTAA (無線啟用) 或 ABP (個人化啟用)。

ABP 不需要聯結程序且會使用靜態金鑰。當您使用 OTAA 時，LoRaWAN 裝置會傳送聯結要求，且網路伺服器可以允許該要求。我們建議您使用 OTAA 來啟用裝置，因為每次啟用都會產生新的工作階段金鑰，這樣會使其更安全。

LoRaWAN 版本

當您使用 OTAA 時，您的 LoRaWAN 裝置和雲端託管的應用程式會共用根金鑰。這些根金鑰取決於您使用的是 v1.0.x 版還是 v1.1 版。v1.0.x 只有一個根金鑰，即 AppKey (應用程式金鑰)，而 v1.1 有兩個根金鑰，即 AppKey (應用程式金鑰) 和 NwkKey (網路金鑰)。工作階段金鑰是在每次啟用時根據根金鑰衍生的。NwkKey 和 AppKey 兩者都是無線廠商所提供的十六進位值 (32 位數)。

無線裝置 EUI

在您選取 Wireless device specification (無線裝置規格) 之後，您會看到主機上顯示無線裝置的 EUI (延伸唯一識別碼) 參數。您可以從裝置或無線廠商的文件中找到此資訊。

- DevEUI：16 位數的十六進位值，這是您裝置獨有的，可在裝置標籤或其文件上找到。
- AppEUI：16 位數的十六進位值，這是聯結伺服器獨有的，可在裝置文件中找到。在 LoRaWAN v1.1 版中，AppEUI 被稱為 JoinEUI。

如需唯一識別碼、工作階段金鑰和根金鑰的詳細資訊，請參閱 LoRa Alliance 文件。

使用 API 將無線裝置規格新增至 AWS IoT Core for LoRaWAN

如果您是使用 API 新增無線裝置，則必須先建立裝置設定檔和服務設定檔，然後才能建立無線裝置。在建立無線裝置時，您將使用裝置設定檔和服務設定檔 ID。如需如何使用 API 建立這些設定檔的相關資訊，請參閱 使用 API 新增裝置設定檔。

下方清單描述 API 動作，其會執行與新增、更新或刪除服務設定檔相關聯的任務。

服務設定檔的 AWS IoT Wireless API 動作

- CreateWirelessDevice
- GetWirelessDevice
- ListWirelessDevices
- UpdateWirelessDevice
- DeleteWirelessDevice

如需可用來建立和管理 AWS IoT Core for LoRaWAN 資源的動作和資料類型完整清單，請參閱 AWS IoT Wireless API 參考。

如何使用 AWS CLI 建立無線裝置

您可以使用 AWS CLI 來建立無線裝置，方法為使用 create-wireless-device 命令。下列範例會建立無線裝置，方法為使用 input.json 檔案來輸入參數。

> **Note**
>
> 您也可以使用 API 中對應於此處顯示的 CLI 命令的方法，在 AWS API 中執行此程序。

input.json 的內容

```
{
    "Description": "My LoRaWAN wireless device"
    "DestinationName": "IoTWirelessDestination"
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Name": "SampleIoTWirelessThing"
    "Type": LoRaWAN
}
```

您可以提供此檔案作為 `create-wireless-device` 命令的輸入。

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

如需您可以使用哪些 CLI 的相關資訊，請參閱 AWS CLI 參考

將設定檔新增至 AWS IoT Core for LoRaWAN

裝置和服務設定檔可以定義為描述常用的裝置組態。這些設定檔描述裝置共用的設定參數，讓您更輕鬆地新增這些裝置。AWS IoT Core for LoRaWAN 支援裝置設定檔和服務設定檔。

要輸入至這些設定檔的組態參數和值是由裝置製造商提供。

新增裝置設定檔

裝置設定檔定義網路伺服器用來設定 LoRaWAN 無線電存取服務的裝置功能和開機參數。它包括參數的選擇，例如 LoRa 頻帶、LoRa 區域參數版本，以及裝置的 MAC 版本。若要進一步了解不同頻帶，請參閱 考慮為閘道和裝置連線選取 LoRa 頻帶。

使用主控台新增裝置設定檔

如果您是使用主控台新增無線裝置，如 使用主控台將您的無線裝置規格新增至 AWS IoT Core for LoRaWAN 所述，則在新增了無線裝置規格之後，您就可以新增裝置設定檔。或者，您也可以在 LoRaWAN 標籤上從 AWS IoT 主控台的 Profiles (設定檔) 頁面新增無線裝置。

您可以選擇預設裝置設定檔或建立新的裝置設定檔。建議您使用預設裝置設定檔。如果您的應用程式需要您建立裝置設定檔，請提供 Device profile name (裝置設定檔名稱)、選取您正在對裝置和閘道使用的 Frequency band (RfRegion) (頻帶 (RfRegion))，並將其他設定保留為預設值，除非裝置文件中另有指定。

使用 API 新增裝置設定檔

如果您是使用 API 新增無線裝置，則必須先建立裝置設定檔，然後才能建立無線裝置。

下方清單描述 API 動作，其會執行與新增、更新或刪除服務設定檔相關聯的任務。

服務設定檔的 AWS IoT Wireless API 動作

- CreateDeviceProfile
- GetDeviceProfile
- ListDeviceProfiles
- UpdateDeviceProfile
- DeleteDeviceProfile

如需可用來建立和管理 AWS IoT Core for LoRaWAN 資源的動作和資料類型完整清單，請參閱 AWS IoT Wireless API 參考。

如何使用 AWS CLI 建立裝置設定檔

您可以使用 AWS CLI 來建立裝置設定檔，方法為使用 create-device-profile 命令。以下範例會建立裝置設定檔。

```
aws iotwireless create-device-profile
```

執行此命令會自動建立具有 ID 的裝置設定檔，您可以在建立無線裝置時使用此 ID。您現在可以使用下列 API 建立服務設定檔，然後使用裝置和服務設定檔建立無線裝置。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

如需您可以使用哪些 CLI 的相關資訊，請參閱 AWS CLI 參考

新增裝置設定檔

服務設定檔描述裝置與應用程式伺服器通訊所需的通訊參數。

使用主控台新增服務設定檔

如果您是使用主控台新增無線裝置，如 使用主控台將您的無線裝置規格新增至 AWS IoT Core for LoRaWAN 所述，則在新增了裝置設定檔之後，您就可以新增服務設定檔。或者，您也可以在 LoRaWAN 標籤上從 AWS IoT 主控台的 Profiles (設定檔) 頁面新增無線裝置。

建議您將設定 AddGWMetaData 保留為啟用狀態，以便您可以收到每個承載的額外閘道中繼資料，例如用於資料傳輸的 RSSI 和 SNR。

使用 API 新增服務設定檔

如果您是使用 API 新增無線裝置，則必須先建立服務設定檔，然後才能建立無線裝置。

下方清單描述 API 動作，其會執行與新增、更新或刪除服務設定檔相關聯的任務。

服務設定檔的 AWS IoT Wireless API 動作

- CreateServiceProfile
- GetServiceProfile
- ListServiceProfiles
- UpdateServiceProfile
- DeleteServiceProfile

如需可用來建立和管理 AWS IoT Core for LoRaWAN 資源的動作和資料類型完整清單，請參閱 AWS IoT Wireless API 參考。

如何使用 AWS CLI 建立服務設定檔

您可以使用 AWS CLI 來建立服務，方法為使用 create-service-profile 命令。以下範例會建立服務設定檔。

```
aws iotwireless create-service-profile
```

執行此命令會自動建立具有 ID 的服務設定檔，您可以在建立無線裝置時使用此 ID。您現在可以使用裝置和服務設定檔來建立無線裝置。

```
{
```

```
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

新增目的地至 AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWAN 描述處理裝置資料以供 AWS 服務使用的 AWS IoT 規則。

因為大部分 LoRaWAN 裝置不會以 AWS 服務可以使用的格式，將資料傳送至 AWS IoT Core for LoRaWAN，所以 AWS IoT 規則必須先處理它。AWS IoT 規則包含解譯裝置資料的 SQL 陳述式，以及將 SQL 陳述式結果傳送至將使用它的服務的主題規則動作。

如果是第一次新增目的地，我們建議您使用主控台。

使用主控台新增目的地

如果是使用主控台新增無線裝置 (如 使用主控台將您的無線裝置規格新增至 AWS IoT Core for LoRaWAN 所述)，則在已將無線裝置規格和設定檔新增至 AWS IoT Core for LoRaWAN 之後，您可以繼續並新增目的地。

您也可以從 AWS IoT 主控台的 Destinations (目的地) 頁面新增 AWS IoT Core for LoRaWAN 目的地。

若要處理裝置的資料，請在建立 AWS IoT Core for LoRaWAN 目的地時指定下列欄位，然後選擇 Add destination (新增目的地)。

- 目的地詳細資訊

  為您的目的地輸入 Destination name (目的地名稱) 和選用描述。
- 規則名稱

  AWS IoT 規則被設定為評估裝置傳送的訊息並處理裝置的資料。規則名稱會映射至目的地。目的地需要規則來處理接收到的訊息。您可以選擇透過叫用 AWS IoT 規則或發佈至 AWS IoT 訊息代理程式的方式來處理訊息。

  - 如果選擇 Enter a rule name (輸入規則名稱)，請輸入名稱然後選擇 Copy (複製)，來複製您在建立 AWS IoT 規則時將輸入的規則名稱。您可以選擇 Create rule (建立規則) 來立即建立規則，或導覽至 AWS IoT 主控台的規則中樞，並使用該名稱建立規則。

    您也可以輸入規則，再使用 Advanced (進階) 設定來指定主題名稱。主題名稱會在規則叫用期間提供，而且可以使用規則中的 `topic` 運算式存取。如需AWS IoT規則的詳細資訊，請參閱 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html。

- 若您選擇發佈至 AWS IoT 訊息代理程式，請輸入主題名稱。然後您可以複製 MQTT 主題名稱，多位訂閱者可以訂閱此主題，以接收發佈至該主題的訊息。如需詳細資訊，請參閱https://docs.aws.amazon.com/iot/latest/developerguide/topics.html。

  如需有關目的地之 AWS IoT 規則的詳細資訊，請參閱 建立規則來處理 LoRaWAN 裝置訊息。

- 角色名稱

  授予裝置資料許可，以存取在 Rule name (規則名稱) 中命名之規則的 IAM 角色。您可以在主控台中建立新的服務角色，或選取現有的服務角色。如果正在建立新的服務角色，您可以輸入角色名稱 (例如 **`IoTWirelessDestinationRole`**)，或為 AWS IoT Core for LoRaWAN 保留為空白以產生新的角色名稱。AWS IoT Core for LoRaWAN 會代表您自動建立具有適當許可的 IAM 角色。

  如需有關 IAM 角色的詳細資訊，請參閱使用 IAM 角色。

使用 API 新增目的地

如果想要改用 CLI 來新增目的地，您必須已為目的地建立規則和 IAM 角色。如需目的地在角色中需要的詳細資訊，請參閱 為您的目的地建立 IAM 角色。

下方清單包含執行與新增、更新或刪除目的地相關聯之任務的 API 動作。

目的地的 AWS IoT Wireless API 動作

- CreateDestination
- GetDestination
- ListDestinations
- UpdateDestination
- DeleteDestination

如需可用來建立和管理 AWS IoT Core for LoRaWAN 資源的動作和資料類型完整清單，請參閱 AWS IoT Wireless API 參考。

如何使用 AWS CLI 來新增目的地

您可以使用 AWS CLI 來新增目的地，方法為使用 create-destination 命令。以下範例顯示如何透過使用 `RuleName` 作為 `expression-type` 參數值而輸入值名稱，以建立目的地。如果要指定要發佈或訂閱訊息代理程式的主題名稱，請將 `expression-type` 參數值變更為 `MqttTopic`。

```
aws iotwireless create-destination \
```

```
    --name IoTWirelessDestination \
    --expression-type RuleName \
    --expression IoTWirelessRule \
    --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

執行此命令會使用指定的目的地名稱、規則名稱和角色名稱建立目的地。如需目的地之規則和角色名稱的相關資訊，請參閱 建立規則來處理 LoRaWAN 裝置訊息 和 為您的目的地建立 IAM 角色。

如需您可以使用哪些 CLI 的相關資訊，請參閱 AWS CLI 參考。

為您的目的地建立 IAM 角色

AWS IoT Core for LoRaWAN 目的地需要 IAM 角色，這些角色會將資料傳送至 AWS IoT 規則所需的許可給與 AWS IoT Core for LoRaWAN。如果尚未定義此類角色，則必須定義它，以便它顯示在角色清單中。

在使用主控台新增目的地時，AWS IoT Core for LoRaWAN 會自動為您建立 IAM 角色，如本主題先前所述。使用 API 或 CLI 新增目的地時，您必須為目的地建立 IAM 角色。

為 AWS IoT Core for LoRaWAN 目的地角色建立 IAM 政策

1. 開啟 IAM 主控台的政策中樞。
2. 選擇 Create policy (建立政策)，然後選擇 JSON 標籤。
3. 在編輯器中，刪除編輯器中的任何內容，然後貼上此政策文件。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeEndpoint",
                "iot:Publish"
            ],
            "Resource": "*"
        }
    ]
}
```

4. 選擇 Review Policy (檢閱政策)，然後在 Name (名稱) 中，輸入此政策的名稱。您將需要此名稱用於下一個程序中。

您也可以在 Description (描述) 中描述此政策 (如果您想要的話)。

5. 選擇 Create policy (建立政策)。

為 AWS IoT Core for LoRaWAN 目的地建立 IAM 角色

1. 開啟 IAM 主控台的角色中樞，然後選擇 Create role (建立角色)。
2. 在 Select type of trusted entity (選取信任的實體類型) 中，選擇 Another AWS 帳戶 (另一個 AWS 帳戶 帳戶)。
3. 在 Account ID (帳戶 ID) 中，請輸入您的 AWS 帳戶 ID，然後選擇 Next: Permissions (下一步：許可)。
4. 在搜尋方塊中，輸入您在上一個程序中建立的 IAM 政策名稱。
5. 在搜尋結果中，檢查您在上一個程序中建立的 IAM 政策。
6. 選擇 Next: Tags (下一步：標籤)，然後選擇 Next: Review (下一步：檢閱)。
7. 在 Role name (角色名稱) 中，輸入此角色的名稱，然後選擇 Create role (建立角色)。
8. 在確認訊息中，選擇您已建立的角色名稱來編輯新角色。
9. 在 Summary (摘要) 上，選擇 Trust relationships (信任關係) 標籤，然後選擇 Edit trust relationship (編輯信任關係)。
10. 在 Policy Document (政策文件) 中，變更 `Principal` 屬性以看起來像此範例。

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

在您變更 `Principal` 屬性之後，完整政策文件應該看起來像此範例。

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

```
  ]
}
```

11. 若要儲存您的變更，請選擇 Update Trust Policy (更新信任政策)。

定義此角色後，在設定 AWS IoT Core for LoRaWAN 目的地時，您可以在角色清單中找到它。

建立規則來處理 LoRaWAN 裝置訊息

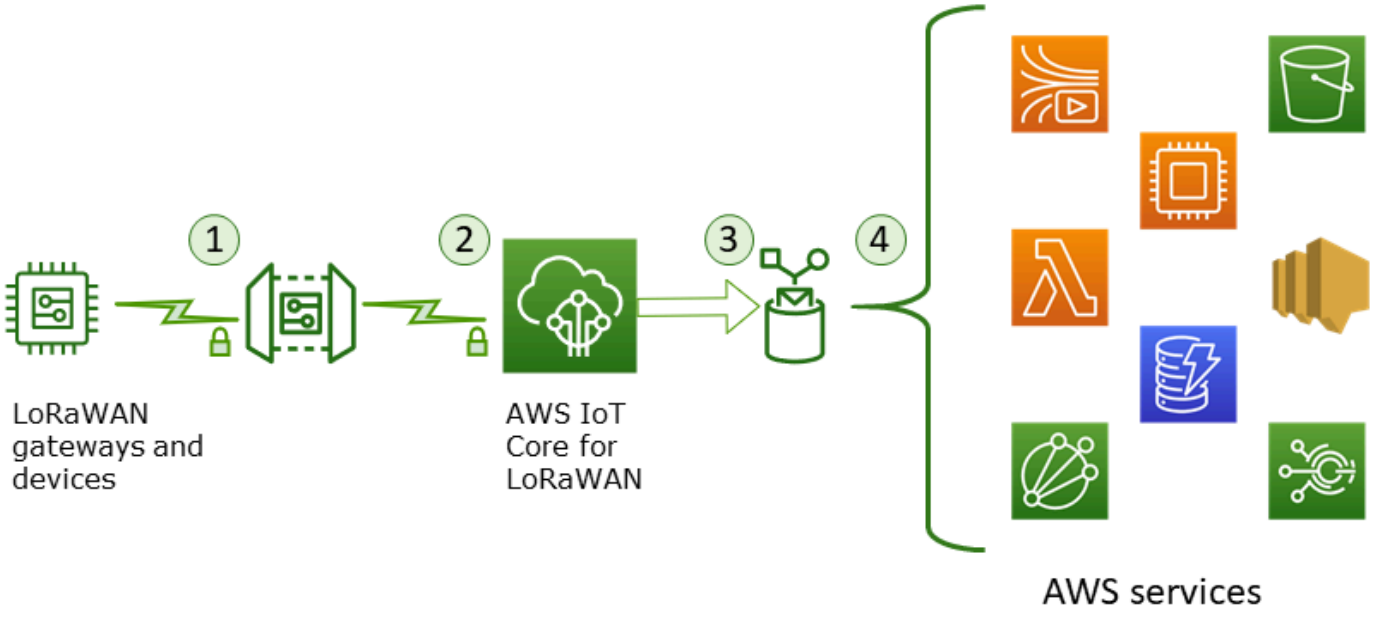
AWS IoT 規則會將裝置訊息傳送至其他服務。AWS IoT 規則也可以處理從 LoRaWAN 裝置接收的二進位訊息，將訊息轉換成其他格式，讓其他服務更容易使用。

AWS IoT Core for LoRaWAN 目的地會將無線裝置與規則建立關聯，而此規則會處理裝置訊息資料以傳送至其他服務。AWS IoT Core for LoRaWAN 一收到裝置的資料，此規則就會對其採取動作。AWS IoT Core for LoRaWAN 目的地可以由其訊息具有相同資料格式和將資料傳送至相同服務的所有裝置共用。

AWS IoT 規則如何處理裝置訊息

AWS IoT 規則如何處理裝置的訊息資料，取決於將接收資料的服務、裝置訊息資料的格式，以及服務所需的資料格式。通常，規則會呼叫 AWS Lambda 函式，將裝置的訊息資料轉換為服務所需的格式，然後將結果傳送至服務。

下圖顯示當訊息資料從無線裝置移至 AWS 服務時，如何保護和處理該訊息資料。



1. LoRaWAN 無線裝置會在傳輸二進位訊息之前，先使用 AES128 CTR 模式加密該訊息。

2. AWS IoT Core for LoRaWAN 會解密二進位訊息，並將解密的二進位訊息承載編碼為 base64 字串。
3. 產生的 base64 編碼訊息會作為訊息承載 (未格式化為 JSON 文件) 傳送至 AWS IoT 規則，此規則已在指派給裝置的目的地中加以描述。
4. AWS IoT 規則會將訊息資料導向至規則組態中所述的服務。

從無線裝置收到的加密二進位承載不會被 AWS IoT Core for LoRaWAN 變更或解譯。解密的二進位訊息承載只會編碼為 base64 字串。若要讓服務存取二進位訊息承載中的資料元素，規則呼叫的函式必須從承載剖析出資料元素。base64 編碼的訊息承載是一種 ASCII 字串，因此它可以儲存成這類形式，以供稍後解析。

建立 LoRaWAN 裝置的規則

AWS IoT Core for LoRaWAN 會使用 AWS IoT 規則，安全地將裝置訊息直接安全地傳送至其他 AWS 服務，而無需使用訊息代理程式。從擷取路徑移除訊息代理程式，可降低成本並最佳化資料流程。

若要讓 AWS IoT Core for LoRaWAN 將裝置訊息傳送至其他 AWS 服務，則需要 AWS IoT Core for LoRaWAN 目的地和指派給該目的地的 AWS IoT 規則。AWS IoT 規則必須包含 SQL 查詢陳述式和至少一個規則動作。

通常，AWS IoT 規則查詢陳述式包含：

- SQL SELECT 子句，用於選取並格式化訊息承載中的資料
- 主題篩選條件 (規則查詢陳述式中的 FROM 物件)，用來識別要使用的訊息
- 選用條件陳述式 (SQL WHERE 子句)，用來指定要對其採取行動的特定條件

以下是規則查詢陳述式的範例：

```
SELECT temperature FROM iot/topic' WHERE temperature > 50
```

當建置 AWS IoT 規則來處理來自 LoRaWAN 裝置的承載，您不必指定 FROM 子句，作為規則查詢物件的一部分。規則查詢語句必須具有 SQL SELECT 子句，並可選擇性地具有 WHERE 子句。如果查詢陳述式使用 FROM 子句，則會將其忽略。

以下範例為可以處理來自 LoRaWAN 裝置之承載的規則查詢陳述式：

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,
       WirelessMetadata.LoRaWAN.DevEui as DevEui,
```

```
        PayloadData
```

在此範例中，PayloadData 是您的 LoRaWAN 裝置所傳送並以 base64 編碼的二進位承載。

以下範例是規則查詢陳述式，它可以對傳入承載執行二進位解碼，並將其轉換為不同的格式，例如 JSON：

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,
       WirelessMetadata.LoRaWAN.DevEui as DevEui,
       aws_lambda("arn:aws:lambda:<region>:<account>:function:<name>",

                  {
                   "PayloadData":PayloadData,
                   "Fport": WirelessMetadata.LoRaWAN.FPort
                  }) as decodingoutput
```

如需使用 SELECT AND WHERE 子句的詳細資訊，請參閱 https://docs.aws.amazon.com/iot/latest/developerguide/iot-sql-reference.html。

如需有關 AWS IoT 規則以及如何建立和使用它們的詳細資訊，請參閱 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html 和 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules-tutorial.html。

如需建立和使用 AWS IoT Core for LoRaWAN 目的地的相關資訊，請參閱 新增目的地至 AWS IoT Core for LoRaWAN。

如需在規則中使用二進位訊息承載的相關資訊，請參閱 https://docs.aws.amazon.com/iot/latest/developerguide/binary-payloads.html。

如需有關資料安全性，以及在其旅程期間用來保護訊息承載之加密的詳細資訊，請參閱 AWS IoT Wireless 的資料保護。

如需顯示 IoT 規則的二進位解碼和實作範例的參考架構，請參閱 GitHub 上的 AWS IoT Core for LoRaWAN 解決方案範例。

連接您的 LoRaWAN 裝置並驗證其連線狀態

在檢查裝置連線狀態之前，您必須已新增裝置，並將其連接至 AWS IoT Core for LoRaWAN。如需如何新增裝置的相關資訊，請參閱 將您的無線裝置新增至 AWS IoT Core for LoRaWAN。

在新增了裝置之後，請參閱裝置的使用者手冊，了解如何起始從 LoRaWAN 裝置傳送上行訊息。

使用主控台檢查裝置連線狀態

若要使用主控台檢查連線狀態，請導覽至 AWS IoT 主控台的 Devices (裝置) 頁面，然後選擇您已新增的裝置。在無線裝置詳細資訊頁面的 Details (詳細資訊) 區段中，您會看到上次接收上行的日期和時間。

使用 API 檢查裝置連線狀態

若要使用 API 檢查連線狀態，請使用 `GetWirelessDeviceStatistics` API。此 API 沒有要求主體，而且只包含上次收到上行時顯示的回應主體。

```
HTTP/1.1 200
Content-type: application/json

{
  “LastUplinkReceivedAt”: “2021-03-24T23:13:08.476015749Z”,
  "LoRaWAN": {
        "DataRate": 5,
        "DevEui": "647fda0000006420",
        "Frequency": 868100000
        "Gateways": [
         {
            "GatewayEui": "c0ee40ffff29df10",
            "Rssi": -67,
            "Snr": 9.75
         }
      ],
  “WirelessDeviceId”: “30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
}
```

後續步驟

既然您已連接裝置並驗證連線狀態，您就可以在 AWS IoT 主控台的 Test (測試) 頁面上使用 MQTT 測試用戶端，來觀察從裝置收到之上行中繼資料的格式。如需詳細資訊，請參閱檢視從 LoRaWAN 裝置傳送的上行訊息格式。

使用 AWS IoT Core for LoRaWAN 設定無線資源的位置

在使用此功能之前，請注意為解析 LoRaWAN 裝置位置資訊而選擇的第三方供應商仰賴於International GNSS Service (IGS)、透過 NASA 運作的 EarthData 或由其他第三方提供或維護的資料摘要

和資料集。這些資料摘要和資料集屬於第三方內容 (如客戶協議所定義) 並且依現狀提供。如需詳細資訊，請參閱 AWS服務條款。

您可以使用 AWS IoT Core for LoRaWAN 指定靜態位置資料，或使用第三方求解器啟用定位功能以即時識別您的裝置位置。您可以新增或更新 LoRaWAN 裝置和/或閘道的位置資訊。

您可以在將裝置或閘道新增至 AWS IoT Core for LoRaWAN 時，或編輯裝置或閘道的組態詳細資訊時，指定位置資訊。系統會將位置資訊指定為 GeoJSON 承載。GeoJSON 格式可用於對地理資料結構進行編碼。承載含有裝置位置的緯度和經度坐標 (以世界大地坐標系統 (WGS84) 為準)。

在求解器運算出您資源的位置後，如果有 Amazon Location Service，您可以啟用 Amazon Location 地圖，其中會顯示資源的位置。使用位置資料，您可以：

- 啟用定位功能來識別並取得 LoRaWAN 裝置的位置。
- 追蹤並監控閘道和裝置的位置。
- 定義處理位置資料任何更新並將資料路由至其他 AWS 服務 的 AWS IoT 規則。如需規則動作的清單，請參閱《AWS IoT 開發人員指南》中的 AWS IoT 規則動作。
- 使用位置資料和 Amazon SNS 建立警示，並在發生任何異常活動時接收裝置的通知。

LoRaWAN 裝置定位功能的運作方式

您可以使用第三方 Wi-Fi 和 GNSS 求解器啟用定位功能以識別裝置的位置。您可以使用此資訊來追蹤和監控裝置。以下步驟說明如何啟用定位功能並檢視 LoRaWAN 裝置的位置資訊。

> **Note**
>
> 第三方求解器只能與具有 LoRa Edge 晶片的 LoRaWAN 裝置搭配使用。其不能與 LoRaWAN 閘道一起使用。對於閘道，您仍然可以指定靜態位置資訊，並在 Amazon Location 地圖上識別位置。

1. 新增裝置

   啟用定位功能之前，請先將裝置新增至 AWS IoT Core for LoRaWAN。LoRaWAN 裝置必須具有 LoRa Edge 晶片組，其是一個超低功耗地理位置平台，以地理位置應用為目標，整合了長距離 LoRa 收發器、多星系 GNSS 掃描器和被動式 Wi-Fi MAC 掃描器。

2. 啟用定位功能

   若要取得裝置的即時位置，請啟用定位功能。當 LoRaWAN 裝置傳送上行訊息時，訊息中包含的 Wi-Fi 和 GNSS 掃描資料會使用地理定位訊框連接埠傳送至 AWS IoT Core for LoRaWAN。

3. 擷取位置資訊

   根據來自收發器的掃描結果，從求解器擷取預估的裝置位置。如果同時使用 Wi-Fi 和 GNSS 掃描結果來計算位置資訊，AWS IoT Core for LoRaWAN 會選擇準確度較高的估計位置。

4. 檢視位置資訊

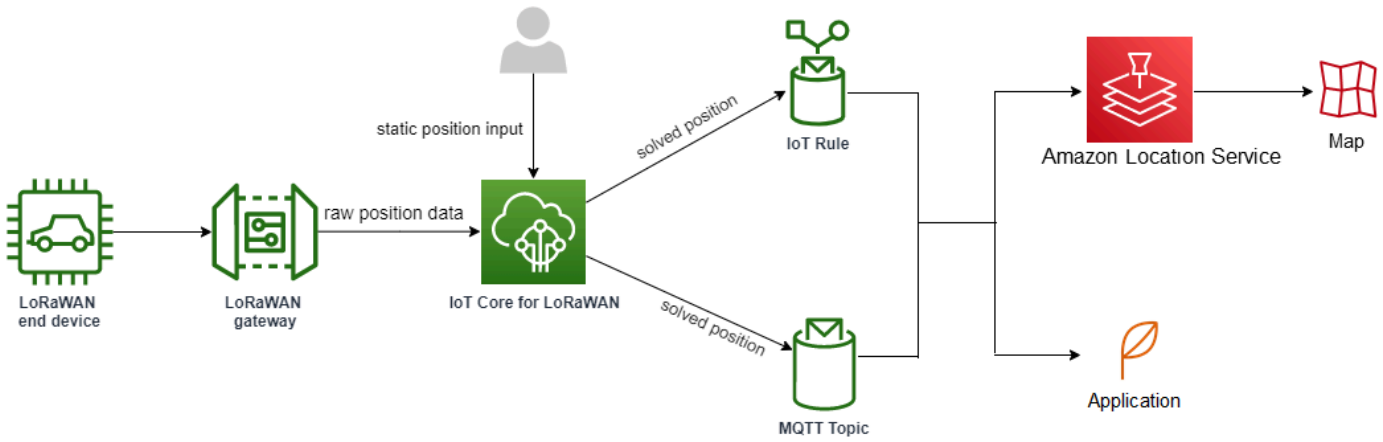
   求解器計算位置資訊之後，也會提供準確度資訊，其會指出求解器計算的位置與您輸入的靜態位置資訊間的差異。您也可以在 Amazon Location 地圖上檢視裝置位置。

> **Note**
>
> 由於求解器無法用於 LoRaWAN 閘道，準確度資訊將報告為 `0.0`。

如需上行訊息格式和用於定位求解器的頻率連接埠的詳細資訊，請參閱 AWS IoT Core for LoRaWAN 傳送至規則引擎的上行訊息。

定位工作流程概觀

下圖顯示 AWS IoT Core for LoRaWAN 如何儲存和更新您裝置和閘道的位置資訊。



1. 指定資源的靜態位置

使用經緯度座標將裝置或閘道的靜態位置資訊指定為 GeoJSON 承載。您也可以指定選用的高度座標。這些座標以 WGS84 座標系統為基礎。如需詳細資訊，請參閱世界大地坐標系統 (WGS84)。

2. 啟用裝置的定位功能

   如果您使用的 LoRaWAN 裝置具有 LoRa Edge 晶片，則可選擇啟用定位功能來即時追蹤裝置位置。當您的裝置傳送上行訊息時，GNSS 和 Wi-Fi 掃描資料會使用地理定位訊框連接埠傳送至 AWS IoT Core for LoRaWAN。然後，求解器會使用此資訊來解析裝置位置。

3. 新增路由位置資料的目的地

   您可以新增描述用於處理裝置資料之 IoT 規則的目的地，並將更新後的位置資訊路由到 AWS IoT Core for LoRaWAN。您也可以在 Amazon Location 地圖上檢視資源的上一個已知位置。

設定資源位置

您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI 來設定資源的位置。

如果裝置具有 LoRa Edge 晶片，您還可以啟用定位功能來計算即時位置資訊。針對閘道應用，您仍然可以輸入靜態位置座標，並使用 Amazon Location 在 Amazon Location 地圖上追蹤閘道位置。

主題

設定 LoRaWAN 閘道的位置

您將閘道新增至 AWS IoT Core for LoRaWAN 時，可以指定靜態位置資料。如果您已啟用 Amazon Location Service，位置資料會顯示在 Amazon Location 地圖上。

> **Note**
>
> 第三方的求解器不能與 LoRaWAN 閘道搭配使用。針對閘道應用，您仍然可以指定靜態位置座標並新增目的地。當求解器不用於計算位置時 (例如在閘道應用中)，準確度資訊將報告為 `0.0`。

您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI 來設定閘道位置。

使用主控台設定閘道的位置

若要使用 AWS Management Console 來設定閘道資源的位置，請先登入主控台，然後前往 AWS IoT 主控台的 Gateways (閘道) 中樞頁面。

新增位置資訊

為閘道新增位置組態

1. 在 Gateways (閘道) 中樞頁面，選擇 Add gateway (新增閘道)。
2. 輸入閘道的 EUI、頻帶 (RFRegion)，以及任何其他閘道詳細資訊和 LoRaWAN 組態資訊。如需詳細資訊，請參閱使用主控台新增閘道。
3. 前往Position information - Optional (位置資訊 - 選擇性)，使用經緯度坐標以及高度座標 (選用) 來輸入閘道的位置資訊。位置資訊以 WGS84 座標系統為基礎。

檢視閘道的位置

設定閘道的位置後，AWS IoT Core for LoRaWAN 會建立名為 `iotwireless.map` 的 Amazon Location 地圖。您可以在 Position (位置) 索引標籤中的閘道詳細資訊頁面上查看此地圖。根據您指定的位置座標，閘道位置會在地圖上顯示為標記。您可以放大或縮小標記，清楚檢視閘道在地圖上的位置。在 Position (位置) 索引標籤中，您也可看到準確度資訊和決定閘道位置的時間戳記

> **Note**
>
> 如果沒有安裝 Amazon Location Service 地圖，您會看到一則訊息，指明您必須使用 Amazon Location Service 來存取地圖和檢視閘道位置。使用 Amazon Location Service 地圖可能會向您的 AWS 帳戶 收取額外費用。如需詳細資訊，請參閱 AWS IoT Core 定價。

地圖 `iotwireless.map` 作為地圖資料的來源，透過 `Get` API 操作存取，例如 `GetMapTile`。如需 `Get` API 與地圖搭配使用的相關資訊，請參閱 Amazon Location Service API reference (《Amazon Location Service API 參考》)。

如需有關此地圖的其他詳細資訊，請前往 Amazon Location Service 主控台，選擇 maps (地圖)，然後選擇 iotwireless.map。如需詳細資訊，請參閱《Amazon Location Service 開發人員指南》中的地圖。

更新閘道的位置組態

若要變更閘道的位置組態，請在閘道詳細資訊頁面中選擇 Edit (編輯)，然後更新位置資訊和目的地。

> **Note**
>
> 無法取得有關歷史位置資料的資訊。更新閘道的位置座標時，即會覆寫先前報告的位置資料。更新位置之後，您會在閘道詳細資訊的 Position (位置) 索引標籤中看到新的位置資訊。時間戳記的變更表示它對應到閘道的最後一個已知位置。

使用 API 設定閘道的位置

您可以使用 AWS IoT Wireless API 或 AWS CLI 指定位置資訊並設定閘道位置。

> **Important**
>
> 不再支援下列 API 動作：UpdatePosition、GetPosition、PutPositionConfiguration、GetPositionConfiguration 以及 ListPositionConfigurations。用於更新和擷取位置資訊的呼叫應改為使用 GetResourcePosition 和 UpdateResourcePosition API 操作。

新增位置資訊

若要為指定的無線閘道新增靜態位置資訊，請使用 UpdateResourcePosition API 操作或 update-resource-position CLI 命令來指定坐標。指定 `WirelessGateway` 作為 `ResourceType`、要更新之無線閘道的 ID 作為 `ResourceIdentifier`，以及定位資訊作為 GeoJSON 承載。

```
aws iotwireless update-resource-position \
    --resource-type WirelessGateway \
    --resource-id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --cli-input-json file://gatewayposition.json
```

下列顯示 *gatewayposition.json* 檔案的內容。

gatewayposition.json 的內容

```
{
     "type": "Point",
     "coordinates": [33.3318, -22.2155, 13.123],
     "properties": {
          "timestamp": "2018-11-30T18:35:24Z"
      }
```

```
}
```

執行這個命令不會產生任何輸出。若要查看您指定的位置資訊，請使用 `GetResourcePosition` API 操作。

取得位置資訊

若要取得指定無線閘道的位置資訊，請使用 GetResourcePosition API 操作或 get-resource-position CLI 命令。指定 `WirelessGateway` 作為 `resourceType`，並提供無線閘道的 ID 作為 `resourceIdentifier`。

```
aws iotwireless get-resource-position \
    --resource-type WirelessGateway \
    --resource-id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

執行這個命令會將無線閘道位置資訊顯示為 GeoJSON 承載。您將看到的資訊包含位置坐標、位置資訊類型、其他屬性，例如對應於閘道最後已知位置的時間戳記。

```
{
    {
    "type": "Point",
     "coordinates": [33.3318,-22.2155,13.123],
     "properties": {
         "timestamp": "2018-11-30T18:35:24Z"
         }
    }
}
```

設定 LoRaWAN 裝置的位置

您將裝置新增至 AWS IoT Core for LoRaWAN 時，可以指定靜態位置資訊、選擇性啟用定位，以及指定目的地。目的地描述了處理裝置位置資訊並將更新後的位置路由至 Amazon Location Service 的 IoT 規則。設定裝置位置後，位置資料會顯示在 Amazon Location 地圖上，其中包含準確度資訊和您指定的目的地。

您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI 來設定裝置的位置。

訊框連接埠和上行訊息格式

如果啟用定位，您必須指定地理位置訊框連接埠，以便將 Wi-Fi 和 GNSS 掃描資料從裝置傳送到 AWS IoT Core for LoRaWAN。位置資訊會使用此訊框連接埠傳送給 AWS IoT Core for LoRaWAN。

LoRaWAN 規格提供資料交付欄位 (FRMPayload) 和連接埠欄位 (FPort)，以區分不同類型的訊息。若要傳送位置資訊，您可以為訊框連接埠指定 1 到 223 之間的任意值。FPort 0 會保留給 MAC 訊息使用，FPort 224 會保留給 MAC 相容性測試使用，連接埠 225-255 則保留給未來的標準化應用程式延伸模組。

AWS IoT Core for LoRaWAN 傳送至規則引擎的上行訊息

新增目的地時，則會建立 AWS IoT 規則，以使用規則引擎將資料路由到 Amazon Location Service。更新後的位置資訊隨即會顯示在 Amazon Location 地圖上。如果尚未啟用定位，目的地會在您更新裝置的靜態位置座標時路由位置資料。

下列程式碼顯示從 AWS IoT Core for LoRaWAN 傳送的上行訊息格式，並提供位置資訊、準確度、求解器組態和無線中繼資料。下面反白顯示的欄位為選用項目。如果沒有垂直準確度資訊，則值為 `null`。

```
{
    // Position configuration parameters for given wireless device
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",

    // Position information for a device in GeoJSON format. Altitude
    // is optional. If no vertical accuracy information is available
    // or positioning isn't activated, the value is set to null.
    // The position information coordinates are listed in the order
    // [longitude, latitude, altitude].
    "coordinates": [33.33000183105469, -22.219999313354492, 99.0],
    "type": "Point",
    "properties": {
         "horizontalAccuracy": number,
         "verticalAccuracy": number",
         "timestamp": "2022-08-19T03:08:35.061Z"
    },

    //Parameters controlled by AWS IoT Core for LoRaWAN
    "WirelessMetadata":
    {
        "LoRaWAN":
        {
            "ADR": false,
            "Bandwidth": 125,
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "0",
            "DevAddr": "00b96cd4",
```

```
            "DevEui": "58a0cb000202c99",
            "FOptLen": 2,
            "FCnt": 1,
            "Fport": 136,
            "Frequency": "868100000",
            "Gateways": [
             {
                    "GatewayEui": "80029cfffe5cf1cc",
                    "Snr": -29,
                    "Rssi": 9.75
             }
             ],
            "MIC": "7255cb07",
            "MType": "UnconfirmedDataUp",
            "Major": "LoRaWANR1",
            "Modulation": "LORA",
            "PolarizationInversion": false,
            "SpreadingFactor": 12,
            "Timestamp": "2021-05-03T03:24:29Z"

        }
    }
}
```

使用主控台設定裝置的位置

若要使用 AWS Management Console 設定並管理裝置的位置，請先登入主控台，然後前往 AWS IoT 主控台的 Devices (裝置) 中樞頁面。

新增位置資訊

若要為裝置新增位置的位置資訊：

1. 在 Device (裝置) 中樞頁面，選擇 Add wireless device (新增無線裝置)。
2. 輸入無線裝置規格、裝置和服務設定檔，以及定義將資料路由到其他 AWS 服務 的 IoT 規則的目的地。如需詳細資訊，請參閱將裝置加入 AWS IoT Core for LoRaWAN。
3. 輸入位置資訊、選擇性啟用地理位置，並指定要用來路由訊息的位置資料目的地。
   - 位置資訊

     使用經緯度座標以及高度座標 (選用) 來指定裝置的位置資料。位置資訊以 WGS84 座標系統為基礎。

- GeoLocation

  如果您希望 AWS IoT Core for LoRaWAN 使用地理位置來計算裝置，請啟用定位功能。其使用第三方 GNSS 和 Wi-Fi 求解器即時識別您裝置的位置。

  若要輸入地理位置資訊，請選擇啟用定位，然後輸入地理位置訊框連接埠，以便將 GNSS 和 Wi-Fi 掃描資料傳送至 AWS IoT Core for LoRaWAN。您會看到預設填入的 FPort 供您參考。不過，您可以選擇 1 和 223 之間的任意值。

- 位置資料目的地

  選擇一個目的地來描述處理裝置位置資料並將資料轉送至 AWS IoT 的 AWS IoT Core for LoRaWAN 規則。僅使用此目的地來路由位置資料。該目的地必須與用於將裝置資料路由到其他 AWS 服務 的目的地不同。

檢視裝置的位置組態

設定裝置的位置後，AWS IoT Core for LoRaWAN 會建立名為 `iotwireless.map` 的 Amazon Location 地圖。您可以在 Position (位置) 索引標籤中的裝置詳細資訊頁面上查看此地圖。根據您指定的位置座標或第三方求解器計算的位置，裝置位置會在地圖上顯示為標記。您可以放大或縮小標記，清楚檢視裝置在地圖上的位置。在 Position (位置) 索引標籤中的裝置詳細資訊頁面上，您也可看到準確度資訊、決定裝置位置的時間戳記，以及您指定的位置資料目的地。

> **Note**
>
> 如果尚未啟用 Amazon Location Service 地圖，您會看到一則訊息，指明您必須使用 Amazon Location Service 來存取地圖和檢視位置。使用 Amazon Location Service 地圖可能會向您的 AWS 帳戶 收取額外費用。如需詳細資訊，請參閱 AWS IoT Core 定價。

地圖 `iotwireless.map` 作為地圖資料的來源，透過 `Get` API 操作存取，例如 `GetMapTile`。如需 `Get` API 與地圖搭配使用的相關資訊，請參閱 Amazon Location Service API reference (《Amazon Location Service API 參考》)。

如需有關此地圖的其他詳細資訊，請前往 Amazon Location Service 主控台，選擇 maps (地圖)，然後選擇 iotwireless.map。如需詳細資訊，請參閱《Amazon Location Service 開發人員指南》中的地圖。

更新裝置的位置組態

若要變更裝置的位置組態，請在裝置詳細資訊頁面中選擇 Edit (編輯)，然後更新位置資訊、任何地理位置設定以及目的地。

> **Note**
>
> 無法取得有關歷史位置資料的資訊。如果更新裝置的位置座標，即會覆寫先前報告的位置資料。更新位置之後，您會在 Position (位置) 索引標籤中的裝置詳細資訊頁面上看到新的位置資訊。時間戳記的變更表示它對應到裝置的最後已知位置。

使用 API 設定裝置位置

您可以使用 AWS IoT Wireless API 或 AWS CLI 來指定位置資訊、設定裝置位置，以及啟用選用的地理位置。

> **Important**
>
> 不再支援下列 API 動作：
> UpdatePosition、GetPosition、PutPositionConfiguration、GetPositionConfiguration 以及 ListPositionConfigurations。用於更新和擷取位置資訊的呼叫應改為使用 GetResourcePosition 和 UpdateResourcePosition API 操作。

新增位置資訊和組態

若要為指定的無線裝置新增位置資訊，請使用 UpdateResourcePosition API 操作或 update-resource-position CLI 命令來指定座標。指定 `WirelessDevice` 作為 `ResourceType`、要更新之無線閘道的 ID 作為 `ResourceIdentifier`，以及定位資訊作為 GeoJSON 承載。

```
aws iotwireless update-resource-position \
    --resource-type WirelessDevice \
    --resource-id "1ffd32c8-8130-4194-96df-622f072a315f" \
    --position [33.33, -33.33, 10.0]
```

下列顯示 *deviceposition.json* 檔案的內容。若要指定用於傳送地理位置資料的 FPort 值，請搭配 CreateWirelessDevice 和 UpdateWirelessDevice API 操作來使用定位物件。

deviceposition.json 的內容

```
{
```

```
     "type": "Point",
     "coordinates": [33.3318, -22.2155, 13.123],
     "properties": {
          "verticalAccuracy": 707,
          "horizontalAccuracy":
          "timestamp": "2018-11-30T18:35:24Z"
      }
}
```

執行這個命令不會產生任何輸出。若要查看您指定的位置資訊，請使用 GetResourcePosition API 操作。

取得位置資訊和組態

若要取得指定無線裝置的位置資訊，請使用 GetResourcePosition API 或 get-resource-position CLI 命令。指定 WirelessDevice 作為 resourceType，並提供作為 resourceIdentifier 的無線裝置 ID。

```
aws iotwireless get-resource-position \
    --resource-type WirelessDevice \
    --resource-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

執行這個命令會將無線裝置位置資訊顯示為 GeoJSON 承載。您將看到的資訊包含位置坐標、位置類型以及若干屬性，其可能包括對應於裝置最後已知位置的準確度及時間戳記資訊。

```
{
     "type": "Point",
     "coordinates": [33.3318, -22.2155, 13.123],
     "properties": {
          "verticalAccuracy": 707,
          "horizontalAccuracy": 389,
          "horizontalConfidenceLevel": 0.68,
          "verticalConfidenceLevel": 0.68,
          "timestamp": "2018-11-30T18:35:24Z"
      }
}
```

使用 AWS IoT Wireless 管理閘道

下列是搭配使用閘道與 AWS IoT Core for LoRaWAN 時的一些重要考量。如需如何新增閘道至 AWS IoT Core for LoRaWAN 的相關資訊，請參閱 將閘道加入 AWS IoT Core for LoRaWAN。

LoRa Basics Station 軟體需求

若要連接到 AWS IoT Core for LoRaWAN，您的 LoRaWAN 閘道須有名為 LoRa Basics Station 的軟體在其上執行。LoRa Basics Station 是由 Semtech Corporation 維護的開放原始碼軟體，並由其 GitHub 儲存庫散發。AWS IoT Core for LoRaWAN 支援 LoRa Basics Station 2.0.4 版及更新版本。最新版本為 2.0.6。

使用來自 AWS Partner Device Catalog 的合格閘道

AWS Partner Device Catalog 包含有資格與 AWS IoT Core for LoRaWAN 搭配使用的閘道和開發人員套件。我們建議您使用這些合格的閘道，因為您不必修改內嵌軟體，即可將閘道連接至 AWS IoT Core。這些閘道已有與 AWS IoT Core for LoRaWAN 相容的 BasicStation 軟體版本。

> **Note**
>
> 如果您有未列示在合作夥伴目錄中的閘道，可作為搭配 AWS IoT Core for LoRaWAN 的合格閘道，則若該閘道正在執行版本為 2.0.4 及更新版本的 LoRa Basics Station 軟體，您可能仍然可以使用它。確定您使用 TLS Server and Client Authentication (TLS 伺服器和用戶端身分驗證)，以驗證您的 LoRaWAN 閘道。

使用 CUPS 和 LNS 通訊協定

LoRa Basics Station 軟體包含兩個子通訊協定，用於將閘道連接到網路伺服器、LoRaWAN 網路伺服器 (LNS)，以及組態與更新伺服器 (CUPS) 通訊協定。

LNS 通訊協定會在 LoRa Basics Station 相容閘道與網路伺服器之間建立資料連線。LoRa 上行和下行訊息會經由安全 WebSockets 透過資料連線進行交換。

CUPS 通訊協定可啟用憑證管理，以及閘道的遠端組態和韌體更新。AWS IoT Core for LoRaWAN 會提供 LNS 和 CUPS 端點，分別用於 LoRaWAN 資料擷取和遠端閘道管理。

如需詳細資訊，請參閱 LNS 通訊協定和 CUPS 通訊協定。

主題

- 設定 LoRaWAN 閘道的信標和篩選功能
- 使用 CUPS 服務搭配 AWS IoT Core for LoRaWAN 來更新閘道韌體
- 選擇閘道以接收 LoRaWAN 下行資料流量

設定 LoRaWAN 閘道的信標和篩選功能

使用 LoRaWAN 裝置時，您可以為 LoRaWAN 閘道設定某些選用參數。參數包括：

- 信標

  您可以為 LoRaWAN 閘道設定信標參數，以做為 B 類 LoRaWAN 裝置的橋接器。這些裝置會在排定的時段接收下行訊息，因此您必須為閘道設定信標參數，才能傳輸這些時間同步化的信標。
- 篩選

  您可以設定LoRaWAN 閘道的 `NetID` 和 `JoinEUI` 參數，用於篩選裝置資料流量。篩選流量有助於節省頻寬使用量，並減少閘道與 LNS 之間的流量。
- 子頻帶

  您可以設定閘道的子頻帶，以指定要使用的特定子頻帶。對於無法在各個子頻帶之間跳躍的無線裝置，您可以使用此功能，僅使用該特定子頻帶中的頻率通道來與裝置進行通訊。

下列主題包含有關這些參數及如何設定這些參數的詳細資訊。信標參數在AWS Management Console中不可用，只能使用 AWS IoT Wireless API 或 AWS CLI 指定。

主題

設定閘道傳送信標至 B 類裝置

如果您將 B 類無線裝置加入 AWS IoT Core for LoRaWAN，裝置會在排定的時段接收下行訊息。裝置會根據閘道傳輸的時間同步信標來開啟這些插槽。為了讓您的閘道傳輸這些時間同步信標，您可以使用 AWS IoT Core for LoRaWAN 設定閘道的某些信標相關參數。

若要設定這些信標參數，您的閘道必須執行 LoRa Basics Station 2.0.6 版軟體。請參閱使用來自 AWS Partner Device Catalog 的合格閘道。

如何設定信標參數

> **Note**
>
> 如果閘道與 B 類無線裝置通訊，您才需設定閘道的信標參數。

在使用 CreateWirelessGateway API 操作將閘道新增至 AWS IoT Core for LoRaWAN 時設定信標參數。呼叫 API 操作時，使用閘道的 Beaconing 物件指定下列參數。設定參數後，閘道會以 128 秒的間隔將信標傳送到您的裝置。

- DataRate：傳輸信標之閘道的資料速率。
- Frequencies：閘道傳輸信標的頻率清單。

下列範例說明如何設定閘道的這些參數。input.json 檔案也會包含其他詳細資訊，例如閘道憑證和佈建憑證。如需使用 CreateWirelessGateway API 操作新增閘道至 AWS IoT Core for LoRaWAN 的詳細資訊，請參閱 使用 API 新增閘道。

> **Note**
>
> 使用 AWS IoT 主控台將閘道新增至 AWS IoT Core for LoRaWAN 時無法使用信標參數。

```
aws iotwireless create-wireless-gateway \
    --name "myLoRaWANGateway" \
    --cli-input-json file://input.json
```

下列顯示 input.json 檔案的內容。

input.json 的內容

```
{
    "Description": "My LoRaWAN gateway",
    "LoRaWAN": {
        "Beaconing": {
          "DataRate": 8,
          "Frequencies": ["923300000","923900000"]
        },
        "GatewayEui": "a1b2c3d4567890ab",
        "RfRegion": US915,
        "JoinEuiFilters": [
         ["0000000000000001", "00000000000000ff"],
         ["000000000000ff00", "000000000000ffff"]
         ],
        "NetIdFilters": ["000000", "000001"],
        "RfRegion": "US915",
        "SubBands": [2]
```

```
    }
}
```

以下程式碼顯示執行這個命令的範例輸出。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:400232685877aa:WirelessGateway/a01b2c34-d44e-567f-abcd-0123e445663a",
    "Id": a01b2c34-d44e-567f-abcd-0123e445663a"
}
```

取得信標參數的資訊

您可以使用 GetWirelessGateway API 操作以取得閘道的信標參數相關資訊。

> **Note**
>
> 如果已經加入閘道，則無法使用 UpdateWirelessGateway API 操作來設定信標參數。若要設定參數，您必須刪除閘道，然後在使用 CreateWirelessGateway API 操作新增閘道時指定參數。

```
aws iotwireless get-wireless-gateway \
    --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --identifier-type WirelessGatewayId
```

執行此命令會傳回閘道和信標參數的相關資訊。

設定閘道的子頻帶和篩選功能

LoRaWAN 閘道執行 LoRa Basics Station 軟體，讓閘道可以連接至 AWS IoT Core for LoRaWAN。若要連接至 AWS IoT Core for LoRaWAN，您的 LoRa 閘道首先會查詢 CUPS 伺服器的 LNS 端點，然後建立 WebSockets 資料與該端點的連線。在建立連線之後，可以透過該連線交換上行和下行框架。

篩選閘道收到的 LoRa 資料框架

在 LoRaWAN 閘道建立與端點的連線之後，AWS IoT Core for LoRaWAN 會以 router_config 訊息回應，指定 LoRa 閘道組態的一組參數，包括篩選參數 NetID 和 JoinEui。如需 router_config 以及如何與 LoRaWAN 網路伺服器 (LNS) 建立連線的詳細資訊，請參閱 LNS 通訊協定。

```
{
"msgtype"    : "router_config"
"NetID"      : [ INT, .. ]
"JoinEui"    : [ [INT,INT], .. ] // ranges: beg,end inclusive
"region"     : STRING             // e.g. "EU863", "US902", ..
"hwspec"     : STRING
"freq_range" : [ INT, INT ]       // min, max (hz)
"DRs"        : [ [INT,INT,INT], .. ]   // sf,bw,dnonly
"sx1301_conf": [ SX1301CONF, .. ]
"nocca"      : BOOL
"nodc"       : BOOL
"nodwell"    : BOOL
}
```

閘道通常透過 Wi-Fi、乙太網路或行動網路之類的高頻寬網路，將 LoRaWAN 裝置資料帶入和帶出 LNS。閘道通常會接收所有訊息，並將到達其中的流量傳遞給 AWS IoT Core for LoRaWAN。不過，您可以設定閘道來篩選某些裝置資料流量，這有助於節省頻寬使用量，並減少閘道與 LNS 之間的流量。

若要設定 LoRa 閘道以篩選資料框架，您可以在 `router_config` 訊息中使用參數 `NetID` 和 `JoinEui`。`NetID` 是接受的 NetID 值清單。任何 LoRa 資料框架若攜帶不是列出的資料框架，將遭到捨棄。`JoinEui` 是 JoinEUI 值的整數值編碼範圍的配對清單。聯結請求框架將由閘道捨棄，除非訊息中的欄位 `JoinEui` 在範圍 [BegEui,EndEui] 內。

頻率通道和子頻帶

對於 US915 和 AU915 RF 區域，無線裝置有 64 個 125KHz 和 8 個 500KHz 上行通道的選擇，以使用 LoRa 閘道存取 LoRaWAN 網路。上行頻率通道分為 8 個子頻帶，每個頻帶有 8 個 125KHz 通道和一個 500KHz 通道。對於 AU915 區域中的每個一般閘道，將支援一或多個子頻帶。

某些無線裝置無法在子頻帶之間跳躍，而且當連接到 AWS IoT Core for LoRaWAN 時，只能在一個子頻帶上使用頻率通道。對於要從這些裝置傳輸的上行封包，將 LoRa 閘道設定為使用該特定子頻帶。對於其他 RF 區域 (例如 EU868) 的閘道，不需要此組態。

使用主控台將您的閘道設定為使用篩選和子頻帶

您可以將閘道設定為使用特定的子頻帶，也可以啟用篩選 LoRa 資料框架的功能。若要使用主控台指定這些參數：

1. 導覽至 AWS IoT 主控台的 AWS IoT Core for LoRaWAN Gateways (閘道) 頁面，然後選擇 Add gateway (新增閘道)。

2. 指定閘道詳細資訊，例如 Gateway's Eui (閘道 Eui)、Frequency band (RFRegion) (頻帶 (RFRegion))，以及選用的 Name (名稱) 和 Description (描述)，然後選擇是否要將 AWS IoT 物件與您的閘道建立關聯。如需如何新增閘道的詳細資訊，請參閱 使用主控台新增閘道。
3. 在 LoRaWAN configuration (LoRaWAN 組態) 區段中，您可以指定子頻帶和篩選資訊。
   - `SubBands`：若要新增子頻帶，請選擇 Add SubBand (新增子頻帶)，並指定整數值清單，指出閘道支援哪些子頻帶。`SubBands` 參數只能在 `RfRegion` US915 和 AU915 中設定，而且在其中一個支援的區域內，必須具有範圍 `[1,8]` 中的值。
   - `NetIdFilters`：若要篩選上行框架，請選擇 Add NetID (新增 NetID)，然後指定閘道使用的字串值清單。來自無線裝置之傳入上行框架的 NetID 必須符合至少一個列出的值，否則框架會遭到捨棄。
   - `JoinEuiFilters`：選擇 Add JoinEui range (新增 JoinEui 範圍)，並指定閘道用來篩選 LoRa 框架的字串值組清單。指定為來自無線裝置之聯結請求一部分的 JoinEUI 值必須在至少其中一個 JoinEuiRange 值的範圍內，每個 JoinEuiRange 值都會列為一對 [BegEui, EndEui]，否則框架會遭到捨棄。
4. 然後，您可以繼續遵循 使用主控台新增閘道 中所述的指示來設定閘道。

在新增了閘道之後，如果您在 AWS IoT 主控台的 AWS IoT Core for LoRaWAN Gateways (閘道) 頁面中選取已新增的閘道，則可以在 Gateway details (閘道詳細資訊) 頁面的 LoRaWAN specific details (LoRaWAN 特定詳細資訊) 區段中看到 `SubBands`，以及篩選條件 `NetIdFilters` 和 `JoinEuiFilters`。

使用 API 將您的閘道設定為使用篩選和子頻帶

您可以使用您用來建立閘道的 CreateWirelessGateway API，以設定您想要使用並啟用篩選功能的子頻帶。使用 `CreateWirelessGateway` API，您可以指定子頻帶和篩選條件，作為您使用 `LoRaWAN` 欄位所提供之閘道組態資訊的一部分。下列顯示了包含此資訊的請求字符。

```
POST /wireless-gateways HTTP/1.1
Content-type: application/json

{
"Arn": "arn:aws:iotwireless:us-east-1:400232685877aa:WirelessGateway/
       a11e3d21-e44c-471c-afca-6716c228336a",
"Description": "Using my first LoRaWAN gateway",
   "LoRaWAN": {
      "GatewayEui": "a1b2c3d4567890ab",
      "JoinEuiFilters": [
        ["0000000000000001", "00000000000000ff"],
```

```
        ["000000000000ff00", "000000000000ffff"]
      ],
      "NetIdFilters": ["000000", "000001"],
      "RfRegion": "US915",
      "SubBands": [2]
   },
   "Name": "myFirstLoRaWANGateway"
   "ThingArn": null,
   "ThingName": null
}
```

您也可以使用 UpdateWirelessGateway API 來更新過濾條件，但不會更新子頻帶。如果 `JoinEuiFilters` 和 `NetIdfilters` 值是空值，這表示欄位沒有更新。如果這些值不是空值，並且包含空清單，則會套用更新。若要取得您指定的欄位值，請使用 GetWirelessGateway API。

使用 CUPS 服務搭配 AWS IoT Core for LoRaWAN 來更新閘道韌體

在閘道上執行的 LoRa Basics Station 軟體會使用組態與更新伺服器 (CUPS) 通訊協定，提供憑證管理和韌體更新界面。CUPS 通訊協定提供安全韌體更新交付與 ECDSA 簽章。

您必須經常更新閘道的韌體。您可以使用 CUPS 服務搭配 AWS IoT Core for LoRaWAN，為閘道提供韌體更新，而這些更新也可以在此閘道中進行簽署。若要更新閘道的韌體，您可以使用 SDK 或 CLI，但不能使用主控台。

更新程序約需 45 分鐘才能完成。如果您是第一次設定閘道來連接至 AWS IoT Core for LoRaWAN，可能需要更長的時間。閘道製造商通常會提供自己的韌體更新檔案和簽章，因此您可以改用這些檔案和簽章，然後繼續進行 將韌體檔案上傳至 S3 儲存貯體並新增 IAM 角色。

如果您沒有韌體更新檔案，請參閱產生韌體更新檔案和簽章，以取得您可以用來改編以適應您應用程式的範例。

若要執行閘道的韌體更新：

- 產生韌體更新檔案和簽章
- 將韌體檔案上傳至 S3 儲存貯體並新增 IAM 角色
- 使用任務定義來排程和執行韌體更新

產生韌體更新檔案和簽章

此程序中的步驟為選用步驟，並取決於您正在使用的閘道。閘道製造商會以更新檔案或指令碼的形式提供自己的韌體更新，而且 Basics Station 會在背景中執行此指令碼。在此情況下，您很可能會在所使用

閘道的版本備註中找到韌體更新檔案。然後，您可以改用該更新檔案或指令碼，然後繼續進行 將韌體檔案上傳至 S3 儲存貯體並新增 IAM 角色。

如果您沒有此指令碼，下列顯示為了產生韌體更新檔案而執行的命令。這些更新也可以進行簽署，以確保程式碼不會遭到變更或損毀，而且裝置會執行只由信任的作者發佈的程式碼。

在此程序中，您將：

- 產生韌體更新檔案
- 產生韌體更新的簽章
- 檢閱後續步驟

產生韌體更新檔案

在閘道上執行的 LoRa Basics Station 軟體能夠在 CUPS 回應中接收韌體更新。如果您沒有由製造商提供的指令碼，請參閱下列韌體更新指令碼，這是針對 Raspberry Pi 型 RAKWireless 閘道所撰寫的。我們有一個基礎指令碼，而且新的工作站二進位檔、版本檔案，以及 `station.conf` 會附加至其中。

> **Note**
>
> 此指令碼是 RAKWireless 閘道特有的，因此您必須改編它，以適應您的應用程序，這取決於您正在使用的閘道。

基礎指令碼

下列顯示 Raspberry Pi 型 RAKWireless 閘道的範例基礎指令碼。您可以將下列命令儲存在檔案 `base.sh` 中，然後在 Raspberry Pi 網頁瀏覽器的終端機中執行指令碼。

```
*#!/bin/bash*
execution_folder=/home/pi/Documents/basicstation/examples/aws_lorawan
station_path="$execution_folder/station"
version_path="$execution_folder/version.txt"
station_conf_path="$execution_folder/station_conf"

# Function to find the Basics Station binary at the end of this script
# and store it in the station path
function prepare_station()
{
 match=$(grep --text --line-number '^STATION:$' $0 | cut -d ':' -f 1)
 payload_start=$((match + 1))
```

```
 match_end=$(grep --text --line-number '^END_STATION:$' $0 | cut -d ':' -f 1)
 payload_end=$((match_end - 1))
 lines=$(($payload_end-$payload_start+1))
 head -n $payload_end $0 | tail -n $lines  > $station_path
}

# Function to find the version.txt at the end of this script
# and store it in the location for version.txt
function prepare_version()
{
  match=$(grep --text --line-number '^VERSION:$' $0 | cut -d ':' -f 1)
  payload_start=$((match + 1))
  match_end=$(grep --text --line-number '^END_VERSION:$' $0 | cut -d ':' -f 1)
  payload_end=$((match_end - 1))
  lines=$(($payload_end-$payload_start+1))
  head -n $payload_end $0 | tail -n $lines  > $version_path
}

# Function to find the version.txt at the end of this script
# and store it in the location for version.txt
function prepare_station_conf()
{
 match=$(grep --text --line-number '^CONF:$' $0 | cut -d ':' -f 1)
 payload_start=$((match + 1))
 match_end=$(grep --text --line-number '^END_CONF:$' $0 | cut -d ':' -f 1)
 payload_end=$((match_end - 1))
 lines=$(($payload_end-$payload_start+1))
 head -n $payload_end $0 | tail -n $lines  > $station_conf_path
}

# Stop the currently running Basics station so that it can be overwritten
# by the new one
killall station

# Store the different files
prepare_station
prepare_versionp
prepare_station_conf

# Provide execute permission for Basics station binary
chmod +x $station_path

# Remove update.bin so that it is not read again next time Basics station starts
rm -f /tmp/update.bin
```

```
# Exit so that rest of this script which has binaries attached does not get executed
exit 0
```

新增承載指令碼

在基礎指令碼中，我們會附加 Basics Station 二進位檔、識別要更新至哪個版本的 version.txt，以及指令碼 (稱為 addpayload.sh) 中的 station.conf。然後，執行此指令碼。

```
*#!/bin/bash
*
base.sh > fwstation

# Add station
echo "STATION:" >> fwstation
cat $1 >> fwstation
echo "" >> fwstation
echo "END_STATION:" >> fwstation

# Add version.txt
echo "VERSION:" >> fwstation
cat $2 >> fwstation
echo "" >> fwstation
echo "END_VERSION:" >> fwstation

# Add station.conf
echo "CONF:" >> fwstation
cat $3 >> fwstation
echo "END_CONF:" >> fwstation

# executable
chmod +x fwstation
```

在執行了這些指令碼之後，您可以在終端機中執行下列命令，以產生韌體更新檔案 (fwstation)。

```
$ ./addpayload.sh station version.txt station.conf
```

產生韌體更新的簽章

LoRa Basics Station 軟體提供已簽署的韌體更新與 ECDSA 簽章。若要支援已簽署的更新，您需要：

- 必須由 ECDSA 私有金鑰產生且小於 128 個位元組的簽章。

- 用於簽章的私有金鑰，而且此金鑰必須以格式為 `sig-%d.key` 的檔案名稱儲存在閘道中。建議使用檔案名稱 `sig-0.key`。
- 透過私有金鑰的 32 位元 CRC。

簽章和 CRC 將傳遞至 AWS IoT Core for LoRaWAN API。若要產生先前的檔案，您可以使用下列指令碼 `gen.sh`，此指令碼是由 GitHub 儲存庫中的 basicstation 範例所激發的。

```
*#!/bin/bash

*function ecdsaKey() {
    # Key not password protected for simplicity
    openssl ecparam -name prime256v1 -genkey | openssl ec -out $1
}

# Generate ECDSA key
ecdsaKey sig-0.prime256v1.pem

# Generate public key
openssl ec -in sig-0.prime256v1.pem -pubout -out sig-0.prime256v1.pub

# Generate signature private key
openssl ec -in sig-0.prime256v1.pub -inform PEM -outform DER -pubin | tail -c 64 >
 sig-0.key

# Generate signature
openssl dgst -sha512 -sign sig-0.prime256v1.pem $1 > sig-0.signature

# Convert signature to base64
openssl enc -base64 -in sig-0.signature -out sig-0.signature.base64

# Print the crc
crc_res=$(crc32 sig-0.key)printf "The crc for the private key=%d\n" $((16#$crc_res))

# Remove the generated files which won't be needed later
rm -rf sig-0.prime256v1.pem sig-0.signature sig-0.prime256v1.pub
```

指令碼產生的私有金鑰應該儲存至閘道。金鑰檔案是二進位格式。

```
./gen_sig.sh fwstation
```

```
read EC key
writing EC key
read EC key
writing EC key
read EC key
writing EC key
The crc for the private key=3434210794

$ cat sig-0.signature.base64
MEQCIDPY/p2ssgXIPNCOgZr+NzeTLpX+WfBo5tYWbh5pQWN3AiBROen+XlIdMScv
AsfVfU/ZScJCalkVNZh4esyS8mNIgA==

$ ls sig-0.key
sig-0.key

$ scp sig-0.key pi@192.168.1.11:/home/pi/Documents/basicstation/examples/iotwireless
```

檢閱後續步驟

既然您已產生韌體和簽章，請移至下一個主題，將韌體檔案 `fwstation` 上傳至 Amazon S3 儲存貯體。儲存貯體是將韌體更新檔案存放為物件的容器。您可以新增 IAM 角色，其將許可 CUPS 伺服器讀取 S3 儲存貯體中的韌體更新檔案。

將韌體檔案上傳至 S3 儲存貯體並新增 IAM 角色

您可以使用 Amazon S3 來建立「儲存貯體」，這是一個容器，其中可以儲存您的韌體更新檔案。您可以將檔案上傳到 S3 儲存貯體，並新增 IAM 角色，允許 CUPS 伺服器從儲存貯體讀取您的更新檔案。如需 Amazon S3 的詳細資訊，請參閱 Amazon S3 入門。

您要上傳的韌體更新檔案取決於您正在使用的閘道。如果您所遵循的程序類似於 產生韌體更新檔案和簽章 所述的程序，您將上傳藉由執行指令碼所產生的 `fwstation` 檔案。

此程序約需 20 分鐘才能完成。

若要上傳您的韌體檔案：

- 建立 Amazon S3 儲存貯體並上傳更新檔案
- 建立許可讀取 S3 儲存貯體的 IAM 角色
- 檢閱後續步驟

建立 Amazon S3 儲存貯體並上傳更新檔案

您可以使用 AWS Management Console 來建立 Amazon S3 儲存貯體，然後將韌體更新檔案上傳至儲存貯體。

建立 S3 儲存貯體

若要建立 S3 儲存貯體，請開啟 Amazon S3 console (Amazon S3 主控台)。如果您尚未登入，請登入，然後執行下列步驟：

1. 選擇建立儲存貯體。
2. 為 Bucket name (儲存貯體名稱) 輸入唯一且有意義的名稱 (例如，`iotwirelessfwupdate`)。如需儲存貯體的建議命名慣例，請參閱 https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html。
3. 確定您已選取 AWS 區域，作為您用來建立 LoRaWAN 閘道和裝置的區域，並已選取 Block all public access (封鎖所有公有存取) 設定，以便您的儲存貯體可以使用預設許可。
4. 針對 Bucket versioning (儲存貯體版本控制) 選擇 Enable (啟用)，這將協助您將多個版本的韌體更新檔案保留在同一儲存貯體中。
5. 確認 Server-side encryption (伺服器端加密) 已設定為 Disable (停用)，然後選擇 Create bucket (建立儲存貯體)。

上傳您的韌體更新檔案

您現在可以在 AWS Management Console 顯示的儲存貯體清單中看到您的儲存貯體。選擇您的儲存貯體並完成下列步驟以上傳您的檔案。

1. 選擇您的儲存貯體，然後選擇 Upload (上傳)。
2. 選擇 Add file (新增檔案)，然後上傳韌體更新檔案。如果已遵循 產生韌體更新檔案和簽章 所述的程序，您將上傳 `fwstation` 檔案，否則上傳閘道製造商所提供的檔案。
3. 確定所有設定都設定為其預設值。確定 Predefined ACLs (預先定義的 ACL) 已設定為 private (私有)，然後選擇 Upload (上傳) 以上傳您的檔案。
4. 複製所上傳檔案的 S3 URI。選擇您的儲存貯體，您會看到您上傳的檔案顯示在 Objects (物件) 的清單中。選擇您的檔案，然後選擇 Copy S3 URI (複製 S3 URI)。URI 將如下所示：`s3://iotwirelessfwupdate/fwstation`，如果您已將儲存貯體命名為類似於上述範例 (`fwstation`)。您將會在建立 IAM 角色時使用 S3 URI。

建立許可讀取 S3 儲存貯體的 IAM 角色

您現在將建立 IAM 角色和政策，許可 CUPS 從 S3 儲存貯體讀取您的韌體更新檔案。

為您的角色建立 IAM 政策

若要為您的 AWS IoT Core for LoRaWAN 目的地角色建立 IAM 政策，請開啟 IAM 主控台的政策中樞，然後完成下列步驟：

1. 選擇 Create policy (建立政策)，然後選擇 JSON 標籤。
2. 刪除編輯器中的任何內容，然後貼上此政策文件。政策許可您存取 `iotwireless` 儲存貯體，以及物件內儲存的韌體更新檔案 `fwstation`。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::iotwirelessfwupdate/fwstation",
                "arn:aws:s3:::iotwirelessfwupdate"
            ]
        }
    ]
}
```

3. 選擇 Review policy (檢閱政策)，然後在 Name (名稱) 中，輸入此政策的名稱 (例如，`IoTWirelessFwUpdatePolicy`)。您將需要此名稱用於下一個程序中。
4. 選擇建立政策。

建立 IAM 角色與附加的政策

您現在將建立 IAM 角色，並連接先前建立的政策，以存取 S3 儲存貯體。開啟 IAM 主控台的角色中樞，然後完成下列步驟：

1. 選擇 Create Role (建立角色)。

2. 在 Select type of trusted entity (選取信任的實體類型) 中，選擇 Another AWS 帳戶 (另一個 AWS 帳戶 帳戶)。
3. 在 Account ID (帳戶 ID) 中，請輸入您的 AWS 帳戶 ID，然後選擇 Next: Permissions (下一步：許可)。
4. 在搜尋方塊中，輸入您在上一個程序中建立的 IAM 政策名稱。檢查您稍早在搜尋結果中建立的 IAM 政策 (例如，`IoTWirelessFwUpdatePolicy`)，然後選擇它。
5. 選擇 Next: Tags (下一步：標籤)，然後選擇 Next: Review (下一步：檢閱)。
6. 在 Role name (角色名稱) 中，輸入此角色的名稱 (例如，`IoTWirelessFwUpdateRole`)，然後選擇 Create role (建立角色)。

編輯 IAM 角色的信任關係

在執行前一個步驟之後顯示的確認訊息中，選擇您建立的角色名稱以編輯該角色。您將編輯角色以新增下列信任關係。

1. 在所建立角色的 Summary (摘要) 區段中，選擇 Trust relationships (信任關係) 標籤，然後選擇 Edit trust relationship (編輯信任關係)。
2. 在 Policy Document (政策文件) 中，變更 `Principal` 屬性以看起來像此範例。

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

在您變更 `Principal` 屬性之後，完整政策文件應該看起來像此範例。

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

3. 若要儲存您的變更，請選擇 Update Trust Policy (更新信任政策)。
4. 為您的角色取得 ARN。選擇您的 IAM 角色，然後在 Summary (摘要) 區段中，您會看到 Role ARN (角色 ARN)，例如 `arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`。複製此 Role ARN (角色 ARN)。

檢閱後續步驟

既然您已建立 S3 儲存貯體，以及允許 CUPS 伺服器讀取 S3 儲存貯體的 IAM 角色，請移至下一個主題以排程並執行韌體更新。保留您先前複製的 S3 URI 和 Role ARN (角色 ARN)，以便您可以輸入它們，來建立為了執行韌體更新而執行的任務定義。

使用任務定義來排程和執行韌體更新

您可以使用任務定義來包含韌體更新的詳細資訊，以及定義更新。AWS IoT Core for LoRaWAN 根據與閘道相關聯的下列三個欄位中的資訊來提供韌體更新。

- 工作站

  Basics Station 軟體的版本和建置時間。若要識別此資訊，您也可以使用閘道所執行的 Basics Station 軟體 (例如，`2.0.5(rpi/std) 2021-03-09 03:45:09`) 來產生它。

- PackageVersion

  韌體版本，由閘道中的檔案 `version.txt` 指定。儘管閘道中可能不存在此資訊，但我們建議您定義韌體版本 (例如，`1.0.0`) 作為一種方式。

- 模型

  閘道正在使用的平台或模型 (例如，Linux)。

此程序需要 20 分鐘才能完成。

若要完成此程序：

- 取得目前在閘道上執行的版本
- 建立無線閘道任務定義
- 執行韌體更新任務並追蹤進度

取得目前在閘道上執行的版本

若要判斷閘道是否符合韌體更新的資格，CUPS 伺服器會在閘道於 CUPS 請求期間呈現所有三個欄位 (Station、PackageVersion 和 Model) 時，檢查它們是否符合資格。當您使用任務定義時，這些欄位會儲存為 CurrentVersion 欄位的一部分。

您可以使用 AWS IoT Core for LoRaWAN API 或 AWS CLI，為您的閘道取得 CurrentVersion。下列命令顯示如何使用 CLI 來取得此資訊。

1. 如果已經佈建閘道，您可以使用 get-wireless-gateway 命令，取得閘道的相關資訊。

```
aws iotwireless get-wireless-gateway \
    --identifier 5a11b0a85a11b0a8 \
        --identifier-type GatewayEui
```

下列顯示此命令的範例輸出：

```
{
    "Name": "Raspberry pi",
    "Id": "1352172b-0602-4b40-896f-54da9ed16b57",
    "Description": "Raspberry pi",
    "LoRaWAN": {
        "GatewayEui": "5a11b0a85a11b0a8",
        "RfRegion": "US915"
    },
    "Arn": "arn:aws:iotwireless:us-
east-1:231894231068:WirelessGateway/1352172b-0602-4b40-896f-54da9ed16b57"
}
```

2. 使用 get-wireless-gateway 命令回報的無線閘道 ID，您可以使用 get-wireless-gateway-firmware-information 命令來取得 CurrentVersion。

```
aws iotwireless get-wireless-gateway-firmware-information \
    --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

下列顯示命令的範例輸出，其中資訊來自 CurrentVersion 顯示的所有三個欄位。

```
{
    "LoRaWAN": {
        "CurrentVersion": {
            "PackageVersion": "1.0.0",
```

```
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
        }
    }
}
```

建立無線閘道任務定義

建立任務定義時，建議您使用 AutoCreateTasks 參數來指定自動建立工作。AutoCreateTasks 適用於任何符合前述所有三個參數的閘道。如果停用此參數，則必須手動將參數指派給閘道。

您可以建立無線閘道任務定義，方法為使用 AWS IoT Core for LoRaWAN API 或 AWS CLI。下列命令顯示如何使用 CLI 建立任務定義。

1. 建立檔案 input.json，其中將包含要傳遞至 CreateWirelessGatewayTaskDefinition API 的資訊。在 input.json 檔案中，提供您先前取得的下列資訊：
   - UpdateDataSource

     提供物件的連結，其中包含您上傳至 S3 儲存貯體的韌體更新檔案 (例如，s3://iotwirelessfwupdate/fwstation)。
   - UpdateDataRole

     為您建立的 IAM 角色提供角色 ARN 的連結，該 IAM 角色許可讀取 S3 儲存貯體 (例如，arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole)。
   - SigKeyCRC 和 UpdateSignature

     此資訊可能是由閘道製造商提供，但如果已遵循 產生韌體更新檔案和簽章 所述的程序，您會在產生簽章時找到此資訊。
   - CurrentVersion

     提供您先前執行 get-wireless-gateway-firmware-information 命令所取得的 CurrentVersion 輸出。

     ```
     cat input.json
     ```

     下列顯示 input.json 檔案的內容。

     ```
     {
         "AutoCreateTasks": true,
     ```

```
    "Name": "FirmwareUpdate",
    "Update":
    {
        "UpdateDataSource" : "s3://iotwirelessfwupdate/fwstation",
        "UpdateDataRole" : "arn:aws:iam::123456789012:role/
IoTWirelessFwUpdateRole",
        "LoRaWAN" :
        {
            "SigKeyCrc": 3434210794,
            "UpdateSignature": "MEQCIDPY/p2ssgXIPNCOgZr+NzeTLpX
+WfBo5tYWbh5pQWN3AiBROen+XlIdMScvAsfVfU/ZScJCalkVNZh4esyS8mNIgA==",
            "CurrentVersion" :
            {
            "PackageVersion": "1.0.0",
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
            }
        }
    }
}
```

2. 將 `input.json` 檔案傳遞至 create-wireless-gateway-task-definition 命令來建立任務定義。

```
aws iotwireless create-wireless-gateway-task-definition \
    --cli-input-json file://input.json
```

下列顯示命令的輸出。

```
{
    "Id": "4ac46ff4-efc5-44fd-9def-e8517077bb12",
    "Arn": "arn:aws:iotwireless:us-
east-1:231894231068:WirelessGatewayTaskDefinition/4ac46ff4-efc5-44fd-9def-
e8517077bb12"
}
```

執行韌體更新任務並追蹤進度

閘道已準備好接收韌體更新，一旦開啟電源，就會連接到 CUPS 伺服器。當 CUPS 伺服器在閘道版本中找到相符項目時，就會排定韌體更新。

任務是程序中的任務定義。當您已將 AutoCreateTasks 設定為 True 來指定自動任務建立時，只要找到相符的閘道，韌體更新任務就會開始。

您可以使用 GetWirelessGatewayTask API 追蹤任務進度。當您執行 get-wireless-gateway-task 命令時，它會將任務狀態顯示為 IN_PROGRESS。

```
aws iotwireless get-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

下列顯示命令的輸出。

```
{
    "WirelessGatewayId": "1352172b-0602-4b40-896f-54da9ed16b57",
    "WirelessGatewayTaskDefinitionId": "ec11f9e7-b037-4fcc-aa60-a43b839f5de3",
    "LastUplinkReceivedAt": "2021-03-12T09:56:12.047Z",
    "TaskCreatedAt": "2021-03-12T09:56:12.047Z",
    "Status": "IN_PROGRESS"
}
```

當您下次執行命令時，如果韌體更新生效，它會顯示更新的欄位 Package、Version 和 Model，而且任務狀態會變更為 COMPLETED。

```
aws iotwireless get-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

下列顯示命令的輸出。

```
{
    "WirelessGatewayId": "1352172b-0602-4b40-896f-54da9ed16b57",
    "WirelessGatewayTaskDefinitionId": "ec11f9e7-b037-4fcc-aa60-a43b839f5de3",
    "LastUplinkReceivedAt": "2021-03-12T09:56:12.047Z",
    "TaskCreatedAt": "2021-03-12T09:56:12.047Z",
    "Status": "COMPLETED"
}
```

在此範例中，我們向您展示了使用 Raspberry Pi 型 RAKWireless 閘道進行的韌體更新。韌體更新指令碼會停止執行中的 BasicStation，以存放更新的 Package、Version 和 Model 欄位，因此必須重新啟動 BasicStation。

```
2021-03-12 09:56:13.108 [CUP:INFO] CUPS provided update.bin
```

```
2021-03-12 09:56:13.108 [CUP:INFO] CUPS provided signature len=70 keycrc=37316C36
2021-03-12 09:56:13.148 [CUP:INFO] ECDSA key#0 -> VERIFIED
2021-03-12 09:56:13.148 [CUP:INFO] Running update.bin as background process
2021-03-12 09:56:13.149 [SYS:VERB] /tmp/update.bin: Forked, waiting...
2021-03-12 09:56:13.151 [SYS:INFO] Process /tmp/update.bin (pid=6873) completed
2021-03-12 09:56:13.152 [CUP:INFO] Interaction with CUPS done - next regular check in
 10s
```

如果韌體更新失敗，您會看到來自 CUPS 伺服器的狀態 FIRST_RETRY，而且閘道會傳送相同的請求。如果 CUPS 伺服器在 SECOND_RETRY 之後無法連接至閘道，它將顯示狀態 FAILED。

在前一個任務是 COMPLETED 或 FAILED 之後，請使用 delete-wireless-gateway-task 命令刪除舊的任務，然後再啟動新任務。

```
aws iotwireless delete-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

選擇閘道以接收 LoRaWAN 下行資料流量

從 AWS IoT Core for LoRaWAN 傳送下行訊息時，您可以選擇要用於下行資料流量的閘道。您可以指定個別閘道，或從閘道清單中選擇以接收下行流量。

如何指定閘道清單

您可以使用 SendDataToWirelessDevice API 操作，指定從 AWS IoT Core for LoRaWAN 傳送下行訊息至裝置時要使用的個別閘道或閘道清單。呼叫 API 操作時，使用閘道的 ParticipatingGateways 物件指定下列參數。

> **Note**
>
> 您要使用的閘道清單不適用於 AWS IoT 主控台。僅在使用 SendDataToWirelessDevice API 操作或 CLI 時才能指定此閘道清單。

- DownlinkMode：指出是以順序模式還是並行模式傳送下行訊息。對於 A 類裝置，請指定 UsingUplinkGateway，僅使用先前上行訊息傳輸中選擇的閘道。
- GatewayList：您要用來傳送下行資料流量的閘道清單。下行承載將以指定的頻率傳送到指定的閘道。這是使用 GatewayListItem 物件清單來表示，包含 GatewayId 和 DownlinkFrequency 對。

- `TransmissionInterval`：在將承載傳輸到下一個閘道之前，AWS IoT Core for LoRaWAN 等待的持續時間。

> **Note**
>
> 只有在將下行訊息傳送至 B 類或 C 類無線裝置時，才能指定此閘道清單。如果您使用 A 類裝置，則傳送下行訊息至裝置時，將使用您在傳送上行訊息時選擇的閘道。

下列範例說明如何指定閘道的這些參數。`input.json` 檔案將包含其他詳細資訊。如需使用 `SendDataToWirelessDevice` API 傳送下行訊息的詳細資訊，請參閱 使用 API 執行下行佇列操作。

> **Note**
>
> 當您使用 AWS IoT 主控台從 AWS IoT Core for LoRaWAN 傳送下行訊息時，無法使用用來指定參與閘道清單的參數。

```
aws iotwireless send-data-to-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
    --transmit-mode "1" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --cli-input-json file://input.json
```

下列顯示 `input.json` 檔案的內容。

input.json 的內容

```
{
    "WirelessMetadata": {
        "LoRaWAN": {
            "FPort": "1",
            "ParticipatingGateways": {
                "DownlinkMode": "SEQUENTIAL",
                "TransmissionInterval": 1200,
                "GatewayList": [
                    {
                        "DownlinkFrequency": 100000000,
```

```
                        "GatewayID": a01b2c34-d44e-567f-abcd-0123e445663a
                    },
                    {
                        "DownlinkFrequency": 100000101,
                        "GatewayID": 12345678-a1b2-3c45-67d8-e90fa1b2c34d
                    }
                ]
            }
        }
    }
}
```

執行此命令的輸出會為下行訊息產生 MessageId。在某些情況下，即使您收到 MessageId，資料封包繩也可能會被丟棄。如需如何解決錯誤的詳細資訊，請參閱 下行訊息佇列錯誤疑難排解。

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

取得參與閘道清單的資訊

您可以列出下行佇列中的訊息，以取得有關參與接收下行訊息之閘道清單的資訊。若要列出訊息，請使用 ListQueuedMessages API。

```
aws iotwireless list-queued-messages \
    --wireless-device-type "LoRaWAN"
```

執行此命令會傳回佇列中訊息及其參數的相關資訊。

利用 AWS IoT Core for LoRaWAN 管理裝置

下列是搭配使用裝置與 AWS IoT Core for LoRaWAN 時的一些重要考量。如需如何新增裝置至 AWS IoT Core for LoRaWAN 的相關資訊，請參閱 將裝置加入 AWS IoT Core for LoRaWAN。

裝置考量

選取要用於與 AWS IoT Core for LoRaWAN 通訊的裝置時，請考慮下列事項。

- 可用的感應器

- 電池容量
- 能源消耗
- 費用
- 天線類型和傳輸範圍

搭配符合 AWS IoT Core for LoRaWAN 資格的閘道使用裝置

您使用的裝置可以與有資格搭配 AWS IoT Core for LoRaWAN 使用的無線閘道配對。您可以在 AWS Partner Device Catalog 找到這些閘道和開發人員套件。我們也建議您考慮將這些裝置接近您的閘道。如需詳細資訊，請參閱使用來自 AWS Partner Device Catalog 的合格閘道。

LoRaWAN 版本

AWS IoT Core for LoRaWAN 支援所有符合 LoRa Alliance 標準化 1.0.x 或 1.1 LoRaWAN 規格的裝置。

啟用模式

在您的 LoRaWAN 裝置可以傳送上行資料之前，您必須先完成稱為「啟用」或「聯結」的程序。若要啟用您的裝置，您可以使用 OTAA (無線啟用) 或 ABP (個人化啟用)。我們建議您使用 OTAA 來啟用裝置，因為每次啟用都會產生新的工作階段金鑰，這樣會使其更安全。

您的無線裝置規格是以 LoRaWAN 版本和啟用模式為基礎，這會決定每次啟動所產生的根金鑰和工作階段金鑰。如需詳細資訊，請參閱使用主控台將您的無線裝置規格新增至 AWS IoT Core for LoRaWAN。

裝置類別

LoRaWAN 裝置可以隨時傳送上行訊息。接聽下行訊息會耗用電池容量，並減少電池持續時間。LoRaWAN 通訊協定會指定三種類別的 LoRaWAN 裝置。

- A 類裝置大部分時間都處於休眠狀態，而且僅短時間接聽下行訊息。這些元件大多是電池供電的感應器，電池壽命長達 10 年。
- B 類裝置可以在排定的下行位置接收訊息。這些裝置大多是電池供電的致動器。
- C 類裝置永遠都不會休眠，並持續接聽傳入的訊息，所以在接收訊息時沒有太多的延遲。這些裝置大多是電源供電的致動器。

如需這些無線裝置考量的詳細資訊，請參閱 進一步了解 LoRaWAN 中提到的資源。

主題

使用 AWS IoT Core for LoRaWAN 執行調適型資料速率 (ADR)

為了最佳化裝置傳輸耗電量，同時確保閘道接收來自終端裝置的訊息，AWS IoT Core for LoRaWAN 會使用調適型資料速率。調適型資料速率會指示終端裝置最佳化資料速率、傳輸功率及重新傳輸次數，並同時嘗試降低閘道收到的封包錯誤率。例如，如果您的終端裝置位於閘道附近，則調適型資料速率就會降低傳輸功率並提高資料速率。

主題

調適型資料速率 (ADR) 的運作方式

若要啟用 ADR，您的裝置必須在訊框標頭中設定 ADR 位元。設定 ADR 位元後，AWS IoT Core for LoRaWAN 會傳送 `LinkADRReq` MAC 命令，而您的裝置會使用包含 ADR 命令的 ACK 狀態的 `LinkADRAns` 命令進行回應。一旦裝置收到 ADR 命令的 ACK 狀態，就會按照來自 AWS IoT Core for LoRaWAN 的 ADR 指示進行，並調整傳輸參數值以達到最佳資料速率。

AWS IoT Core for LoRaWAN ADR 演算法會使用上行中繼資料歷程記錄中的 SINR 資訊，來判斷裝置要使用的最佳傳輸功率和資料速率。此演算法會使用框架標頭中設定了 ADR 位元後開始傳送的最近 20 個上行訊息。為了判斷重新傳輸次數，它會使用封包錯誤率 (PER)，也就是遺失的封包總數百分比。使用此演算法時，您只能控制資料速率的範圍，也就是資料速率的下限和上限。

設定資料速率限制 (CLI)

根據預設，當您在 LoRaWAN 裝置的訊框標頭中設定 ADR 位元時，AWS IoT Core for LoRaWAN 將會執行 ADR。您可以在使用 AWS IoT Wireless API 操作 `CreateServiceProfile` 或 AWS CLI 命令 `create-service-profile` 為 LoRaWAN 裝置建立服務設定檔時，控制資料速率的下限和上限。

> **Note**
>
> 從 AWS Management Console 建立服務設定檔時，則無法指定資料速率的上限和下限。只能使用 AWS IoT Wireless API 或 AWS CLI 進行指定。

若要指定資料速率的下限和上限，請使用 `DrMin` 和 `DrMax` 參數搭配 `CreateServiceProfile` API 操作。預設的資料速率下限和上限分別為 0 和 15。例如，下列 CLI 命令會設定資料速率下限 3 和上限 12。

```
aws iotwireless create-service-profile \
    --lorawan DrMin=3,DrMax=12
```

執行此命令會產生服務設定檔的 ID 和 Amazon Resource Name (ARN)。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

您可以使用 AWS IoT Wireless API 操作 GetServiceProfile 或 AWS CLI 命令 get-service-profile 取得所指定參數的值。

```
aws iotwireless get-service-profile --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

執行此命令會產生服務設定檔參數的值。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:651419225604:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "LoRaWAN": {
        "UlRate": 60,
        "UlBucketSize": 4096,
        "DlRate": 60,
        "DlBucketSize": 4096,
        "AddGwMetadata": false,
        "DevStatusReqFreq": 24,
        "ReportDevStatusBattery": false,
```

```
        "ReportDevStatusMargin": false,
        "DrMin": 3,
        "DrMax": 12,
        "PrAllowed": false,
        "HrAllowed": false,
        "RaAllowed": false,
        "NwkGeoLoc": false,
        "TargetPer": 5,
        "MinGwDiversity": 1
    }
}
```

如果您已建立多個設定檔，則可以使用 API 操作 `ListServiceProfiles` 或 AWS CLI 命令 `list-service-profiles` 列出您 AWS 帳戶 中的服務設定檔，然後使用 `GetServiceProfile` API 或 `get-service-profile` CLI 命令擷取您自訂資料速率限制的服務設定檔。

管理您的 LoRaWAN 裝置和 AWS IoT 之間的通訊

將 LoRaWAN 裝置連接至 AWS IoT Core for LoRaWAN 後，裝置即可開始將訊息傳送至雲端。上行訊息是從您的裝置傳送並由 AWS IoT Core for LoRaWAN 接收的訊息。LoRaWAN 裝置可以隨時傳送上行訊息，然後將訊息轉寄至其他 AWS 服務 和雲端託管應用程式。從 AWS IoT Core for LoRaWAN 和其他 AWS 服務 和應用程式傳送至您的裝置的訊息稱為下行訊息。

下面介紹如何查看和管理裝置和雲端之間所傳送的上行和下行訊息。您可以維護下行訊息佇列，並依照這些訊息新增至佇列的順序，將這些訊息傳送至您的裝置。

主題

- 檢視從 LoRaWAN 裝置傳送的上行訊息格式
- 將要傳送至 LoRaWAN 裝置的下行訊息排入佇列

檢視從 LoRaWAN 裝置傳送的上行訊息格式

在將 LoRaWAN 裝置連接至 AWS IoT Core for LoRaWAN 之後，您可以觀察將從無線裝置接收的上行訊息格式。

在可以觀察上行訊息之前

您必須已加入無線裝置，並將裝置連接至 AWS IoT，以便它可以傳輸和接收資料。如需如何將裝置加入至 AWS IoT Core for LoRaWAN 的相關資訊，請參閱 將裝置加入 AWS IoT Core for LoRaWAN。

上行訊息包含哪些內容？

LoRaWAN 裝置藉由使用 LoRaWAN 閘道來連接至 AWS IoT Core for LoRaWAN。您從裝置收到的上行訊息將包含下列資訊。

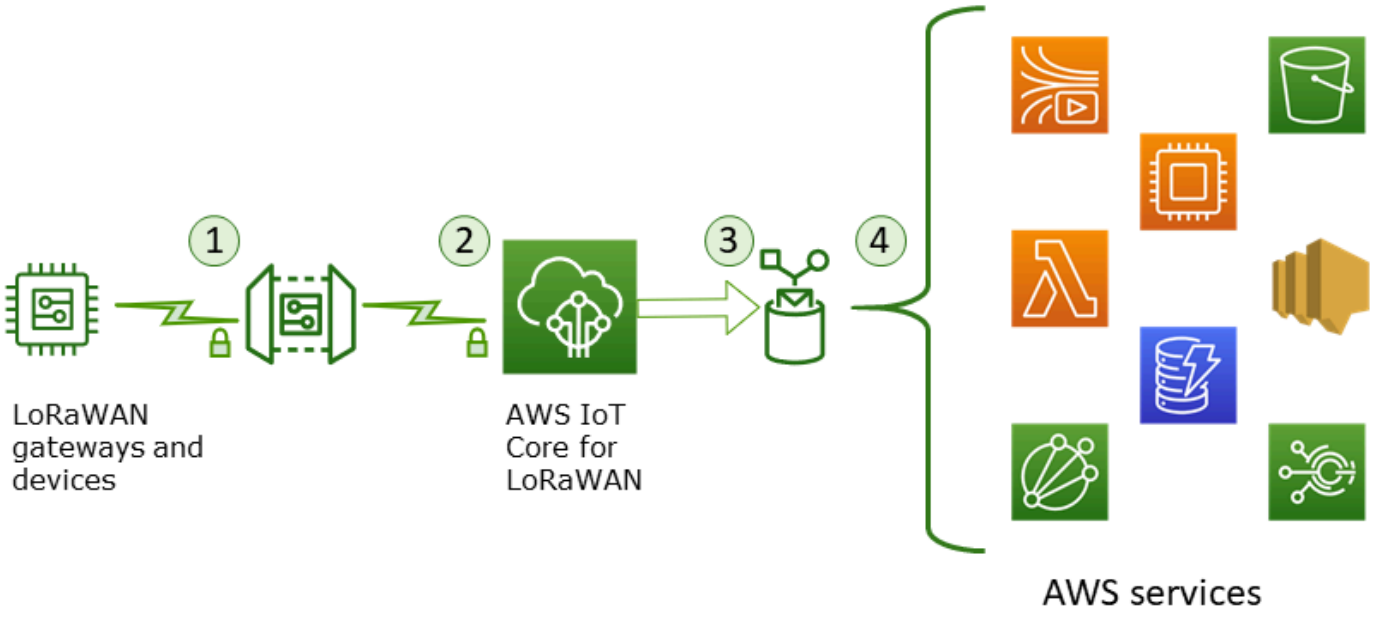
- 對應至從無線裝置傳送之加密承載訊息的承載資料。
- 無線中繼資料，其中包括：
  - 裝置資訊，例如 DevEui、資料速率，以及裝置操作所在的頻率通道。
  - 連接至裝置之閘道的選用額外參數和閘道資訊。閘道參數包括閘道的 EUI、SNR 和 RSSi。

  透過使用無線中繼資料，您可以取得有關無線裝置的實用資訊，以及在裝置與 AWS IoT 之間傳輸的資料。例如，您可以使用 `AckedMessageId` 參數，來檢查裝置是否已收到最後確認的下行訊息。或者，如果選擇包含閘道資訊，您可以識別是否要切換到更接近裝置的更強閘道通道。

如何觀察上行訊息？

在加入了您的裝置之後，您可以在 AWS IoT 主控台的 Test (測試) 頁面上使用 MQTT 測試用戶端，以訂閱您在建立目的地時所指定的主題。在連接您的裝置並開始傳送承載資料之後，您會開始看到訊息。

此圖表會識別連接至 AWS IoT Core for LoRaWAN 之 LoRaWAN 系統中的重要元素，這會顯示主要資料平面和資料流經系統的方式。



當無線裝置開始傳送上行資料時，AWS IoT Core for LoRaWAN 會將無線中繼資料資訊與承載一起包裝，然後傳送到您的 AWS 應用程式。

上行訊息範例

下列範例顯示從裝置接收之上行訊息的格式。

```
{
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",
    "PayloadData": "Cc48AAAAAAAAAA=",
    "WirelessMetadata":
    {
        "LoRaWAN":
        {
            "ADR": false,
            "Bandwidth": 125,
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "0",
            "DevAddr": "00b96cd4",
            "DevEui": "58a0cb000202c99",
            "FOptLen": 2,
            "FCnt": 1,
            "Fport": 136,
            "Frequency": "868100000",
            "Gateways": [
             {
                    "GatewayEui": "80029cfffe5cf1cc",
                    "Snr": -29,
                    "Rssi": 9.75
             }
             ],
            "MIC": "7255cb07",
            "MType": "UnconfirmedDataUp",
            "Major": "LoRaWANR1",
            "Modulation": "LORA",
            "PolarizationInversion": false,
            "SpreadingFactor": 12,
            "Timestamp": "2021-05-03T03:24:29Z"

        }
    }
}
```

從上行中繼資料中排除閘道中繼資料

如果您想要從上行中繼資料中排除閘道中繼資料資訊，請在建立服務設定檔時停用 AddGwMetadata 參數。如需停用此參數的相關資訊，請參閱 新增裝置設定檔。

在此情況下，您不會看到上行中繼資料中的 Gateways 區段，如下列範例所示。

```
{
    "WirelessDeviceId": "0d9a439b-e77a-4573-a791-49d5c0f4db95",
    "PayloadData": "AAAAAAAA//8=",
    "WirelessMetadata": {
        "LoRaWAN": {
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "1",
            "DevAddr": "01920f27",
            "DevEui": "ffffff10000163b0",
            "FCnt": 1,
            "FPort": 5,
            "Timestamp": "2021-04-29T05:19:43.646Z"
    }
  }
}
```

將要傳送至 LoRaWAN 裝置的下行訊息排入佇列

雲端託管應用程式和其他 AWS 服務 可以向您的無線裝置傳送下行訊息。下行訊息是從 AWS IoT Core for LoRaWAN 傳送至您的無線裝置的訊息。您可以為已加入 AWS IoT Core for LoRaWAN 的每台裝置安排和傳送下行訊息。

如果您有多個裝置要為其傳送下行訊息，則可以使用多點傳送群組。多點傳送組中的裝置會共用相同的多點傳送地址，然後將資料分配至整個收件人裝置群組。如需詳細資訊，請參閱建立多點傳送群組，以將下行承載傳送至多個裝置。

下行訊息佇列的運作方式

LoRaWAN 裝置的裝置類別決定了佇列中的訊息傳送至裝置的方式。A 類裝置會將上行訊息傳送至 AWS IoT Core for LoRaWAN 以指示裝置可用於接收下行訊息。B 類裝置可以在常規的下行位置接收訊息。C 類裝置可以隨時接收下行訊息。如需裝置分級的詳細資訊，請參閱 裝置類別。

下面顯示如何將訊息排入佇列並將其傳送至 A 類裝置。

1. AWS IoT Core for LoRaWAN 會透過訊框連接埠、承載資料，以及使用 AWS IoT 主控台或 AWS IoT Wireless API 所指定的確認模式參數，來緩衝您新增至佇列的下行訊息。
2. 您的 LoRaWAN 裝置傳送一則上行訊息，指示其處於線上狀態，並且可以開始接收下行訊息。
3. 如已將多個下行訊息新增至佇列，則 AWS IoT Core for LoRaWAN 將佇列中的第一則下行訊息傳送至已設定確認 (ACK) 旗標的裝置。
4. 您的裝置會立即將上行訊息傳送至 AWS IoT Core for LoRaWAN，或是進入睡眠狀態，直到下一個上行訊息產生且在訊息中包含 ACK 旗標。
5. AWS IoT Core for LoRaWAN 接收含有 ACK 旗標的上行訊息時，會清除佇列中的下行訊息，指示您的裝置已成功收到下行訊息。如果檢查三次後上行訊息中仍缺少 ACK 旗標，則會丟棄該訊息。

使用主控台執行下行佇列操作

您可以使用 AWS Management Console 將下行訊息排入佇列，並根據需要清除個別訊息或整個佇列。若為 A 類裝置，在從裝置接收上行訊息以指示裝置處於線上狀態後，已排入佇列的訊息將會傳送至裝置。訊息傳送後，即會自動從佇列中清除。

將下行訊息排入佇列

建立下行訊息佇列

1. 前往 AWS IoT 主控台的裝置集線器，然後選擇要將下行訊息排入佇列的裝置。
2. 在裝置詳細資訊頁面的 Downlink messages (下行訊息) 部分中，選擇 Queue downlink messages (將下行訊息排入佇列)。
3. 請指定以下參數以設定下行訊息：
   - FPort：選擇裝置與 AWS IoT Core for LoRaWAN 通訊的訊框連接埠。
   - Payload (承載)：指定您要傳送至裝置的承載訊息。承載大小上限為 242 個位元組。如果啟用了調適型資料速率 (ADR)，AWS IoT Core for LoRaWAN 會使用其為您的承載大小選擇最佳的資料速率。您可以根據需要進一步最佳化資料速率。
   - Acknowledge mode (確認模式)：確認您的裝置是否已收到下行訊息。如果訊息需要此模式，您將會在資料串流中看到含有 ACK 旗標的上行訊息，且訊息將從佇列中清除。
4. 若要將下行訊息新增至佇列，請選擇 Submit (提交)。

您的下行訊息現已新增至佇列。如果您未看到訊息或接收到錯誤，則可以疑難排解錯誤，如 下行訊息佇列錯誤疑難排解 中所述。

> **Note**
>
> 將下行訊息新增至佇列後，您便無法再編輯參數 FPort、Payload (承載) 和 Acknowledge mode (確認模式)。若要使用不同的參數值來傳送下行訊息，您可以刪除此訊息，並使用更新的參數值將新的下行訊息排入佇列。

佇列會列出您新增的下行訊息。若要查看裝置和 AWS IoT Core for LoRaWAN 之間交換的上行和下行訊息的承載，您可以使用網路分析器。如需詳細資訊，請參閱使用網路分析器即時監控無線資源機群。

列出下行訊息佇列

您建立的下行訊息即會新增至佇列。每個後續下行訊息均會在此訊息之後新增至佇列中。您可以在裝置詳細資訊頁面的 Downlink messages (下行訊息) 部分查看下行訊息清單。接收上行後，訊息將傳送至裝置。在您的裝置收到下行訊息後，訊息將從佇列中移除。然後，下一則訊息將在佇列中向上移動，以便傳送至您的裝置。

刪除個別下行訊息或清除整個佇列

每則下行訊息在傳送至您的裝置後，均會自動從佇列中清除。您亦可刪除個別訊息或清除整個下行佇列。這些動作無法復原。

- 如果您在佇列中找到不想傳送的訊息，請選擇訊息，然後選擇 Delete (刪除)。
- 如果您不希望將佇列中的任何訊息傳送至您的裝置，則可以選擇 Clear downlink queue (清除下行佇列)。

使用 API 執行下行佇列操作

您可以使用 AWS IoT Wireless API 將下行訊息排入佇列，並根據需要清除個別訊息或整個佇列。

將下行訊息排入佇列

若要建立下行訊息佇列，請使用 `SendDataToWirelessDevice` API 操作或 `send-data-to-wireless-device` CLI 命令。

```
aws iotwireless send-data-to-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
    --transmit-mode "1" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata LoRaWAN={FPort=1}
```

執行此命令的輸出會為下行訊息產生 MessageId。在某些情況下，即使您收到 MessageId，資料封包繩也可能會被丟棄。如需如何解決錯誤的詳細資訊，請參閱 下行訊息佇列錯誤疑難排解。

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

列出佇列中的下行訊息

若要列出佇列中的所有下行訊息，請使用 ListQueuedMessages API 操作或 list-queued-messages CLI 命令。

```
aws iotwireless list-queued-messages
```

預設情況下，執行此命令時最多可顯示 10 則下行訊息。

移除個別下行訊息或清除整個佇列

若要從佇列中移除個別訊息或清除整個佇列，請使用 DeleteQueuedMessages API 操作或 delete-queued-messages CLI 命令。

- 若要移除個別訊息，請提供您想要為無線裝置移除的訊息的 messageID (由 wirelessDeviceId 指定)。
- 若要清除整個下行佇列，請針對您的無線裝置將 messageID 指定為 * (由 wirelessDeviceId 指定)。

下行訊息佇列錯誤疑難排解

若您並未看到預期的結果，請檢查以下事項：

- 下行訊息不會顯示在 AWS IoT 主控台中

  如果在新增 (如 使用主控台執行下行佇列操作 中所述) 後在佇列中看不到下行訊息，這可能是因為您的裝置尚未完成名為啟用或聯結程序的程序。當您的裝置加入 AWS IoT Core for LoRaWAN 時，此程序即完成。如需詳細資訊，請參閱使用主控台將您的無線裝置規格新增至 AWS IoT Core for LoRaWAN。

  在將您的裝置加入 AWS IoT Core for LoRaWAN 後，您可以使用網路分析器或 Amazon CloudWatch 來監控您的裝置，以檢查聯結和重新聯結是否成功。如需詳細資訊，請參閱監控工具。
- 使用 API 時遺失下行訊息資料封包

當您使用 SendDataToWirelessDevice API 操作時，API 會傳回唯一的 MessageId。但是，這無法確認您的 LoRaWAN 裝置是否已收到下行訊息。如果您的裝置尚未完成聯結程序，則下行資料封包可能會被丟棄。如需如何解決此錯誤的詳細資訊，請參閱上一節。

- 傳送下行訊息時發生缺少 ARN 錯誤

  從佇列將下行訊息傳送至您的裝置時，您可能會接收到遺失 Amazon 資源名稱 (ARN) 錯誤。由於未為接收下行訊息的裝置正確指定目的地，即可能會出現此錯誤。若要解決此錯誤，請檢查裝置的目的地詳細資訊。

管理來自公有 LoRaWAN 裝置網路的 LoRaWAN 流量 (Everynet)

使用公開可用的 LoRaWAN 網路，在幾分鐘內就能將您的 LoRaWAN 裝置連線到雲端。AWS IoT Core for LoRaWAN 現在可支援美國和英國境內的 Everynet 網路覆蓋範圍。使用公有網路時，每部裝置每月都須支付公有網路連線費。定價適用於有提供公有網路連線的所有 AWS 區域。如需有關此功能定價的詳細資訊，請參閱 AWS IoT Core 定價頁面。

> **Important**
>
> 公有網路是由 Everynet 直接操作並作為服務提供。使用此功能之前，請參閱適用的 AWS 服務條款。此外，如果您透過 AWS IoT Core for LoRaWAN 使用公有網路，特定 LoRaWAN 裝置資訊 (例如 DevEUI 和 JoinEUI) 將會在 AWS IoT Core for LoRaWAN 可用的區域中複寫。

AWS IoT Core for LoRaWAN 依照 LoRa Alliance 漫遊規格支援公有 LoRaWAN 網路，如 LoRaWAN 後端介面 1.0 規格所述。公有網路功能可用於連線家用網路外部的終端裝置。為了支援此功能，AWS IoT Core for LoRaWAN 與 Everynet 合作提供更廣的無線電涵蓋範圍。

使用公有 LoRaWAN 網路的好處

您的 LoRaWAN 裝置可以使用公有網路連線到雲端，這樣可以縮短部署時間，並減少維護私有 LoRaWAN 網路所需的時間和成本。

透過使用公有 LoRaWAN 網路，您可獲得諸如涵蓋範圍擴展、無須無線電網路執行核心以及涵蓋範圍密集化等好處。此功能可用於：

- 在裝置離開家用網路時提供涵蓋範圍，例如公有 LoRaWAN 網路支援架構一節所示圖中的裝置 A。
- 將涵蓋範圍擴大至沒有 LoRa 閘道可連線的裝置，例如公有 LoRaWAN 網路支援架構一節所示圖中的裝置 B。然後，裝置可以使用合作夥伴提供的閘道連線到家用網路。

您的 LoRaWAN 裝置可以使用公有網路，透過漫遊功能連線到雲端，這樣可以縮短部署時間，並減少維護私有 LoRaWAN 網路所需的時間和成本。

下列各節說明公有網路支援架構、公有 LoRaWAN 網路支援的運作方式，以及如何使用此功能。

主題

LoRaWAN 公有網路支援的運作方式

AWS IoT Core for LoRaWAN 依照 LoRa Alliance 規格，支援被動漫遊功能。使用被動漫遊時，漫遊程序對終端裝置完全透明。在家用網路外部漫遊的終端裝置可以連線到該網路中的閘道，並使用應用程式伺服器交換上行和下行資料。在整個漫遊過程中，裝置保持與家用網路的連線。

> **Note**
>
> AWS IoT Core for LoRaWAN 僅支援被動漫遊的無狀態功能。不支援切換漫遊。在切換漫遊中，當您的裝置移動到家用網路外部時，會切換到不同的電信業者。

主題

公有 LoRaWAN 網路的概念

以下說明 AWS IoT Core for LoRaWAN 支援的公有網路功能採用的概念。

LoRaWAN 網路伺服器 (LNS)

LNS 是獨立的私有伺服器，可以在您的內部部署執行，也可以是雲端服務。AWS IoT Core for LoRaWAN 是在雲端上提供服務的 LNS。

家用網路伺服器 (hNS)

家用網路是裝置所屬的網路。家用網路伺服器 (hNS) 是 AWS IoT Core for LoRaWAN 儲存裝置佈建資料 (例如 `DevEUI`、`AppEUI` 和工作階段金鑰) 的 LNS。

已瀏覽網路伺服器 (vNS)

已瀏覽網路是裝置離開家用網路時所涵蓋的網路。到訪網路伺服器 (vNS) 是與 hNS 簽訂業務和技術合約的 LNS，可為終端裝置提供服務。AWS 合作夥伴 Everynet 會作為到訪網路提供涵蓋範圍。

服務網路伺服器 (sNS)

服務網路伺服器 (sNS) 是處理裝置 MAC 命令的 LNS。一個 LoRa 工作階段只能有一個 sNS。

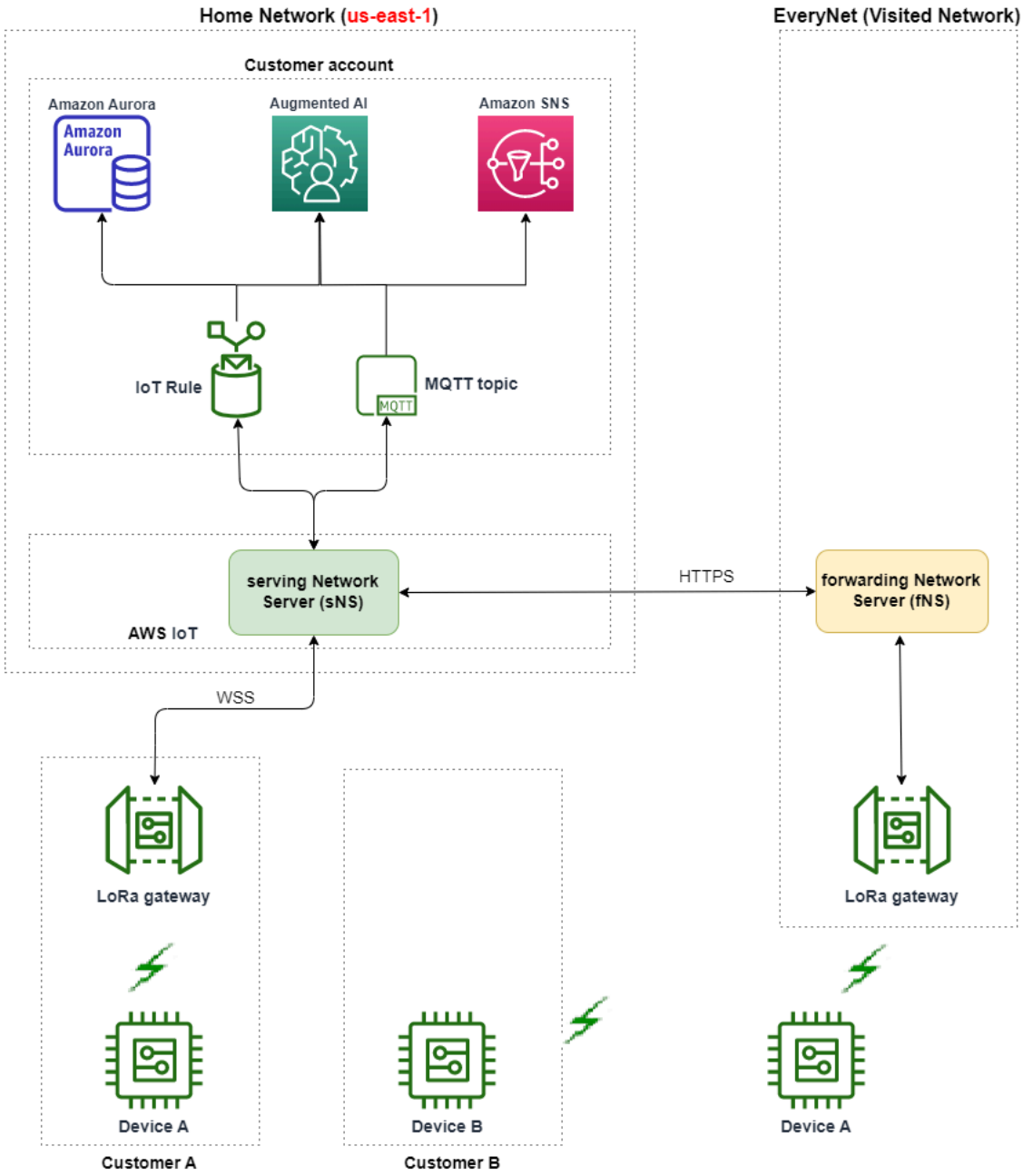
轉送網路伺服器 (fNS)

轉送網路伺服器 (fNS) 是管理無線電閘道的 LNS。一個 LoRa 工作階段中可能有零個或多個 fNS。此網路伺服器管理將從裝置收到的資料封包轉送到家用網路。

公有 LoRaWAN 網路支援架構

以下架構圖說明 AWS IoT Core for LoRaWAN 如何與 Everynet 合作提供公有網路連線。在此情況下，裝置 A 會透過 LoRa 閘道連接至 AWS IoT Core for LoRaWAN 提供的 hNS (家用網路伺服器)。當裝置 A 移動到原服務網路之外時，它便進入到訪網路，並由 Everynet 提供的到訪網路伺服器 (vNS) 提供涵蓋範圍。vNS 也將涵蓋範圍擴展到裝置 B，而裝置 B 沒有 LoRa 閘道可連線。

您可以在 AWS IoT 主控台中檢視公有網路涵蓋範圍資訊，如下節所述。



AWS IoT Core for LoRaWAN 依照 LoRa Alliance LoRaWAN 漫遊中樞技術建議，使用漫遊中心功能。漫遊中樞為 Everynet 提供端點，用於路由從終端裝置接收到的流量。在這種情況下，Everynet 充當轉送網路伺服器 (fNS)，轉送從裝置接收到的流量。它使用由 LoRa Alliance 規格定義的 HTTP RESTful API。

> **Note**
>
> 如果您的裝置從原服務網路離開並進入原服務網路和 Everynet 都能涵蓋的位置，它會使用先到先得的政策來決定是連接到 LoRa 閘道，還是連接到 Everynet 的閘道。

造訪公有網路時，hNS 和提供服務的網路伺服器 (sNS) 會加以區隔。然後在 sNS 和 hNS 之間交換上行封包。

如何使用公有網路支援

若要啟用 Everynet 的公有網路支援，請在建立服務設定檔時啟用特定漫遊參數。在此 Beta 版中，當您使用 AWS IoT Wireless API 或 AWS CLI 時，您可以使用這些參數。下列各節說明您必須啟用的參數，以及如何使用 AWS CLI 啟用公有網路。

> **Note**
>
> 您只能在建立新的服務設定檔時啟用公有網路支援。您無法使用這些參數更新現有的設定檔來啟用公有網路。

主題

- 漫遊參數
- 啟用裝置的公有網路支援

漫遊參數

請在為裝置建立服務設定檔時，指定下列參數。在從 AWS IoT 主控台的設定檔中樞或使用 AWS IoT Wireless API 操作 `CreateServiceProfile` 或 AWS CLI 命令 `create-service-profile` 新增服務設定檔時，指定這些參數。

> **Note**
>
> AWS IoT Core for LoRaWAN 不支援切換漫遊。建立服務設定檔時，您無法啟用 `HrAllowed` 參數來指定是否使用切換漫遊。

- 允許漫遊啟用 (RaAllowed)：此參數指定是否啟用漫遊啟用。漫遊啟用可讓終端裝置在 vNS 涵蓋範圍內啟動。使用漫遊功能時，RaAllowed 必須設定為 true。
- 允許被動漫遊 (PrAllowed)：此參數指定是否啟用被動漫遊。使用漫遊功能時，PrAllowed 必須設定為 true。

啟用裝置的公有網路支援

若要在您的裝置上啟用公有 LoRaWAN 網路支援，請執行下列程序。

> **Note**
>
> 您只能針對 OTAA 裝置啟用公有網路功能。使用 ABP 作為啟用方法的裝置不支援此功能。

1. 使用漫遊參數建立服務設定檔

   啟用漫遊參數以建立服務設定檔。

   > **Note**
   >
   > 當您為要與此服務設定檔產生關聯的裝置建立裝置設定檔時，建議您指定較大的 RxDelay1 參數值，至少大於 2 秒。

   - 使用 AWS IoT 主控台

     前往 AWS IoT 主控台的設定檔中樞，然後選擇新增服務設定檔。建立設定檔時，選擇啟用公有網路。

   - 使用 AWS IoT Wireless API

     若要在建立服務設定檔時啟用漫遊，請使用 CreateServiceProfile API 操作或 create-service-profile CLI 命令，如下列範例所示。

     ```
     aws iotwireless create-service-profile \
         --region us-east-1 \
         --name roamingprofile1 \
         --lorawan '{"AddGwMetadata":true,"PrAllowed":true,"RaAllowed":true}'
     ```

     執行這個命令會傳回服務設定檔的 ARN 和 ID 作為輸出。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

2. 檢查服務設定檔中的漫遊參數

   若要檢查您指定的漫遊參數，您可以在主控台中檢視服務設定檔，或使用`get-service-profile` CLI 命令，如下列範例所示。

   - 使用 AWS IoT 主控台

     前往 AWS IoT 主控台的設定檔中樞，然後選擇您建立的設定檔。在詳細資料頁面的設定檔設定索引標籤中，您會看到 RaAllowed 和 PrAllowed 設定為 `true`。

   - 使用 AWS IoT Wireless API

     若要檢視您啟用的漫遊參數，請使用 GetServiceProfile API 操作或 `get-service-profile` CLI 命令，如下列範例所示。

     ```
     aws iotwireless get-service-profile \
         --region us-east-1 \
         --id 12345678-a1b2-3c45-67d8-e90fa1b2c34d
     ```

     執行此命令會傳回服務設定檔詳細資訊作為輸出，包括漫遊參數 `RaAllowed` 和 `PrAllowed` 的值。

     ```
     {
         "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
     a1b2-3c45-67d8-e90fa1b2c34d",
         "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
         "Name": "roamingprofile1"
         "LoRaWAN": {
             "UlRate": 60,
             "UlBucketSize": 4096,
             "DlRate": 60,
             "DlBucketSize": 4096,
             "AddGwMetadata": true,
             "DevStatusReqFreq": 24,
             "ReportDevStatusBattery": false,
     ```

```
        "ReportDevStatusMargin": false,
        "DrMin": 0,
        "DrMax": 15,
        "PrAllowed": true,
        "RaAllowed": true,
        "NwkGeoLoc": false,
        "TargetPer": 5,
        "MinGwDiversity": 1
    }
}
```

3. 將服務設定檔附加至裝置

將您使用漫遊參數建立的服務設定檔附加到終端裝置。您也可以建立裝置設定檔並新增無線裝置的目的地。您會使用此目的地來路由從裝置傳送的上行訊息。如需有關建立裝置設定檔和目的地的詳細資訊，請參閱 新增裝置設定檔 和 新增目的地至 AWS IoT Core for LoRaWAN。

- 加入新裝置

  如果您尚未加入裝置，請在將裝置新增至 AWS IoT Core for LoRaWAN 時指定使用的服務設定檔。以下命令顯示如何使用 `create-wireless-device` CLI 命令，使用您建立之服務設定檔的 ID 新增裝置。如需有關使用主控台來新增服務設定檔的資訊，請參閱 使用主控台將您的無線裝置規格新增至 AWS IoT Core for LoRaWAN。

```
aws iotwireless create-wireless-device --cli-input-json file://createdevice.json
```

  下列顯示 *createdevice.json* 檔案的內容。

  createdevice.json 的內容

```
{
    "Name": "DeviceA",
    "Type": LoRaWAN,
    "DestinationName": "RoamingDestination1",
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
```

```
        "DevEui": "ac12efc654d23fc2"
    },
}
```

執行此命令的輸出會產生無線裝置的 ARN 和 ID 做為輸出。

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

- 更新現有裝置

  如果您已經加入裝置，您可以更新現有的無線裝置以使用此服務設定檔。下列命令顯示如何使用 `update-wireless-device` CLI 命令，使用您建立的服務設定檔 ID 來更新裝置。

```
aws iotwireless update-wireless-device \
    --id "1ffd32c8-8130-4194-96df-622f072a315f" \
    --service-profile-id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --description "Using roaming service profile A"
```

  此命令不會產生任何輸出。您可以使用 `GetWirelessDevice` API 或 `get-wireless-device` CLI 命令來取得更新的資訊。

4. 使用 Everynet 將裝置連線到雲端

由於漫遊已啟用，您的裝置現在必須執行聯結才能取得新的 `DevAddr`。如果您使用 OTAA，LoRaWAN 裝置會傳送聯結要求，且網路伺服器可以允許該要求。然後它就可以使用 Everynet 提供的網路涵蓋範圍連線到 AWS 雲端。如需如何為裝置執行啟用程序或聯結的指示，請參閱裝置文件。

> **Note**
>
> - 您只能為使用 OTAA 作為啟用方法的裝置啟用漫遊功能並連線至公有網路。不支援 ABP 裝置。如需如何為裝置執行啟用程序或聯結的指示，請參閱裝置文件。請參閱[啟用模式]。

- 若要停用裝置的漫遊功能，您可以取消裝置與此服務設定檔的關聯，然後將裝置與其他漫遊參數設定為 `false` 的服務設定檔產生關聯。切換至此服務設定檔後，您的裝置必須執行另一次加入，如此裝置才不會繼續在公有網路上執行。

5. 交換上行和下行訊息

   裝置加入 AWS IoT Core for LoRaWAN 後，您就可以開始在裝置和雲端之間交換訊息。

   - 檢視上行訊息

     當您從裝置傳送上行訊息時，AWS IoT Core for LoRaWAN 會使用您先前設定的目的地，將這些訊息傳遞給您的 AWS 帳戶。這些訊息將透過 Everynet 的網路從您的裝置傳送到雲端。

     您可以使用 AWS IoT 規則名稱檢視訊息，或使用 MQTT 用戶端來訂閱建立目的地時指定的 MQTT 主題。如需有關規則名稱和您指定之其他目的地詳細資訊的詳細資訊，請參閱 使用主控台新增目的地。

     如需檢視上行訊息和格式的詳細資訊，請參閱 檢視從 LoRaWAN 裝置傳送的上行訊息格式。

   - 傳送下行訊息

     您可以使用主控台，或使用 AWS IoT Wireless API 命令 `SendDataToWirelessDevice` 或 AWS CLI 命令 `send-data-to-wireless-device`，將下行訊息排入佇列，並將其傳送至裝置。如需佇列和傳送下行訊息的詳細資訊，請參閱 將要傳送至 LoRaWAN 裝置的下行訊息排入佇列。

     下列程式碼顯示如何使用 `send-data-to-wireless-device` CLI 命令傳送下行訊息的範例。您可以指定接收資料的無線裝置 ID、承載、是否使用確認模式以及無線中繼資料。

     ```
     aws iotwireless send-data-to-wireless-device \
         --id "1ffd32c8-8130-4194-96df-622f072a315f" \
         --transmit-mode "1" \
         --payload-data "SGVsbG8gVG8gRGV2c2lt" \
         --wireless-metadata LoRaWAN={FPort=1}
     ```

     執行此命令的輸出會為下行訊息產生 `MessageId`。

> **Note**
>
> 在某些情況下，即使您收到 `MessageId`，資料封包繩也可能會被丟棄。如需疑難排解此類案例及解決這些情況的相關資訊，請參閱 下行訊息佇列錯誤疑難排解。

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

- 檢視涵蓋範圍資訊

  啟用公有網路之後，您可以在 AWS IoT 主控台中檢視網路涵蓋範圍資訊。前往 AWS IoT 主控台的涵蓋範圍中樞，然後搜尋位置，即可在地圖上查看裝置的涵蓋範圍資訊。

  > **Note**
  >
  > 此功能使用 Amazon Location Service 在 Amazon Location 地圖上顯示裝置的涵蓋範圍資訊。使用 Amazon Location 地圖之前，請先檢閱 Amazon Location Service 的條款與條件。請注意，AWS 可能會將您的 API 查詢傳輸給您選擇的第三方資料提供者，該提供者可能不在您目前使用的 AWS 區域 內。如需詳細資訊，請參閱 AWS服務條款。

針對 LoRaWAN 裝置和多播群組執行無線韌體更新 (FUOTA)

您可以執行無線韌體更新來更新單一 LoRaWAN 裝置或一組裝置的裝置韌體。若要更新裝置韌體或將下行承載傳送至多個裝置，請建立多播群組。透過使用多播，來源可以將資料傳送至單一多播群組，然後再將資料分配至收件人裝置群組。

AWS IoT Core for LoRaWAN 對 FUOTA 和多點傳送群組的支援依據是 LoRa Alliance 的以下規格：

- LoRaWAN 遠端多點傳送設定規格：TS005-2.0.0
- LoRaWAN 分段資料區塊運輸規格：TS004-2.0.0
- LoRaWAN 應用程式層時鐘同步規格：TS003-2.0.0

> **Note**
>
> AWS IoT Core for LoRaWAN 會根據 LoRa Alliance 規格自動執行時鐘同步處理。透過使用函數 `AppTimeReq`，其會將伺服器端時間回覆至使用 ClockSync 訊號請求該時間的裝置。

下列主題說明如何建立多播群組和執行 FUOTA。

主題

準備好用於多點傳送和 FUOTA 組態的裝置

在將無線裝置新增至 AWS IoT Core for LoRaWAN 時，您可以使用主控台或 CLI，準備好多點傳送設定和 FUOTA 組態的無線裝置。如果是首次執行此組態，建議您使用主控台。若要管理多點傳送群組，並從群組中新增或移除多個裝置，建議您使用 CLI 來管理大量資源。

GenAppKey 和 FPorts

在新增無線裝置時，必須先設定以下參數，才能將裝置新增至多播群組或執行 FUOTA。在設定這些參數之前，請先確定裝置可支援 FUOTA 和多點傳送，並且無線裝置規格為 `OTAA v1.1` 或 `OTAAv1.0.x`。

- `GenAppKey`：若為支援 LoRaWAN 版本 1.0.x 和使用多點傳送群組的裝置，`GenAppKey` 是衍生出多點傳送群組工作階段金鑰的裝置特定根金鑰。

  > **Note**
  >
  > 若為使用無線規格的 `OTAA v1.1` 的 LoRaWAN 裝置，`AppKey` 的使用目的則與 `GenAppKey` 相同。

  若要設定參數來啟動資料傳輸，AWS IoT Core for LoRaWAN 會將工作階段金鑰分配至終端裝置。如需 LoRaWAN 版本的詳細資訊，請參閱 LoRaWAN 版本。

> **Note**
>
> AWS IoT Core for LoRaWAN 會存放以加密格式提供的 GenAppKey 資訊。

- FPorts：根據 FUOTA 的 LoRaWAN 規格和多點傳送群組，AWS IoT Core for LoRaWAN 會為以下 FPorts 參數欄位指派預設值。如果已指派以下任何一項 FPort 數值，則可從 1 到 223 中選擇一個可用的不同數值。
  - Multicast：200

    此 FPort 數值用於多點傳送群組。
  - FUOTA：201

    此 FPort 數值用於 FUOTA。
  - ClockSync：202

    此 FPort 數值用於時鐘同步處理。

多點傳送和 FUOTA 的裝置設定檔

在多點傳送工作階段開始時，B 類或 C 類分發時段會用來將下行訊息傳送至群組中的裝置上。為多點傳送和 FUOTA 新增的裝置必須可支援 B 類或 C 類操作模式。請依裝置支援的裝置類別，為已啟用 B 類或 C 類模式的裝置選擇裝置設定檔。

如需有關裝置設定檔的詳細資訊，請參閱 將設定檔新增至 AWS IoT Core for LoRaWAN。

使用主控台為多點傳送和 FUOTA 準備好裝置

若要使用主控台來指定多點傳送設定和 FUOTA 的 FPort 及 GenAppKey 參數：

1. 導覽至 AWS IoT 主控台的裝置中樞，然後選擇 Add wireless device (新增無線裝置)。
2. 選擇 Wireless device specification (無線裝置規格)。裝置必須使用 OTAA，才能啟用裝置。在選擇 OTAA v1.0.x 或 OTAA v1.1 時，系統會出現 FUOTA configuration-Optional (FUOTA 組態-選用) 區段。
3. 輸入無線裝置的延伸唯一識別碼 (EUI) 參數。
4. 展開 FUOTA configuration-Optional (FUOTA 組態-選用) 區段，然後選擇 This device supports firmware updates over the air (FUOTA) (此裝置支援無線韌體更新 (FUOTA)。此時可以輸入多點

傳送、FUOTA 和時鐘同步的 FPort 數值。如果選擇 0TAA v1.0.x 作為無線裝置規格，請輸入 GenAppKey。

5. 透過選擇設定檔和路由訊息的目的地，將裝置新增至 AWS IoT Core for LoRaWAN。若要設定連結至裝置的裝置設定檔，請選擇 Supports Class B (支援 B 類) 或 Supports Class C (支援 C 類) 模式，或同時選取這兩種模式。

> **Note**
>
> 若要指定 FUOTA 組態參數，必須使用 AWS IoT 主控台的裝置中樞。如果使用 AWS IoT 主控台的 Intro (介紹) 頁面加入裝置，則不會出現這些參數。

如需有關無線裝置規格和使用裝置的詳細資訊，請參閱 將您的無線裝置新增至 AWS IoT Core for LoRaWAN。

> **Note**
>
> 只有在建立無線裝置時，才能指定這些參數。無法在更新現有裝置時變更或指定參數。

使用 API 操作為多點傳送和 FUOTA 準備好裝置

若要使用多播群組或執行 FUOTA，請使用 `CreateWirelessDevice` API 操作或 `create-wireless-device` CLI 命令來設定這些參數。除了指定應用程式金鑰和 FPort 參數之外，請確定連結至裝置的裝置設定檔可支援 B 類或 C 類模式的其中一種，或同時支援兩者。

您可以提供 `input.json` 檔案作為 `create-wireless-device` 命令的輸入。

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

其中：

input.json 的內容

```
{
    "Description": "My LoRaWAN wireless device"
    "DestinationName": "IoTWirelessDestination"
    "LoRaWAN": {
```

```
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "FPorts": {
            "ClockSync": 202,
            "Fuota": 201,
            "Multicast": 200
      },
        "OtaaV1_0_x": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "AppEui": "b4c231a359bc2e3d",
            "GenAppKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Name": "SampleIoTWirelessThing"
    "Type": LoRaWAN
}
```

如需您可以使用哪些 CLI 命令的相關資訊，請參閱《AWS CLI 參考》。

> **Note**
>
> 在指定這些參數的值之後，便無法使用 UpdateWirelessDevice API 操作來更新參數值。不過，您可以改為使用參數 GenAppKey 和 FPorts 的值來建立新裝置。

若要取得這些參數特定值的相關資訊，您可以使用 GetWirelessDevice API 操作或 get-wireless-device CLI 命令。

後續步驟

在設定參數後，您可以建立多點傳送群組和 FUOTA 任務，以傳送下行承載或更新 LoRaWAN 裝置的韌體。

- 如需有關建立多點傳送群組的資訊，請參閱 建立多點傳送群組以及將裝置新增至群組。
- 如需有關建立 FUOTA 任務的資訊，請參閱 建立 FUOTA 任務並提供韌體映像。

建立多點傳送群組，以將下行承載傳送至多個裝置

若要將下行承載傳送至多個裝置，請建立多點傳送群組。透過使用多點傳送，來源可以將資料傳送至單一多點傳送地址，然後將資料分配至整個收件人裝置群組。

多點傳送群組中的裝置會共用相同的多點傳送地址、工作階段金鑰和框架計數器。藉由使用相同的工作階段金鑰，多點傳送群組中的裝置便可在下行傳輸啟動時解密訊息。多點傳送群組只支援下行。其不會確認裝置是否已接收下行承載。

藉由 AWS IoT Core for LoRaWAN 的多點傳送群組，您便可以：

- 使用裝置設定檔、RFRegion 或裝置類別來篩選裝置清單，然後將這些裝置新增至多點傳送群組。
- 在 48 小時分發時段內，排定和傳送一或多個下行承載訊息至多點傳送群組中的裝置。
- 在多點傳送工作階段開始時，讓裝置暫時切換到 B 類或 C 類模式以接收下行訊息。
- 監控多點傳送群組設定及其裝置狀態，並針對任何問題進行疑難排解。
- 使用「無線韌體更新」(FUOTA)，將韌體更新安全地部署到多點傳送群組的裝置中。

下列影片說明如何建立 AWS IoT Core for LoRaWAN 多播群組，以及逐步解說將裝置新增至群組並排程下行訊息至群組的程序。

以下主題會展示如何建立多點傳送群組和排定下行訊息。

主題

- 建立多點傳送群組以及將裝置新增至群組
- 監控及疑難排解多點傳送群組和群組中裝置的狀態
- 排定將下行訊息傳送至多點傳送群組中的裝置

建立多點傳送群組以及將裝置新增至群組

您可以使用主控台或 CLI 建立多點傳送群組。如果是首次建立多點傳送群組，建議您使用主控台來新增多點傳送群組。您可以使用 CLI 來管理多點傳送群組，並在群組中新增或移除裝置。

在與新增的終端裝置交換訊號後，AWS IoT Core for LoRaWAN 會使用終端裝置建立共用金鑰，並設定資料傳輸的參數。

必要條件

在可以建立多點傳送群組以及將裝置新增至群組之前：

- 透過指定 FUOTA 組態參數 `GenAppKey` 和 `FPorts`，準備好多點傳送和 FUOTA 設定的裝置。如需詳細資訊，請參閱準備好用於多點傳送和 FUOTA 組態的裝置。

- 檢查裝置是否支援 B 類或 C 類操作模式。您可以依裝置支援的裝置類別，選擇啟用了 Supports Class B (支援 B 類) 或 Supports Class C (支援 C 類)模式的裝置設定檔，或者同時啟用了這兩種模式的設定檔。如需有關裝置設定檔的詳細資訊，請參閱 將設定檔新增至 AWS IoT Core for LoRaWAN。

  在多點傳送工作階段開始時，B 類或 C 類分發時段會用來將下行訊息傳送至群組中的裝置上。

使用主控台建立多點傳送群組

若要使用主控台建立多點傳送群組，請前往 AWS IoT 主控台的 Multicast groups (多點傳送群組) 頁面，然後選擇 Create multicast group (建立多點傳送群組)。

1. 建立多點傳送群組

   若要建立多點傳送群組，請指定群組的多點傳送屬性和標籤。

   1. 指定多點傳送屬性

      若要指定多點傳送屬性，請輸入多點傳送群組的以下資訊。

      - 名稱：輸入多點傳送群組的唯一名稱。名稱僅可包含字母、數字、連字號和底線。不可含有空格。
      - 描述：可以選擇為多點傳送群組提供描述。描述的長度最高可達 2,048 個字元。

   2. 多點傳送群組的標籤

      可以選擇提供任何鍵/值對作為多點傳送群組的標籤。若要繼續建立多點傳送群組，請選擇 Next (下一步)。

2. 將裝置新增至多點傳送群組

   可以將個別裝置或裝置群組新增至多點傳送群組。若要新增裝置：

   1. 指定 RFRegion

      指定 RFRegion 或多點傳送群組的頻帶。多點傳送群組的 RFRegion 必須符合新增至多點傳送群組的裝置 RFRegion。如需有關 RFRegion 的詳細資訊，請參閱 考慮為閘道和裝置連線選取 LoRa 頻帶。

   2. 選取多點傳送裝置類別

      選擇是否讓多點傳送群組中的裝置在多點傳送工作階段開始時切換至 B 類或 C 類模式。B 類工作階段可以在一般下行槽上接收下行訊息，而 C 類工作階段則可以隨時接收下行訊息。

3. 選擇要新增至群組的裝置。

   選擇要個別還是大量新增裝置至多點傳送群組。

   - 若要個別新增裝置，請輸入要新增至群組的各裝置無線裝置 ID。
   - 若要大量新增裝置，可以依據裝置設定檔或標籤來篩選要新增的裝置。若要採用裝置設定檔，可以新增具有可支援 B 類或 C 類或這兩類的裝置設定檔的裝置。

4. 若要建立多點傳送群組，請選擇 Create (建立)。

   多點傳送群組詳細資料和新增的裝置會出現在群組中。如需有關多點傳送群組和裝置狀態的資訊，以及疑難排解任何問題的相關資訊，請參閱 監控及疑難排解多點傳送群組和群組中裝置的狀態。

在建立多點傳送群組之後，可以選擇 Action (動作)，以編輯、刪除或新增裝置至多點傳送群組。在新增裝置後，便可排定工作階段，將下行承載傳送至群組中的裝置。

使用 API 來建立多點傳送群組

若要使用 API 建立多點傳送群組，並將裝置新增至群組：

1. 建立多點傳送群組

   若要建立多點傳送群組，請使用 CreateMulticastGroup API 操作或 create-multicast-group CLI 命令。您可以提供 input.json 檔案作為 create-multicast-group 命令的輸入。

```
aws iotwireless create-multicast-group \
    --cli-input-json file://input.json
```

   其中：

   input.json 的內容

```
{
   "Description": "Multicast group to send downlink payload and perform FUOTA.",
   "LoRaWAN": {
      "DlClass": "ClassB",
      "RfRegion": "US915"
   },
   "Name": "MC_group_FUOTA"
```

```
}
```

您建立多點傳送群組之後，便可使用以下 API 操作或 CLI 命令來更新、刪除或取得多點傳送群組的相關資訊。

- UpdateMulticastGroup 或 update-multicast-group
- GetMulticastGroup 或 get-multicast-group
- ListMulticastGroups 或 list-multicast-groups
- DeleteMulticastGroup 或 delete-multicast-group

2. 將裝置新增至多點傳送群組

可以個別或大量新增裝置至多點傳送群組。

- 若要大量新增裝置至多點傳送群組，請使用 StartBulkAssociateWirelessDeviceWithMulticastGroup API 操作或 start-bulk-associate-wireless-device-with-multicast-group CLI 命令。若要篩選要大量關聯至多點傳送群組的裝置，請提供查詢字串。以下內容會顯示如何新增裝置群組，該群組具有一組與特定 ID 連結的裝置設定檔。

```
aws iotwireless start-bulk-associate-wireless-device-with-multicast-group \
    --id "12abd34e-5f67-89c2-9293-593b1bd862e0" \
    --cli-input-json file://input.json
```

其中：

input.json 的內容

```
{
     "QueryString": "DeviceProfileName: MyWirelessDevice AND DeviceProfileId:
 d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf",
     "Tags": [
        {
            "Key": "Multicast",
            "Value": "ClassB"
        }
    ]
}
```

此處，multicast-groups/d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf/bulk 是用來將裝置與群組建立關聯的 URL。

- 若要個別新增裝置至多點傳送群組，請使用 AssociateWirelessDeviceWithMulticastGroup API 操作或 associate-wireless-device-with-multicast-group CLI 。為要新增至群組的每個裝置提供無線裝置 ID。

```
aws iotwireless associate-wireless-device-with-multicast-group \
    --id "12abd34e-5f67-89c2-9293-593b1bd862e0" \
    --wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
```

在建立多點傳送群組之後，便可使用以下 API 操作或 CLI 命令來取得多點傳送群組的相關資訊，或取消關聯裝置。

- DisassociateWirelessDeviceFromMulticastGroup 或 disassociate-wireless-device-from-multicast-group
- StartBulkDisassociateWirelessDeviceFromMulticastGroup 或 start-bulk-disassociate-wireless-device-from-multicast-group
- ListWirelessDevices 或 list-wireless-devices

> **Note**
>
> ListWirelessDevices API 操作可用來列出一般無線裝置，以及與多點傳送群組或 FUOTA 任務相關聯的無線裝置。
>
> - 若要列出與多點傳送群組相關聯的無線裝置，請使用 ListWirelessDevices API 操作搭配 MulticastGroupID 作為篩選條件。
> - 若要列出與 FUOTA 任務相關聯的無線裝置，請使用 ListWirelessDevices API 操作搭配 FuotaTaskID 作為篩選條件。

後續步驟

在建立多點傳送群組並新增裝置之後，便可繼續新增裝置，並監控多點傳送群組和裝置的狀態。如果已成功將裝置新增至群組中，則可以設定並排定傳送下行訊息至裝置。在傳送下行訊息之前，裝置狀態必須為 Multicast setup ready (多點傳送設定就緒)。在排定下行訊息後，該狀態會變更為 Session attempting (工作階段嘗試中)。如需詳細資訊，請參閱排定將下行訊息傳送至多點傳送群組中的裝置。

如果要更新多點傳送群組中裝置的韌體，可以使用 AWS IoT Core for LoRaWAN 執行無線韌體更新 (FUOTA)。如需詳細資訊，請參閱AWS IoT Core for LoRaWAN 裝置的無線韌體更新 (FUOTA)。

如果尚未新增裝置，或是在多點傳送群組或裝置狀態中看到錯誤訊息，將滑鼠游標移至錯誤訊息上方，即可取得更多資訊並解決問題。若仍能看到錯誤訊息，且需要如何疑難排解和解決問題的詳細資訊，請參閱 監控及疑難排解多點傳送群組和群組中裝置的狀態。

監控及疑難排解多點傳送群組和群組中裝置的狀態

在新增裝置並建立多點傳送群組之後，請開啟 AWS Management Console。導覽至 AWS IoT 主控台的 Multicast groups (多點傳送群組) 頁面，然後選擇建立的多點傳送群組來檢視其詳細資訊。您會看到多點傳送群組的相關資訊、已新增的裝置數量，以及裝置狀態詳細資料。您可以使用狀態資訊來追蹤多點傳送工作階段的進度，並疑難排解任何錯誤。

多點傳送群組狀態

多點傳送群組可能會有以下一種顯示在 AWS Management Console 中的狀態訊息。

- 待定

  此狀態表示已建立多點傳送群組，但尚不具有多點傳送工作階段。您會在群組建立後看到系統顯示此狀態訊息。在此期間，您可以更新多點傳送群組，並將裝置與群組建立關聯或解除關聯。在狀態從 Pending (待定) 變更後，便無法將其他裝置新增至群組。
- 工作階段嘗試中

  將裝置成功新增至多點傳送群組後，您會在群組具有排定的多點傳送工作階段時看到系統顯示此狀態訊息。在此期間，您無法更新或新增裝置至多點傳送群組。如果取消多點傳送工作階段，群組狀態會變更為 Pending (待定)。
- 工作階段中

  在多點傳送工作階段處於最早的工作階段時間時，您會看到系統顯示此狀態訊息。在多點傳送群組與具有持續韌體更新工作階段的 FUOTA 任務相關聯時，該群組也會繼續處於此狀態。

  如果工作階段中沒有相關聯的 FUOTA 任務，並且多點傳送工作階段因工作階段時間超過逾時時間或取消了多點傳送工作階段而被取消，則群組狀態會變更為 Pending (待定)。
- 等待刪除中

  如果刪除多點傳送群組，其群組狀態會變更為 Delete waiting (等待刪除中)。刪除為永久刪除，且無法還原。此動作可能需要一段時間，而且在多點傳送群組遭到刪除前，群組狀態皆仍為 Delete_Waiting。在多點傳送群組進入此狀態之後，便無法轉換至其他任一種狀態。

多點傳送群組中裝置的狀態

多點傳送群組中的裝置可能會有以下一種顯示在 AWS Management Console 中的狀態訊息。將滑鼠游標暫留在每個狀態訊息上，即可取得其表示內容的詳細資訊。

- 套件嘗試中

  在裝置與多點傳送群組建立關聯之後，裝置狀態為 Package attempting (套件嘗試中)。此狀態表示 AWS IoT Core for LoRaWAN 尚未確認裝置是否支援多點傳送設定和操作。

- 不支援套件

  在裝置與多點傳送群組建立關聯之後，AWS IoT Core for LoRaWAN 會檢查裝置的韌體是否能進行多點傳送設定和操作。如果裝置沒有受支援的多點傳送套件，其狀態為 Package unsupported (不支援套件)。若要解決此錯誤，請檢查裝置的韌體是否能進行多點傳送設定和操作。

- 多點傳送安裝嘗試中

  如果與多點傳送群組相關聯的裝置能進行多點傳送設定和作業，則狀態為 Multicast setup attempting (多點傳送安裝嘗試中)。此狀態表示裝置尚未完成多點傳送設定。

- 多點傳送安裝就緒

  裝置已完成多點傳送設定，且已新增至多點傳送群組。此狀態表示裝置已準備好進行多點傳送工作階段，且可將下行訊息傳送到這些裝置。狀態也會指出何時可以使用 FUOTA 來更新群組中裝置的韌體。

- 工作階段嘗試中

  已針對多點傳送群組中的裝置，排定多點傳送工作階段。在多點傳送群組工作階段開始時，裝置狀態為 Session attempting (工作階段嘗試中)，且就可否為工作階段啟用 B 類或 C 類分發時段傳送了請求。如果設定多點傳送工作階段花費的時間超過逾時時間，或者取消多點傳送工作階段，則狀態會變更為 Multicast setup done (多點傳送設定已完成)。

- 工作階段中

  此狀態表示已啟用 B 類或 C 類分發時段，且裝置有進行中的多點傳送工作階段。在此期間，可以從 AWS IoT Core for LoRaWAN 傳送下行訊息至多點傳送群組中的裝置。如果更新工作階段時間，其會覆寫目前的工作階段，且狀態會變更為 Session attempting (工作階段嘗試中)。如果工作階段時間結束或取消多點傳送工作階段，狀態會變更為 Multicast setup ready (多點傳送設定就緒)。

後續步驟

您已經了解多點傳送群組和群組中裝置的不同狀態，以及如何疑難排解任何問題 (例如裝置無法進行多點傳送設定)，現在可以排定將下行訊息傳送至裝置，且多點傳送群組將會處於 In session (工作階段中)。如需有關排定下行訊息的資訊，請參閱 排定將下行訊息傳送至多點傳送群組中的裝置。

排定將下行訊息傳送至多點傳送群組中的裝置

在成功將裝置新增至多點傳送群組之後，便可開始多點傳送工作階段，並設定要傳送至這些裝置的下行訊息。必須將下行訊息排定在 48 小時內，且多點傳送的開始時間必須至少晚於目前時間的 30 分鐘。

> **Note**
>
> 多點傳送群組中的裝置無法在收到下行訊息時進行確認。

必要條件

必須先建立一個多點傳送群組，並成功新增裝置至要傳送下行訊息的群組，才可以傳送下行訊息。在排定多點傳送工作階段的開始時間後，您便無法新增更多裝置。如需詳細資訊，請參閱建立多點傳送群組以及將裝置新增至群組。

如果有任何裝置未能成功新增，則多點傳送群組和裝置狀態將包含有助您解決錯誤的資訊。若錯誤仍然存在，請參閱 監控及疑難排解多點傳送群組和群組中裝置的狀態 了解解決這些錯誤的相關資訊。

使用主控台排定下行訊息

若要使用主控台傳送下行訊息，請前往 AWS IoT 主控台的 Multicast groups (多點傳送群組) 頁面，然後選擇建立的多點傳送群組。在 multicast group details (多點傳送群組詳細資訊) 頁面中，選擇 Schedule downlink message (排定下行訊息)，然後選擇 Schedule downlink session (排定下行工作階段)。

1. 排定下行訊息時段

   您可以為要傳送至多點傳送群組中裝置的下行訊息，設定一個時間範圍。必須在 48 小時內排定下行訊息。

   若要排定多點傳送工作階段，請指定以下參數：

   - Start date (開始日期) 和 Start time (開始時間)：開始日期和時間必須至少為目前時間的 30 分鐘後和 48 小時前。

> **Note**
>
> 應以 UTC 指定時間，因此請在排定下行時段時考慮到時差問題。

- 工作階段逾時：若未收到任何下行訊息，則希望讓多點傳送工作階段逾時的時間。最小逾時為 60 秒。B 類多點傳送群組的最大逾時值為 2 天，C 類多點傳送群組則為 18 小時。

2. 設定下行訊息

   若要設定下行訊息，請指定以下參數：

   - 資料傳輸速率：選擇下行訊息的資料傳輸速率。資料傳輸速率依 RFRegion 和承載規模而定。US915 區域的預設資料傳輸速率是 8，EU868 區域的預設資料傳輸速率是 0。
   - 頻率：選擇傳送下行訊息的頻率。若要避免訊息衝突，請依據 RFRegion 來選擇可用頻率。
   - FPort：選擇一個可用的頻率連接埠，將下行訊息傳送至裝置。
   - 承載：根據資料傳輸速率指定承載的規模上限。若使用預設資料傳輸速率，US915 RfRegion 中的承載規模上限為 33 位元組，EU868 RfRegion 中則為 51 個位元組。若使用較高的資料傳輸速率，則承載規模上限可達 242 位元組。

   若要排定下行訊息，請選擇 Schedule (排程)。

使用 API 排定下行訊息

若要使用 API 排定下行訊息，請使用 `StartMulticastGroupSession` API 操作或 `start-multicast-group-session` CLI 命令。

您可以使用以下 API 操作或 CLI 命令來取得多點傳送群組的相關資訊或刪除多點傳送群組。

- `GetMulticastGroupSession` 或 `get-multicast-group-session`
- `DeleteMulticastGroupSession` 或 `delete-multicast-group-session`

若要在工作階段開始後將資料傳送至多點傳送群組，請使用 `SendDataToMulticastGroup` API 操作或 `send-data-to-multicast-group` CLI 命令。

後續步驟

在設定要傳送至裝置的下行訊息後，系統會於工作階段開始時傳送該訊息。多點傳送群組中的裝置無法確認是否收到訊息。

設定其他下行訊息

您還可以設定要傳送至多點傳送組中裝置的其他下行訊息：

- 若要從主控台設定其他下行訊息：
  1. 前往 AWS IoT 主控台的 Multicast groups (多點傳送群組) 頁面，然後選擇建立的多點傳送群組。
  2. 在 multicast group details (多點傳送群組詳細資訊) 頁面中，選擇 Schedule downlink message (排定下行訊息)，然後選擇 Configure additional downlink message (設定其他下行訊息)。
  3. 以為第一個下行訊息設定這些參數的相似方法，指定參數 Data rate (資料傳輸速率)、Frequency (頻率)、FPort 以及 Payload (承載)。
- 若要使用 API 或 CLI 來設定其他下行訊息，請呼叫每個其他下行訊息的 `SendDataToMulticastGroup` API 操作或 `send-data-to-multicast-group`CLI 命令。

更新工作階段排程

您也可以更新工作階段排程，為多點傳送工作階段使用新的開始日期和時間。新的工作階段排程會覆寫之前排定的工作階段。

> **Note**
>
> 僅在必要時更新多點傳送工作階段。這些更新可能會讓一組裝置長時間處於喚醒狀態，繼而耗盡電池電量。

- 若要從主控台更新工作階段排程：
  1. 前往 AWS IoT 主控台的 Multicast groups (多點傳送群組) 頁面，然後選擇建立的多點傳送群組。
  2. 在 multicast group details (多點傳送群組詳細資訊) 頁面中，選擇 Schedule downlink message (排定下行訊息)，然後選擇 Update session schedule (更新工作階段排程)。
  3. 以為第一個下行訊息指定這些參數的相似方法，指定參數 State date (開始日期)、Start time (開始時間) 以及 Session timeout (工作階段逾時)。
- 若要從 API 或 CLI 更新工作階段排程，請使用 `StartMulticastGroupSession` API 操作或 `start-multicast-group-session` CLI 命令。

AWS IoT Core for LoRaWAN 裝置的無線韌體更新 (FUOTA)

使用「無線韌體更新」(FUOTA)，將韌體更新部署到 AWS IoT Core for LoRaWAN 裝置。

您可以使用 FUOTA 將韌體更新傳送至個別裝置或裝置群組。您也可以透過建立多點傳送群組，將韌體更新傳送至多個裝置。首先將裝置新增至多點傳送群組，然後將韌體更新映像傳送至所有裝置。建議以數位方式簽署韌體映像，以便接收映像的裝置可以確認其是否來自正確來源。

使用 AWS IoT Core for LoRaWAN 的 FUOTA，您可以：

- 將新韌體映像或差異映像部署到單一裝置或裝置群組。
- 在韌體部署到裝置後驗證其真確性及完整性。
- 監控部署進度，並在部署失敗時進行問題偵錯。

AWS IoT Core for LoRaWAN 對 FUOTA 和多點傳送群組的支援依據是 LoRa Alliance 的以下規格：

- LoRaWAN 遠端多點傳送設定規格：TS005-2.0.0
- LoRaWAN 分段資料區塊運輸規格：TS004-2.0.0
- LoRaWAN 應用程式層時鐘同步規格：TS003-2.0.0

> **Note**
>
> AWS IoT Core for LoRaWAN 會根據 LoRa Alliance 規格自動執行時鐘同步處理。透過使用函數 `AppTimeReq`，其會將伺服器端時間回覆至使用 ClockSync 訊號請求該時間的裝置。

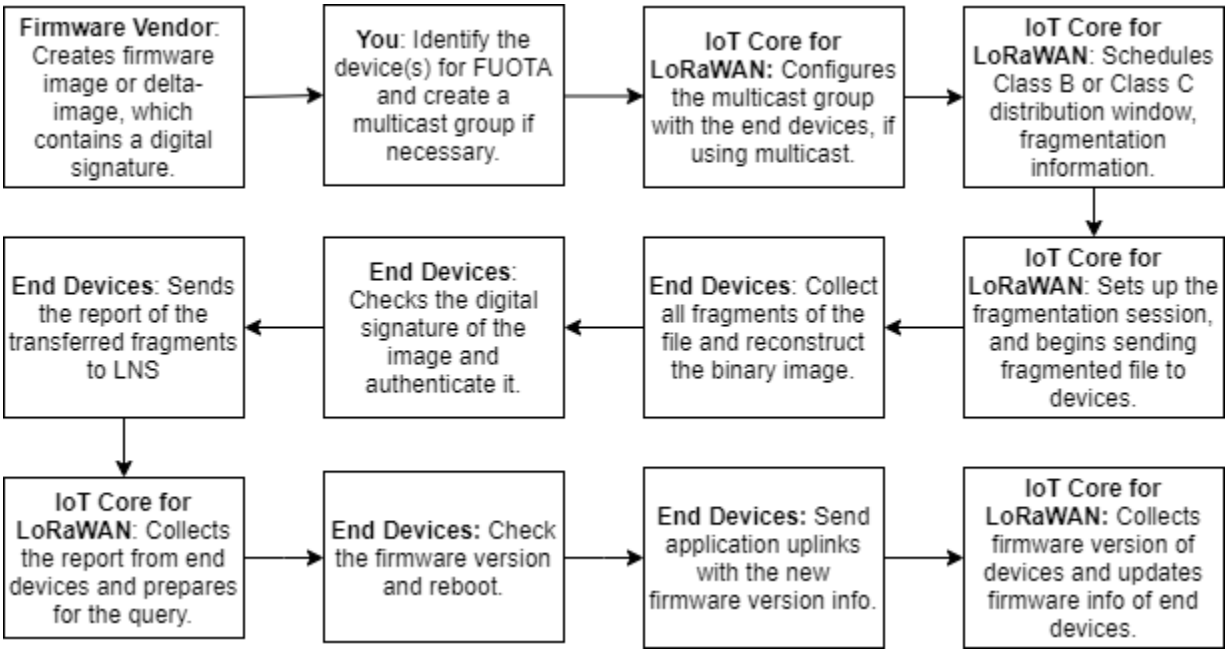
下列影片說明如何建立 AWS IoT Core for LoRaWAN FUOTA 任務，以及逐步解說新增裝置至任務和排程 FUOTA 任務的程序。

下列主題說明如何執行 FUOTA。

- FUOTA 程序概觀
- 建立 FUOTA 任務並提供韌體映像
- 將裝置和多點傳送群組新增至 FUOTA 任務，並排定 FUOTA 工作階段。
- 監控 FUOTA 任務狀態以及新增至任務的裝置並解決相關問題

FUOTA 程序概觀

以下圖表會顯示 AWS IoT Core for LoRaWAN 如何為終端裝置執行 FUOTA 程序。若要將個別裝置新增至 FUOTA 工作階段，則可略過建立和設定多點傳送群組的步驟。您可以將裝置直接新增至 FUOTA 工作階段，接著 AWS IoT Core for LoRaWAN 會開始進行韌體更新程序。



若要為裝置執行 FUOTA，請先建立數位簽署的韌體映像，並設定要新增至 FUOTA 任務的裝置和多播群組。在開始 FUOTA 工作階段後，終端裝置會收集所有片段、從片段重建映像、將狀態回報至 AWS IoT Core for LoRaWAN，然後套用新的韌體映像。

以下會說明 FUOTA 程序中的不同步驟：

1. 使用數位簽章建立韌體映像或差異映像

   若要讓 AWS IoT Core for LoRaWAN 為 LoRaWAN 裝置執行 FUOTA，建議您在傳送無線韌體更新時，對韌體映像或差異映像進行數位簽署。接著，接收映像的裝置便可確認其是否來自正確來源。

   韌體映像大小不得超過 1 MB。韌體大小越大，完成更新程序所需的時間就越長。若要更快速地傳輸資料，或者新映像大於 1 MB，請使用差異映像，其為新映像的一部分，即新韌體映像與上一個映像之間的差異。

   > **Note**
   >
   > AWS IoT Core for LoRaWAN 不會提供數位簽章產生工具和韌體版本管理系統。您可以使用任何第三方工具來產生韌體映像的數位簽章。建議您使用數位簽章工具，例如嵌入 ARM Mbed GitHub 儲存庫的工具，其還包括用來產生差異映像以及供裝置使用該映像的工具。

2. 識別和設定要進行 FUOTA 的裝置

   識別要進行 FUOTA 的裝置後，請將韌體更新傳送至個別或多個裝置。

   - 若要將韌體更新傳送至多個裝置，請建立多點傳送群組，並使用終端裝置來設定多點傳送群組。如需詳細資訊，請參閱建立多點傳送群組，以將下行承載傳送至多個裝置。
   - 若要將韌體更新傳送至個別裝置，請將這些裝置新增至 FUOTA 工作階段，然後執行韌體更新。

3. 排定分發時段並設定分段工作階段

   如果已建立多點傳送群組，則可以指定 B 類或 C 類分發時段，以便判斷裝置何時可以從 AWS IoT Core for LoRaWAN 接收片段。裝置在切換至 B 類或 C 類模式之前，可能會先在 A 類中運作。您也必須指定工作階段的開始時間。

   B 類或 C 類裝置會在指定的分發時段喚醒，並開始接收下行封包。以 C 類模式操作的裝置會比 B 類裝置消耗更多電力。如需詳細資訊，請參閱裝置類別。

4. 終端裝置會將狀態回報至 AWS IoT Core for LoRaWAN 並更新韌體映像

   設定分段工作階段之後，終端裝置和 AWS IoT Core for LoRaWAN 會執行以下步驟來更新裝置的韌體。

   1. 由於 LoRaWAN 裝置的資料傳輸速率低，因此若要開始 FUOTA 程序，AWS IoT Core for LoRaWAN 便會設定分段工作階段來為韌體映像分段。然後，其會將這些片段傳送至終端裝置。
   2. 在 AWS IoT Core for LoRaWAN 傳送映像片段後，LoRaWAN 終端裝置便會執行以下任務。
      a. 收集片段，再從這些片段中重建二進位映像。
      b. 檢查重建映像的數位簽章以驗證映像，並確認映像來自正確的來源。
      c. 將 AWS IoT Core for LoRaWAN 的韌體版本與最新版本進行比較。
      d. 回報已傳輸至 AWS IoT Core for LoRaWAN 的分段映像，然後套用新的韌體映像。

      > **Note**
      >
      > 在某些情況下，終端裝置會先回報已傳輸至 AWS IoT Core for LoRaWAN 的分段映像，再檢查韌體映像的數位簽章。

您已經了解 FUOTA 程序，現在可以建立 FUOTA 任務並將裝置新增至任務來更新其韌體。如需詳細資訊，請參閱建立 FUOTA 任務並提供韌體映像。

建立 FUOTA 任務並提供韌體映像

若要更新 LoRaWAN 裝置的韌體，請先建立 FUOTA 任務，並提供要用於更新的數位簽署韌體映像。然後，您便可將裝置和多點傳送群組新增至任務，並排定 FUOTA 工作階段。在工作階段啟動時，AWS IoT Core for LoRaWAN 會設定分段工作階段，終端裝置會收集片段、重建映像並套用新韌體。如需 FUOTA 程序的詳細資訊，請參閱 FUOTA 程序概觀。

以下內容會顯示如何建立 FUOTA 任務，以及如何上傳存放在 S3 儲存貯體中的韌體映像或差異映像。

必要條件

在執行 FUOTA 之前，必須先數位簽署韌體映像，以便終端裝置在套用映像時能驗證映像的真偽。您可以使用任何第三方工具來產生韌體映像的數位簽章。建議您使用數位簽章工具，例如嵌入 ARM Mbed GitHub 儲存庫的工具，其還包括用來產生差異映像以及供裝置使用該映像的工具。

使用主控台建立 FUOTA 任務並上傳韌體映像

若要使用主控台建立 FUOTA 任務和上傳韌體映像，請前往 FUOTA tasks (FUOTA 任務) 索引標籤，然後選擇 Create FUOTA task (建立 FUOTA 任務)。

1. 建立 FUOTA 任務

   若要建立 FUOTA 任務，請指定任務屬性和標籤。

   1. 指定 FUOTA 任務屬性

      若要指定 FUOTA 任務屬性，請輸入 FUOTA 任務的以下資訊。

      - 名稱：輸入 FUOTA 任務的唯一名稱。名稱僅可包含字母、數字、連字號和底線。不可含有空格。
      - 描述：可以選擇為多點傳送群組提供描述。描述欄位最長可達 2,048 個字元。
      - RFRegion：設定 FUOTA 任務的頻帶。頻帶必須符合用來佈建無線裝置或多點傳送群組的頻帶。

   2. FUOTA 任務的標籤

      您可以選擇提供任何鍵值對，作為 FUOTA 任務的標籤。請選擇 Next (下一步) 以繼續建立任務。

2. 上傳韌體映像

選擇要用來更新新增至 FUOTA 任務的裝置韌體的韌體映像檔案。韌體映像檔案會存放在 S3 儲存貯體中。您可以提供 AWS IoT Core for LoRaWAN 許可，讓其代表您存取韌體映像。建議您以數位方式簽署韌體映像，以便在執行韌體更新時驗證其真偽。

1. 選擇韌體映像檔案

您可以將新韌體映像檔案上傳至 S3 儲存貯體，或選擇已上傳至 S3 儲存貯體的現有映像。

> **Note**
> 韌體映像檔案大小不得超過 1 MB。韌體大小越大，完成更新程序所需的時間就越長。

- 若要使用現有映像，請選擇 Select an existing firmware image (選取現有韌體映像)，然後選擇 Browse S3 (瀏覽 S3)，再選擇要使用的韌體映像檔案。

  AWS IoT Core for LoRaWAN 會填入 S3 URL，其為前往 S3 儲存貯體中韌體映像檔案的路徑。路徑的格式為 `s3://bucket_name/file_name`。若要在 Amazon Simple Storage Service 主控台中檢視檔案，請選擇 View (檢視)。
- 上傳新韌體映像。
  a. 選擇 Upload a new firmware image (上傳新韌體映像)，然後上傳韌體映像。映像檔案大小不得超過 1 MB。
  b. 若要建立 S3 儲存貯體並輸入 Bucket name (儲存貯體名稱) 以存放韌體映像檔案，請選擇 Create S3 bucket (建立 S3 儲存貯體)。

2. 存取儲存貯體的權限

您可以建立新的服務角色或選擇現有角色，讓 AWS IoT Core for LoRaWAN 代表您存取 S3 儲存貯體中的韌體映像檔案。選擇 Next (下一步)。

若要建立新角色，您可以輸入角色名稱，或者保留空白，以便自動產生隨機名稱。若要檢視授予 S3 儲存貯體存取權限的政策許可，請選擇 View policy permissions (檢視政策許可)。

如需使用 S3 儲存貯體來存放映像以及授予 AWS IoT Core for LoRaWAN 存取權限的相關資訊，請參閱 將韌體檔案上傳至 S3 儲存貯體並新增 IAM 角色。

3. 檢閱和建立

   若要建立 FUOTA 任務，請檢閱指定的 FUOTA 任務和組態詳細資訊，然後選擇 Create task (建立任務)。

使用 API 建立 FUOTA 任務並上傳韌體映像

若要建立 FUOTA 任務並使用 API 指定韌體映像檔案，請使用 CreateFuotaTask API 操作或 create-fuota-task CLI 命令。您可以提供 input.json 檔案作為 create-fuota-task 命令的輸入。在使用 API 或 CLI 時，必須已將作為輸入提供的韌體映像檔案上傳至 S3 儲存貯體。您也可以指定 IAM 角色，授予 AWS IoT Core for LoRaWAN 存取 S3 儲存貯體中韌體映像的權限。

```
aws iotwireless create-fuota-task \
    --cli-input-json file://input.json
```

其中：

input.json 的內容

```
{
   "Description": "FUOTA task to update firmware of devices in multicast group.",
   "FirmwareUpdateImage": "S3:/firmware_bucket/firmware_image
   "FirmwareUpdateRole": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI"
   "LoRaWAN": {
      "RfRegion": "US915"
   },
   "Name": "FUOTA_Task_MC"
}
```

在建立 FUOTA 任務之後，便可使用以下 API 操作或 CLI 命令來更新、刪除或取得 FUOTA 任務的相關資訊。

- UpdateFuotaTask 或 update-fuota-task
- GetFuotaTask 或 get-fuota-task
- ListFuotaTasks 或 list-fuota-tasks
- DeleteFuotaTask 或 delete-fuota-task

後續步驟

您已建立 FUOTA 任務和提供韌體映像，因此可以將裝置新增至任務以更新其韌體。可以將個別裝置或多點傳送群組新增至任務。如需詳細資訊，請參閱將裝置和多點傳送群組新增至 FUOTA 任務，並排定 FUOTA 工作階段。。

將裝置和多點傳送群組新增至 FUOTA 任務，並排定 FUOTA 工作階段。

在建立 FUOTA 任務之後，便可將裝置新增至要更新韌體的任務。在成功將裝置新增至 FUOTA 任務後，便可排定 FUOTA 工作階段來更新裝置韌體。

- 如果只有少量裝置，則可以將這些裝置直接新增至 FUOTA 任務。
- 如果有大量要更新韌體的裝置，則可將這些裝置新增至多點傳送群組，然後將多點傳送群組新增至 FUOTA 任務。如需有關建立和使用多點傳送群組的資訊，請參閱 建立多點傳送群組，以將下行承載傳送至多個裝置。

> **Note**
>
> 可以將個別裝置或多點傳送群組擇一新增至 FUOTA 任務。您無法將裝置和多點傳送群組同時新增至任務。

在新增裝置或多點傳送群組之後，便可開始韌體更新工作階段。AWS IoT Core for LoRaWAN 會收集韌體映像、將映像分段，然後以加密格式存放片段。終端裝置會收集片段並套用新的韌體映像。韌體更新所需時間依映像大小以及映像分段方式而定。在韌體完成更新後，系統便會刪除 AWS IoT Core for LoRaWAN 存放的韌體映像加密片段。您仍可在 S3 儲存貯體中找到韌體映像。

必要條件

在可以將裝置或多點傳送群組新增至 FUOTA 任務之前，請先執行以下動作。

- 必須已建立 FUOTA 任務並提供韌體映像。如需詳細資訊，請參閱建立 FUOTA 任務並提供韌體映像。
- 佈建要更新其裝置韌體的無線裝置。如需加入裝置的相關資訊，請參閱 將裝置加入 AWS IoT Core for LoRaWAN。
- 若要更新多個裝置的韌體，可以將其新增至多點傳送群組。如需詳細資訊，請參閱建立多點傳送群組，以將下行承載傳送至多個裝置。

- 在將裝置加入 AWS IoT Core for LoRaWAN 時，請指定 FUOTA 組態參數 `FPorts`。如果使用 LoRaWAN v1.0.x 裝置，則也必須指定 `GenAppKey`。如需有關 FUOTA 組態參數的詳細資訊，請參閱 準備好用於多點傳送和 FUOTA 組態的裝置。

使用主控台將裝置新增至 FUOTA 任務，並排定 FUOTA 工作階段

若要使用主控台來新增裝置或多點傳送群組，並排程 FUOTA 工作階段，請前往主控台的 FUOTA tasks (FUOTA 任務) 索引標籤。然後，選擇為其新增裝置的 FUOTA 任務，並執行韌體更新。

新增裝置和多點傳送群組

1. 可以選擇將個別裝置或多點傳送群組新增至 FUOTA 任務。不過，無法將個別裝置和多點傳送群組同時新增至相同的 FUOTA 任務。依照以下程序使用主控台來新增裝置。

   1. 在 FUOTA task details (FUOTA 任務詳細資訊) 中，選擇 Add device (新增裝置)。
   2. 為新增至任務的裝置選擇頻帶或 RFRegion。此數值必須符合您為 FUOTA 任務選擇的 RFRegion。
   3. 選擇要將個別還是多點傳送群組新增至任務。
      - 若要新增個別裝置，請選擇 Add individual devices (新增個別裝置)，然後輸入要新增至 FUOTA 任務之各裝置的裝置 ID。
      - 若要新增多點傳送群組，請選擇 Add multicast groups (新增多點傳送群組)，然後將多點傳送群組新增至任務。您可以使用裝置設定檔或標籤來篩選要新增至任務的多點傳送群組。在依據裝置設定檔進行篩選時，可以選擇啟用了 Supports Class B (支援 B 類) 或 Supports Class C (支援 C 類) 裝置設定檔的多點傳送群組。
2. 排定 FUOTA 工作階段

   在成功新增裝置或多點傳送群組之後，便可排定 FUOTA 工作階段。若要排定工作階段，請執行以下動作。

   1. 選擇要為其更新裝置韌體的 FUOTA 任務，然後選擇 Schedule FUOTA session (排定 FUOTA 工作階段)。
   2. 指定 FUOTA 工作階段的 Start date (開始日期) 和 Start time (開始時間)。請確定開始時間是目前時間之後的 30 分鐘或更晚時間。

使用 API 將裝置新增至 FUOTA 任務，並排定 FUOTA 工作階段

您可以使用 AWS IoT Wireless API 或 CLI，將無線裝置或多點傳送群組新增至 FUOTA 任務。然後，便可排定 FUOTA 工作階段。

1. 新增裝置和多點傳送群組

   您可以將無線裝置或多點傳送群組與 FUOTA 任務建立關聯。

   - 請使用 AssociateWirelessDeviceWithFuotaTask API 操作或 associate-wireless-device-with-fuota-task CLI 命令來為個別裝置與 FUOTA 任務建立關聯，並提供 `WirelessDeviceID` 作為輸入。

     ```
     aws iotwireless associate-wireless-device-with-fuota-task \
         --id "01a23cde-5678-4a5b-ab1d-33456808ecb2"
         --wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
     ```

   - 請使用 AssociateMulticastGroupWithFuotaTask API 操作或 associate-multicast-group-with-fuota-task CLI 命令來為多點傳送群組與 FUOTA 任務建立關聯，並提供 `MulticastGroupID` 作為輸入。

     ```
     aws iotwireless associate-multicast-group-with-FUOTA-task \
         --id 01a23cde-5678-4a5b-ab1d-33456808ecb2"
         --multicast-group-id
     ```

   在將無線裝置或多點傳送群組關聯至 FUOTA 任務後，請使用以下 API 操作或 CLI 命令來列出裝置或多點傳送群組，或取消其與任務的關聯。

   - DisassociateWirelessDeviceFromFuotaTask 或 disassociate-wireless-device-from-fuota-task
   - DisassociateMulticastGroupFromFuotaTask 或 disassociate-multicast-group-from-fuota-task
   - ListWirelessDevices 或 list-wireless-devices
   - ListMulticastGroups 或 list-multicast-groups-by-fuota-task

     > **Note**
     >
     > API：

- 在將 MulticastGroupID 作為篩選條件使用時，ListWirelessDevices 會列出一般無線裝置，以及與多點傳送群組相關聯的裝置。在將 FuotaTaskID 作為篩選條件使用時，API 會列出與 FUOTA 任務相關聯的無線裝置。
- 在將 FuotaTaskID 作為篩選條件使用時，ListMulticastGroups 會列出一般多點傳送群組，以及與 FUOTA 任務相關聯的多點傳送群組。

2. 排定 FUOTA 工作階段

   在成功將裝置或多點傳送群組新增至 FUOTA 任務後，便可開始 FUOTA 工作階段來更新裝置韌體。開始時間必須為目前時間之後的 30 分鐘或更晚時間。若要使用 API 或 CLI 排定 FUOTA 工作階段，請使用 StartFuotaTask API 操作或 start-fuota-task CLI 命令。

   在開始 FUOTA 工作階段之後，便無法再將裝置或多點傳送群組新增至任務。可以使用 GetFuotaTask API 操作或 get-fuota-task CLI 命令來取得有關 FUOTA 工作階段的資訊。

監控 FUOTA 任務狀態以及新增至任務的裝置並解決相關問題

在佈建無線裝置並建立可能要使用的任何多點傳送群組之後，便可以執行以下步驟來開始 FUOTA 工作階段。

FUOTA 任務狀態

FUOTA 任務可能會有以下一種顯示在 AWS Management Console 中的狀態訊息。

- 待定

  此狀態表示已建立 FUOTA 任務，但尚不具有韌體更新工作階段。您會在任務建立後看到系統顯示此狀態訊息。在此期間，您可以更新 FUOTA 任務，並將裝置或多點傳送組與任務建立關聯或解除關聯。在狀態從 Pending (待定) 變更後，便無法將其他裝置新增至任務。

- FUOTA 工作階段等待中

  在將裝置成功新增至 FUOTA 任務後，您會在任務具有排定的韌體更新工作階段時看到系統顯示此狀態訊息。在此期間，您無法更新或新增裝置至 FUOTA 工作階段。如果取消 FUOTA 工作階段，群組狀態會變更為 Pending (待定)。

- 在 FUOTA 工作階段中

  在 FUOTA 工作階段開始時，您會看到系統顯示此狀態訊息。分段工作階段開始運作，終端裝置會收集片段、重建韌體映像、比較新韌體版本和原始版本，以及套用新映像。

- FUOTA 已完成

  在終端裝置向 AWS IoT Core for LoRaWAN 回報已套用新韌體映像後，或在工作階段逾時時，系統會將 FUOTA 工作階段標示為已完成並會顯示此狀態。

  您也會在已下任一種情況下看到此狀態，因此請務必檢查韌體更新是否已正確套用至裝置。

  - 在 FUOTA 任務狀態為 FUOTA session waiting (FUOTA 工作階段等待中)，且發生 S3 儲存貯體錯誤 (例如 S3 儲存貯體中的映像檔案連結不正確，或 AWS IoT Core for LoRaWAN 沒有足夠的許可來存取儲存貯體中的檔案) 時。
  - 在 FUOTA 任務狀態為 FUOTA session waiting (FUOTA 工作階段等待中)，且具有開始 FUOTA 工作階段的請求，但 FUOTA 任務中未收到來自裝置或多點傳送群組的回應時。
  - 在 FUOTA 任務狀態為 FUOTA session waiting (FUOTA 工作階段等待中)，且裝置或多點傳送群組在特定期間內未傳送任何片段以至於工作階段逾時的時候。
- 等待刪除中

  如果刪除處於任何其他狀態的 FUOTA 任務，系統會顯示此狀態。刪除動作為永久性動作，且無法還原。此動作可能需要一段時間，並且在 FUOTA 任務刪除前，任務狀態仍為 Delete waiting (等待刪除中)。在 FUOTA 任務進入此狀態之後，便無法轉換至其他狀態。

FUOTA 任務中裝置的狀態

FUOTA 任務中的裝置可能會有以下一種顯示在 AWS Management Console 中的狀態訊息。將滑鼠游標暫留在每個狀態訊息上，即可取得其表示內容的詳細資訊。

- 初始

  在達 FUOTA 工作階段的開始時間時，AWS IoT Core for LoRaWAN 會檢查裝置是否具有受支援的韌體更新套件。如果裝置具有受支援的套件，便會開始裝置的 FUOTA 工作階段。韌體映像會分段，而片段會傳送到裝置。在系統顯示此狀態時，即表示裝置的 FUOTA 工作階段尚未開始。
- 不支援套件

  如果裝置不具有受支援的 FUOTA 套件，系統會顯示此狀態。如果不支援韌體更新套件，裝置的 FUOTA 工作階段便無法開始。若要解決此錯誤，請檢查裝置的韌體是否可以使用 FUOTA 接收韌體更新。
- 不支援分段演算法

在 FUOTA 工作階段開始時，AWS IoT Core for LoRaWAN 會為裝置設定分段工作階段。如果系統顯示此狀態，即表示使用的分段演算法類型無法套用於裝置的韌體更新。發生此錯誤是因為裝置沒有受支援的 FUOTA 套件。若要解決此錯誤，請檢查裝置的韌體是否可以使用 FUOTA 接收韌體更新。

- 記憶體不足

  在 AWS IoT Core for LoRaWAN 傳送映像片段後，終端裝置會收集映像片段，並從這些片段中重建二進位映像。如果裝置沒有足夠的記憶體來組裝韌體映像的傳入片段，系統便會顯示此狀態，而此情況可能導致韌體更新工作階段提前結束。若要解決此錯誤，請檢查裝置的硬體是否可接收此更新。如果裝置無法接收此更新，請使用差異映像來更新韌體。

- 不支援分段索引

  分段索引可識別四個同時可行的分段工作階段中的某一個工作階段。如果裝置不支援指定的分段索引值，系統便會顯示此狀態。若要解決此問題，請執行以下其中一或多個動作。

  - 開始裝置的新 FUOTA 任務。
  - 如果錯誤仍然出現，請從單點傳送模式切換為多點傳送模式。
  - 如果錯誤仍未解決，請檢查裝置韌體。

- 記憶體錯誤

  此狀態表示裝置在接收來自 AWS IoT Core for LoRaWAN 的片段時發生記憶體錯誤。如果發生此錯誤，裝置可能無法接收此更新。若要解決此錯誤，請檢查裝置的硬體是否可接收此更新。若有需要，請使用差異映像來更新裝置韌體。

- 錯誤描述項

  裝置不支援指定的描述項。描述項是一個欄位，用來描述將在分段工作階段期間傳輸的檔案。如果看到此錯誤，請聯絡 AWS 支援 中心。

- 重新顯示工作階段計數

  此狀態表示裝置先前已使用過此工作階段計數。若要解決錯誤，請為裝置開始新的 FUOTA 任務。

- 缺少片段

  若裝置從 AWS IoT Core for LoRaWAN 收集映像片段，其會從獨立且編碼的片段中重建新的韌體映像。如果裝置未收到所有片段，則無法重建新映像，系統會顯示此狀態。若要解決錯誤，請為裝置開始新的 FUOTA 任務。

- MIC 錯誤

在裝置從已收集片段重建新韌體映像時，其會執行 MIC (訊息完整性檢查) 來驗證映像的真偽，以及映像是否來自正確來源。如果裝置在重新組裝片段後偵測到 MIC 中有不相符的項目，便會顯示此狀態。若要解決錯誤，請為裝置開始新的 FUOTA 任務。

- 成功

  裝置的 FUOTA 工作階段已成功。

> **Note**
>
> 雖然此狀態訊息表示裝置已從片段中重建映像並已進行驗證，但在裝置回報狀態至 AWS IoT Core for LoRaWAN 時，裝置韌體可能尚未更新。檢查裝置韌體是否已更新。

後續步驟

您已了解 FUOTA 任務及其裝置的不同狀態，以及如何疑難排解相關問題。如需有關上述各種狀態的詳細資訊，請參閱 LoRaWAN 分段資料區塊傳輸規格：TS004-1.0.0。

使用網路分析器即時監控無線資源機群

網路分析儀使用預設的 WebSocket 連線來接收無線連線資源的即時追蹤訊息日誌。藉由使用網路分析器，您可以新增要監控的資源、啟用追蹤訊息工作階段，以及即時開始接收追蹤訊息。

您也可以使用 Amazon CloudWatch 來監控資源。若要使用 CloudWatch，則可以設定 IAM 角色來設定記錄，然後等待主控台中顯示日誌項目。網路分析器可大幅減少設定連線並開始接收追蹤訊息所需的時間，能夠為您的資源叢集提供即時日誌資訊。如需使用 CloudWatch 進行監控的詳細資訊，請參閱 使用 Amazon CloudWatch Logs 監控您的 AWS IoT Wireless 資源。

藉由縮短設定時間並使用來自追蹤訊息的資訊，可以更有效地監控資源、取得有意義的洞察以及排解錯誤。您可以監控 LoRaWAN 裝置和 LoRaWAN 閘道。例如，可以在加入一個 LoRaWAN 裝置時，快速識別聯結錯誤。若要偵錯，請使用提供的追蹤訊息日誌中的資訊。

如何使用網路分析器

若要監控資源叢集並開始接收追蹤訊息，請執行下列步驟

1. 建立網路分析器組態並新增資源

請先建立網路分析器組態，並將資源新增至組態，才能啟用追蹤訊息。首先，指定組態設定，其中包括日誌層級和無線裝置框架資訊。然後使用無線閘道和無線裝置識別碼新增要監控的無線資源。

2. 使用 WebSockets 串流追蹤訊息

   您可以使用 IAM 角色的認證來產生預先簽署的請求 URL，以使用 WebSocket 通訊協定串流網路分析器追蹤訊息。

3. 啟用追蹤訊息工作階段並監控追蹤訊息

   若要開始接收追蹤訊息，請啟用追蹤訊息工作階段。若要避免產生額外的成本，您可以停用或關閉網路分析器追蹤訊息工作階段。

下列影片說明 AWS IoT Core for LoRaWAN 網路分析器的運作方式，以及逐步解說使用網路分析器新增資源和追蹤聯結活動的程序。

下列主題說明如何建立組態、新增資源和啟用追蹤訊息工作階段。

主題

為網路分析器加入必要的 IAM 角色

使用網路分析器時，必須授予使用者許可使用 API 操作 UpdateNetworkAnalyzerConfiguration 和 GetNetworkAnalyzerConfiguration 存取網路分析器資源。以下顯示您用於授予許可的 IAM 政策。

適用於網路分析器的 IAM 政策

使用下列任何一項：

- 完整存取無線政策

  授予 AWS IoT Core for LoRaWAN 完整存取政策，方法是將 AWSIoTWirelessFullAccess 政策連接至您的角色。如需詳細資訊，請參閱 AWSIoTWirelessFullAccess 政策摘要。

- 用於取得和更新 API 的範圍 IAM 政策

  建立以下 IAM 政策，方法是前往 IAM 主控台的 Create policy (建立政策) 頁面，並在 Visual editor (視覺化編輯器) 索引標籤：

  1. Service (服務) 選擇 IoTWireless。
  2. 在 Access level (存取層級) 下，展開 Read (讀)，選擇 GetNetworkAnalyzerConfiguration，然後展開 Write (寫)，選擇 UpdateNetworkAnalyzerConfiguration。
  3. 選擇 Next:Tags (下一步：標籤)，然後輸入政策 Name (名稱)，例如 IoTWirelessNetworkAnalyzerPolicy。選擇建立政策。

  下列顯示您建立的政策 IoTWirelessNetworkAnalyzerPolicy。如需建立政策的詳細資訊，請參閱建立 IAM 政策。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

用於存取特定資源的範圍政策

若要設定更精細的存取控制，您必須將無線閘道和裝置新增至 Resource (資源) 欄位。以下政策使用萬用字元 ARN 授予對所有閘道和裝置的存取權限。您可以使用 `WirelessGatewayId` 和 `WirelessDeviceId` 控制對特定閘道和裝置的存取。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
```

```
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": [
                "arn:aws:iotwireless:*:{accountId}:WirelessDevice/*",
                "arn:aws:iotwireless:*:{accountId}:WirelessGateway/*",
                "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
            ]
        }
    ]
}
```

若要授予使用者使用網路分析器但不使用任何無線閘道或裝置的許可，請使用以下政策。除非指定，暗中拒絕使用資源的許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": [
                "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
            ]
        }
    ]
}
```

後續步驟

現在您已建立政策，您可以將資源加入網路分析器組態，並接收這些資源的追蹤訊息資訊。如需詳細資訊，請參閱建立網路分析器組態並新增資源。

建立網路分析器組態並新增資源

在串流傳輸追蹤訊息之前，請先建立網路分析器組態，並將要監控的資源加入這個組態。建立組態時，您可以：

- 指定組態名稱和選用描述。
- 自訂組態設定，例如框架資訊和日誌訊息的詳細資訊層級。
- 加入希望監控的資源。資源可以是無線裝置、無線閘道或兩者皆是。

您指定的組態設定會決定您將針對加入組態中的資源接收到的追蹤訊息資訊。您可能也需要建立多個組態，具體取決於您的監控使用案例。

以下說明如何建立組態並新增資源。

主題

建立網路分析器組態

您必須先建立網路分析器組態，才能監控無線閘道或無線裝置。建立組態時，您只需要指定組態名稱。您可以自訂您的組態設定，並將您想要監控的資源新增至組態，即使組態已建立之後也是如此。組態設定會決定您將針對這些資源接收到的追蹤訊息資訊。

您可能需要建立多個組態，具體取決於要監控的資源以及您希望接收的資訊層級。例如，您可以建立一個組態，僅顯示 AWS 帳戶 中一組閘道的錯誤資訊。您也可以建立一個組態，顯示要監控之無線裝置的所有資訊。

以下部分介紹各種組態設定以及如何建立組態。

組態設定

建立或更新網路分析器組態時，您也可以自訂下列參數，以篩選日誌串流資訊。

- 框架資訊

  這個設定是要追蹤訊息之無線裝置資源的框架資訊。框架資訊可用來偵錯網路伺服器與終端裝置之間的通訊。依預設會啟用此功能。

- 日誌層級

  您可以檢視 Info (資訊) 或 Error (錯誤) 日誌，也可以關閉記錄功能。

  - Info

日誌層級為 Info (資訊) 的日誌更詳細，並且同時包含錯誤日誌串流和資訊日誌串流。資訊日誌可用來檢視裝置或閘道狀態的變更。

> **Note**
>
> 收集更詳細的日誌串流會產生額外的成本。如需定價的詳細資訊，請參閱 AWS IoT Core 定價。

- 錯誤

  日誌層級為 Error (錯誤) 的日誌不夠詳細且只會顯示錯誤資訊。當應用程式發生錯誤 (例如裝置連線錯誤) 時，您就可以使用這些日誌。藉由使用來自日誌串流的資訊，您可以識別並排解機群中資源的錯誤。

使用主控台建立組態

您可以建立網路分析器組態，並使用 AWS IoT 主控台或 AWS IoT Wireless API 自訂選用參數。您也可以建立多個組態，之後刪除不再使用的任何組態。

建立網路分析器組態

1. 開啟 AWS IoT 主控台的網路分析器中樞並選擇 Create configuration (建立組態)。
2. 指定組態設定
   - 名稱、描述和標籤

     指定一個唯一的組態名稱，只能含有字母、數字、連字號或底線。使用選用的 Description (描述) 欄位以提供有關組態的資訊，以及 Tags (標籤) 欄位加入有關組態之中繼資料的鍵值對。如需有關命名和描述資源的詳細資訊，請參閱 描述您的 AWS IoT Wireless 資源。
   - 組態設定

     選擇是否停用框架資訊並使用 Selelct log levels (選取日誌層級)，以此選擇要用於追蹤訊息的日誌層級。選擇 Next (下一步)。
3. 新增資源到組態。您可以立即加入資源，也可以選擇 Create (建立)，然後之後加入您的資源。若要之後加入資源，請選擇 Create (建立)。

   在 Network Analyzer hub page (網路分析器中樞頁面)，您會看到您所建立的組態及其設定。若要檢視新組態的詳細資訊，請選擇組態名稱。

刪除您的網路分析器組態

您可以建立多個網路分析器組態，具體取決於要監控的資源，以及您希望接收的追蹤訊息資訊層級。

從主控台中移除組態

1. 前往 AWS IoT 主控台的網路分析器中樞並選擇您要移除的組態。
2. 選擇動作，然後選擇刪除。

使用 API 建立組態

若要使用 API 建立網路分析器組態，請使用 CreateNetworkAnalyzerConfiguration API 操作或 create-network-analyzer-configuration CLI 命令。

建立組態時，您只需要指定組態名稱。您也可以使用此 API 操作指定組態設定，並在建立組態時加入資源。或者，您可以稍後使用 UpdateNetworkAnalyzerConfiguration API 操作或 update-network-analyzer-configuration CLI 命令。

- 建立組態

  建立組態時，您必須指定名稱。例如，下列命令只提供名稱和選用描述來建立組態。預設情況下，組態已啟用框架資訊，並使用 INFO 日誌層級。

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_Network_Analyzer_Config \
    --description "My first network analyzer configuration"
```

  執行這個命令會顯示網路分析器組態的 ARN 和 ID。

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-
e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

- 使用資源建立組態

  若要自訂組態設定，請使用trace-content 參數。若要新增資源，請使用 WirelessDevices 和 WirelessGateways 參數指定您要新增至組態的閘道、裝置或者兩者。例如，以下命令可自訂組態

設定，並將無線資源加入您的組態，這些資源由其 WirelessGatewayID 和 WirelessDeviceID 指定。

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_NetworkAnalyzer_Config \
    --trace-content WirelessDeviceFrameInfo=DISABLED,LogLevel="ERROR" \
    --wireless-gateways "12345678-a1b2-3c45-67d8-e90fa1b2c34d" "90123456-
de1f-2b3b-4c5c-bb1112223cd1"
    --wireless-devices "1ffd32c8-8130-4194-96df-622f072a315f"
```

以下範例顯示執行命令的輸出：

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-
e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

列出網路分析器組態

您可以建立多個網路分析器組態，具體取決於要監控的資源，以及您希望接收的追蹤訊息資訊詳細程度。建立這些組態後，您可以使用 ListNetworkAnalyzerConfigurations API 操作或 list-network-analyzer-configuration CLI 命令取得這些組態的清單。

```
aws iotwireless list-network-analyzer-configurations
```

執行這個命令會顯示您 AWS 帳戶 中的所有網路分析器組態。您也可以使用 max-results 參數指定要顯示多少組態。以下顯示執行這個命令的輸出。

```
{
   "NetworkAnalyzerConfigurationList": [
      {
         "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
         "Name": "My_Network_Analyzer_Config1"
      },
      {
         "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/90123456-a1a2-9a87-65b4-c12bf3c2d09a",
```

```
        "Name": "My_Network_Analyzer_Config2"
      }
   ]
}
```

刪除您的網路分析器組態

您可以用 DeleteNetworkAnalyzerConfiguration API 操作或 delete-network-analyzer-configuration CLI 命令刪除不再使用的組態。

```
aws iotwireless delete-network-analyzer-configuration \
    --configuration-name My_NetworkAnalyzer_Config
```

執行這個命令不會產生任何輸出。若要檢視可用的組態，您可以使用 ListNetworkAnalyzerConfigurations API 操作。

後續步驟

現在您已建立網路分析器組態，可以在組態加入資源或更新組態設定。如需詳細資訊，請參閱新增資源並更新網路分析器組態。

新增資源並更新網路分析器組態

必須先將資源新增至組態，才能啟動追蹤訊息。您只能使用一個預設網路分析器組態。AWS IoT Core for LoRaWAN 會將名稱 NetworkAnalyzerConfig_Default 指派給此組態，且此欄位無法編輯。在從主控台使用網路分析器時，此組態會自動新增至您的 AWS 帳戶。

您可以新增想要監控的資源至此預設組態。資源可以是 LoRaWAN 裝置和 LoRaWAN 閘道。若要將每個資源新增至組態，請使用無線閘道和無線裝置識別碼。

組態設定

若要進行設定，請先將資源新增至預設組態，然後啟用追蹤訊息。收到追蹤訊息日誌之後，您也可以自訂下列參數，以更新預設組態並篩選日誌串流。

- 框架資訊

  此設定是追蹤訊息用無線裝置資源的框架資訊。框架資訊預設為啟用，可用來偵錯網路伺服器與終端裝置之間的通訊。
- 日誌層級

  您可以檢視 Info (資訊) 或 Error (錯誤) 日誌，也可以關閉記錄功能。

- Info

  日誌層級為 Info (資訊) 的日誌更詳細，並且包含資訊豐富且提供錯誤的日誌串流。資訊豐富的日誌可用來檢視裝置或閘道狀態的變更。

  > **Note**
  >
  > 收集更詳細的日誌串流會產生額外的成本。如需定價的詳細資訊，請參閱 AWS IoT Core 定價。

- 錯誤

  日誌層級為 Error (錯誤) 的日誌不夠詳細且只會顯示錯誤資訊。當應用程式發生錯誤 (例如裝置連線錯誤) 時，您就可以使用這些日誌。藉由使用來自日誌串流的資訊，您可以識別並排解機群中資源的錯誤。

必要條件

在新增資源之前，您必須先將想監控的閘道和裝置加入 AWS IoT Core for LoRaWAN。如需詳細資訊，請參閱將閘道和裝置連接至 AWS IoT Core for LoRaWAN。

新增資源並使用主控台更新網路分析器組態

您可以新增資源，並使用 AWS IoT 主控台或 AWS IoT Wireless API 自訂選用參數。除了資源之外，您還可以編輯組態設定並儲存更新的組態。

新增資源到組態 (主控台)

1. 開啟 AWS IoT 主控台的網路分析器中樞，然後選擇網路分析器組態 NetworkAnalyzerConfig_Default。
2. 選擇 Add resource (新增資源)。
3. 使用無線閘道和無線裝置識別碼新增要監控的資源。您最多可以新增 250 個無線閘道或無線裝置。若要新增資源：
   a. 使用 View gateways (檢視閘道) 或 View device (檢視裝置) 索引標籤，查看已新增至 AWS 帳戶的閘道和裝置清單。
   b. 複製想要監控的裝置或閘道的 `WirelessDeviceID` 或 `WirelessGatewayID`，然後輸入對應資源的識別碼值。

c. 若要繼續新增資源，請選擇 Add gateway (新增閘道) 或 Add device (新增裝置)，然後新增無線閘道或裝置。如果不再想要監控新增的資源，請選擇 Remove resource (移除資源)。

4. 新增所有資源之後，請選擇 Add (新增)。

   您會在網路分析器中樞頁面中看到新增的閘道和裝置數目。您仍然可以繼續新增閘道和裝置，直到啟用追蹤訊息工作階段為止。啟動工作階段之後，若要新增資源，則必須停用工作階段。

編輯網路分析器組態 (主控台)

您也可以編輯網路分析器組態，並選擇是否要停用追蹤訊息日誌的框架資訊和日誌層級。

1. 開啟 AWS IoT 主控台的網路分析器中樞，然後選擇網路分析器組態 NetworkAnalyzerConfig_Default。
2. 選擇 Edit (編輯)。
3. 選擇是否停用框架資訊並使用 Selelct log levels (選取日誌層級)，以此選擇要用於追蹤訊息的日誌層級。選擇 Save (儲存)。

   您會在網路分析器組態的詳細資訊頁面中看到自己指定的組態設定。

新增資源並使用 API 更新網路分析器組態

您可以使用 AWS IoT Wireless API 操作或 AWS IoT Wireless CLI 命令來新增資源並更新網路分析器組態的組態設定。

- 若要新增資源並更新網路分析器組態，請使用 UpdateNetworkAnalyzerConfiguration API 或 update-network-analyzer-configuration CLI。
  - 新增資源

    對於要新增的無線裝置，請使用 `WirelessDevicesToAdd` 輸入裝置的 `WirelessDeviceID` 作為字串組。對於要新增的無線閘道，請使用 `WirelessGatewaysToAdd` 輸入閘道的 `WirelessGatewayID` 作為字串組。
  - 編輯組態

    若要編輯網路分析器組態，請使用 `TraceContent` 參數來指定 `WirelessDeviceFrameInfo` 是否應為 `ENABLED` 或 `DISABLED`，以及 `LogLevel` 參數是否應為 `INFO`、`ERROR` 或 `DISABLED`。

```
{
```

```
   "TraceContent": {
      "LogLevel": "string",
      "WirelessDeviceFrameInfo": "string"
   },
   "WirelessDevicesToAdd": [ "string" ],
   "WirelessDevicesToRemove": [ "string" ],
   "WirelessGatewaysToAdd": [ "string" ],
   "WirelessGatewaysToRemove": [ "string" ]
}
```

- 若要取得有關組態和已新增資源的資訊，請使用 GetNetworkAnalyzerConfiguration API 操作或 get-network-analyzer-configuration 命令。提供網路分析器組態的名稱 `NetworkAnalyzerConfig_Default` 作為輸入。

後續步驟

您已新增資源並為組態指定任何選用組態設定，接下來可以使用 WebSocket 通訊協定建立與 AWS IoT Core for LoRaWAN 的連線以使用網路分析器。然後，您可以啟用追蹤訊息並開始接收資源的追蹤訊息。如需詳細資訊，請參閱串流網路分析器使用 WebSocket 追蹤消息。

串流網路分析器使用 WebSocket 追蹤消息

在使用 WebSocket 通訊協定時，您可以即時串流網路分析器追蹤訊息。傳送請求時，服務會用 JSON 結構回應。啟用追蹤訊息之後，您可以使用訊息日誌取得與資源相關的資訊及排解錯誤。如需詳細資訊，請參閱 WebSocket 通訊協定。

以下主題展示串流網路分析器如何使用 WebSocket 追蹤訊息。

主題

- 使用 WebSocket 庫生成預先簽署的請求
- WebSocket 訊息和狀態碼

使用 WebSocket 庫生成預先簽署的請求

下文顯示如何生成預先簽署的請求，以便您可以使用 WebSocket 庫將請求傳送到服務。

將 WebSocket 請求的政策新增到 IAM 角色

若要使用 WebSocket 通訊協定呼叫網路分析器，您需要將以下政策附加至提出此請求的 AWS Identity and Access Management (IAM) 角色。

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

建立預先簽署的 URL

為您的 WebSocket 請求建構 URL，其中包含在您的應用程式和網路分析器之間建立通訊時所需的資訊。若要驗證請求的身分，WebSocket 串流使用 Amazon Signature 第 4 版程序來簽署請求。如需有關 Signature 第 4 版的詳細資訊，請參閱《Amazon Web Services 一般參考》中的簽署 AWS API 請求。

若要呼叫網路分析器，請使用 StartNetworkAnalyzerStream 請求 URL。將使用先前提及的 IAM 角色的憑證簽署請求。URL 採用下列格式，並會新增分行符以增加可讀性。

```
GET wss://api.iotwireless.<region>.amazonaws.com/start-network-analyzer-stream?X-Amz-
Algorithm=AWS4-HMAC-SHA256
   &X-Amz-Credential=Signature Version 4 credential scope
   &X-Amz-Date=date
   &X-Amz-Expires=time in seconds until expiration
   &X-Amz-Security-Token=security-token
   &X-Amz-Signature=Signature Version 4 signature
   &X-Amz-SignedHeaders=host
```

使用以下值作為 Singature 第 4 版的參數：

- X-Amz-Algorithm：您在簽署程序中使用的演算法。唯一有效的值為 AWS4-HMAC-SHA256。
- X-Amz-Credential：透過串連存取金鑰 ID 和憑證範圍元件所形成的以斜線 (「/」) 分隔的字串。憑證範圍包括 YYYYMMDD 格式的日期、AWS 區域、服務名稱和特殊終止字串 (aws4_request)。
- X-Amz-Date：簽章的建立日期與時間。依照《Amazon Web Services 一般參考》中 Signature 第 4 版中的處理日期指示生成日期與時間。
- X-Amz-Expires：憑證在逾期前尚有多少時間 (以秒為單位)。最大值為 300 秒 (5 分鐘)。

- X-Amz-Security-Token：(選用) 用於臨時安全憑證的 Signature 第 4 版字符。如果您指定此參數，請將其包含在正式請求中。如需詳細資訊，請參閱《AWS Identity and Access Management 使用者指南》中的請求臨時安全憑證。
- X-Amz-Signature：您為請求所生成的 Signature 第 4 版簽章。
- X-Amz-SignedHeaders：在為請求建立簽章時，所簽署的標頭。唯一有效的值為 host。

建構請求 URL 以及建立 Signature 第 4 版簽章

若要為請求建構 URL 及建立 Signature 第 4 版簽章，請使用下列步驟。這些範例是虛擬程式碼。

任務 1：建立正式請求

建立字串，其中包括來自您的請求的資訊 (使用標準化格式)。這可確保當 AWS 收到請求時，可以計算出您在 任務 3：計算簽章 中計算出的相同簽章。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的為 Signature 第 4 版建立正式請求。

1. 在應用程式中定義請求的變數。

```
# HTTP verb
method = "GET"
# Service name
service = "iotwireless"
# AWS ##
region = "AWS ##"
# Service streaming endpoint
endpoint = "wss://api.iotwireless.region.amazonaws.com"
# Host
host = "api.iotwireless.<region>.amazonaws.com"
# Date and time of request
amz-date = YYYYMMDD'T'HHMMSS'Z'
# Date without time for credential scope
datestamp = YYYYMMDD
```

2. 建立正式的 URI (統一資源識別符)。正式 URI 是介於網域與查詢字串之間的部分 URI。

```
canonical_uri = "/start-network-analyzer-stream"
```

3. 建立正式標頭和已簽章標頭。請注意正式標頭中的結尾 \n。

   - 附加小寫標頭名稱，後面接著冒號。

- 為該標頭附加逗號分隔的值清單。不要在有多個值的標頭中排序值。
- 附加新的一行 (\n)。

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. 演算法必須符合雜湊演算法。您必須使用 SHA-256。

```
algorithm = "AWS4-HMAC-SHA256"
```

5. 建立憑證的範圍，將衍生的金鑰範圍限定於提出請求的日期、區域和服務。

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. 建立正式查詢字串。查詢字串值必須是 URL 編碼並以名稱排序。

- 依照字元字碼指標的參數名稱遞增排序。具有重複名稱的參數應依數值排序。例如，以大寫字母 F 開頭的參數名稱，放在以小寫字母 b 開頭的參數名稱之前。
- URI 編碼不能執行 RFC 3986 所定義的任何未預留字元：A–Z、a–z、0–9、連字號 (-)、底線 (_)、句點 (.) 和波狀符號 (~)。
- 對所有其他含有 %XY 的字元執行百分比編碼，其中 X 和 Y 是十六進位字元 (0-9 和大寫 A-F)。例如，空間字元必須編碼為 %20 (而非像有些編碼結構描述那樣使用「+」)，而延伸的 UTF-8 字元必須採用 %XY%ZA%BC 格式。
- 對參數值中的任何等於 (=) 字元進行雙倍編碼。

```
canonical_querystring  = "X-Amz-Algorithm=" + algorithm
canonical_querystring += "&X-Amz-Credential="+ URI-encode(access key + "/" +
 credential_scope)
canonical_querystring += "&X-Amz-Date=" + amz_date
canonical_querystring += "&X-Amz-Expires=300"
canonical_querystring += "&X-Amz-Security-Token=" + token
canonical_querystring += "&X-Amz-SignedHeaders=" + signed_headers
canonical_querystring += "&language-code=en-US&media-encoding=pcm&sample-
rate=16000"
```

7. 建立承載的雜湊。對於 GET 請求，承載為空字串。

```
payload_hash = HashSHA256(("").Encode("utf-8")).HexDigest()
```

8. 結合所有元素來建立正式請求。

```
canonical_request = method + '\n'
   + canonical_uri + '\n'
   + canonical_querystring + '\n'
   + canonical_headers + '\n'
   + signed_headers + '\n'
   + payload_hash
```

任務 2：建立要簽署的字串

要簽署的字串包含有關您的請求的中繼資料。您在下一個步驟計算請求簽章時，使用此字串登入。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的建立 Signature 第 4 版的登入字串。

```
string_to_sign=algorithm + "\n"
   + amz_date + "\n"
   + credential_scope + "\n"
   + HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

任務 3：計算簽章

從您的 AWS 私密存取金鑰衍生簽署金鑰。為了提供更大程度的保護，衍生的金鑰有專屬的日期、服務和 AWS 區域。您使用衍生的金鑰來簽署請求。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的為 AWS Signature 第 4 版計算簽章。

此程式碼假設您已實作 `GetSignatureKey` 函數來衍生簽署金鑰。如需詳細資訊和範例函數，請參閱《Amazon Web Services 一般參考》中的如何衍生 Signature 第 4 版的簽署密鑰範例。

函數 `HMAC(key, data)` 代表 HMAC-SHA256 函數，它會以二進位格式傳回結果。

```
#Create the signing key
signing_key = GetSignatureKey(secret_key, datestamp, region, service)

# Sign the string_to_sign using the signing key
signature = HMAC.new(signing_key, (string_to_sign).Encode("utf-8"), Sha256()).HexDigest
```

任務 4：將簽署資訊添加至請求並建立請求 URL

您計算簽章之後，請將簽章新增到查詢字串。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的新增簽章至請求。

```
#Add the authentication information to the query string
canonical_querystring += "&X-Amz-Signature=" + signature

# Sign the string_to_sign using the signing key
request_url = endpoint + canonical_uri + "?" + canonical_querystring
```

後續步驟

您現在可以使用請求 URL 與您的 WebSocket 程式庫，以便向服務提出請求並觀察訊息。如需詳細資訊，請參閱WebSocket 訊息和狀態碼。

WebSocket 訊息和狀態碼

建立預先簽署的請求之後，您可以使用請求 URL 與您的 WebSocket 程式庫或適合您程式設計語言的程式庫，向服務提出請求。如需如何產生此預先簽署請求的詳細資訊，請參閱 使用 WebSocket 庫生成預先簽署的請求。

WebSocket 訊息

可以使用 WebSocket 通訊協定建立雙向連線。訊息可以從用戶端傳輸到服務器，也可以從服務器傳輸到用戶端。不過，網路分析器只支援從伺服器傳送到用戶端的訊息。意外從用戶端接收的任何訊息，若訊息來自用戶端，則伺服器將自動關閉 WebSocket 連線。

當收到請求並啟用追蹤訊息工作階段時，伺服器以 JSON 結構回應，即為承載。如需有關承載的詳細資訊，以及如何從 AWS Management Console 啟用追蹤訊息，請參閱 即時檢視並監控網路分析器追蹤訊息日誌。

WebSocket 狀態碼

下文顯示了伺服器到用戶端的通訊的 WebSocket 狀態碼。WebSocket 狀態碼遵循連線正常關閉的 RFC 標準。

下文顯示受支援的狀態碼：

- 1000

  這個狀態碼表示正常關閉，代表已建立 WebSocket 連線且已完成請求。當工作階段閒置時即可觀察到此狀態，這會造成連線逾時。

- 1002

  此狀態碼表示端點因通訊協定錯誤而終止連線。

- 1003

  此狀態碼表示錯誤狀態，其中端點因接收的資料格式無法接受而終止連線。端點僅支援文字資料，如果它收到二進位訊息或來自使用不支援格式的用戶端的訊息，則可能會顯示此狀態碼。

- 1008

  此狀態碼表示錯誤狀態，其中端點因收到違反其政策的訊息而終止連線。此狀態為一般狀態，會在其他狀態碼 (例如 1003 或 1009) 不適用時顯示。如果需要隱藏政策或授權失敗 (例如簽章過期)，也會顯示此狀態。

- 1011

  此狀態碼表示錯誤狀態，其中伺服器因遇到未預料的情況或內部錯誤，導致無法滿足請求而終止連線。

後續步驟

現在您已經學會了如何生成預先簽署的請求，以及如何使用 WebSocket 連線來觀察伺服器的訊息；接下來，您可以啟用追蹤訊息，並開始接收無線閘道和無線裝置資源的訊息日誌。如需詳細資訊，請參閱即時檢視並監控網路分析器追蹤訊息日誌。

即時檢視並監控網路分析器追蹤訊息日誌

如果已將資源新增至網路分析器組態，則可以啟動追蹤訊息，開始接收資源的追蹤訊息。您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI。

必要條件

使用網路分析器啟用追蹤訊息之前，您必須具備：

- 已新增要監控的資源到預設的網路分析器組態。如需詳細資訊，請參閱新增資源並更新網路分析器組態。
- 使用 `StartNetworkAnalyzerStream` 請求 URL 生產預先簽署的請求。將使用提出此請求的 AWS Identity and Access Management 角色憑證來簽署請求。如需詳細資訊，請參閱建立預先簽署的 URL。

使用主控台啟用追蹤訊息

若要啟用追蹤訊息

1. 開啟 AWS IoT 主控台的網路分析器中樞，然後選擇網路分析器組態 NetworkAnalyzerConfig_Default。
2. 在網路分析器組態的詳細資料頁面中，選擇 Activate trace messaging (啟用追蹤簡訊)，然後選擇 Activate (啟用)。

   您會開始接收追蹤訊息，其中最新的追蹤訊息會先出現在主控台中。

> **Note**
>
> 訊息工作階段啟動後，接收追蹤訊息可能會產生額外的成本，直到您停用工作階段或離開追蹤工作階段為止。如需定價的詳細資訊，請參閱 AWS IoT Core 定價。

檢視和監控追蹤訊息

啟用追蹤訊息之後，就會建立 WebSocket 連線，並會即時出現追蹤訊息 (最先出現最新訊息)。您可以自訂偏好設定，指定要在每個頁面中顯示的追蹤訊息數目，並僅顯示每個訊息的相關欄位。例如，您可以自訂追蹤訊息日誌，僅顯示將 Log level (日誌層級) 設定成 ERROR 的無限閘道資源，以便快速識別和偵錯閘道的錯誤。追蹤訊息包含下列資訊。

- Message Number (訊息編號)：唯一的編號，可顯示最先收到的最新訊息。
- Resource ID (資源 ID)：資源的無線閘道或無線裝置 ID。
- Timestamp (時間戳記)：收到訊息的時間。
- 訊息 ID：AWS IoT Core for LoRaWAN 指派給每則收到的訊息的識別符。
- FPort：使用 WebSocket 連線與裝置通訊的頻率連接埠。
- DevEui：無線裝置的延伸唯一識別碼 (EUI)。
- Resource (資源)：不論受監控的資源是無線裝置或無線閘道都一樣。
- Event (事件)：無線裝置日誌訊息的事件，可以是 Join (加入)、Rejoin (重新加入)、Uplink_Data、Downlink_Data 或 Registration (註冊)。
- Log level (日誌層級)：裝置 INFO 或 ERROR 日誌串流的相關資訊。

網絡分析器 JSON 日誌訊息

您也可以一次選擇一個追蹤訊息，以檢視該訊息的 JSON 承載。根據您在追蹤訊息日誌中選取的訊息，您會在 JSON 承載中看到表示包含 2 個部分的資訊：CustomerLog 和 LoRaFrame。

CustomerLog

JSON 的 CustomerLog 部分會顯示接收訊息的資源類型和識別碼、日誌層級以及訊息內容。下列範例展示 CustomerLog 日誌訊息。您可以使用 JSON 中的 `message` 欄位取得有關錯誤以及如何解決錯誤的詳細資訊。

LoRaFrame

JSON 的 LoRaFrame 部分具有 Message ID (訊息 ID)，並包含裝置實體承載和無線中繼資料的相關資訊。

下列範例顯示追蹤訊息的結構。

```
export type TraceMessage = {
  ResourceId: string;
  Timestamp: string;
  LoRaFrame:
  {
    MessageId: string;
    PhysicalPayload: any;
    WirelessMetadata:
    {
      fPort: number;
      dataRate: number;
      devEui: string;
      frequency: number,
      timestamp: string;
    },
  }
  CustomerLog:
  {
    resource: string;
    wirelessDeviceId: string;
    wirelessDeviceType: string;
    event: string;
    logLevel: string;
    messageId: string;
    message: string;
```

```
  },
};
```

檢閱和後續步驟

在本節中，您已檢視追蹤訊息，並了解如何使用這些資訊來偵錯。檢視所有訊息後，您可以：

- 停用追蹤訊息

  若要避免產生任何額外成本，您可以停用追蹤訊息工作階段。停用工作階段會中斷 WebSocket 連線，可免於收到其他追蹤訊息。您仍然可以繼續檢視主控台中的現有訊息。
- 編輯組態的框架資訊

  您可以編輯網路分析器組態、選擇是否要停用框架資訊以及選擇訊息的日誌層級。更新組態之前，請考慮停用追蹤訊息工作階段。若要進行這些編輯，請開啟 AWS IoT 主控台中的網路分析器詳細資訊頁面並選擇 Edit (編輯)。然後，您可以使用新的組態設定來更新組態，並啟動追蹤訊息來查看更新訊息。
- 新增資源到組態

  您也可以在網路分析器組態中新增更多資源，並即時監控這些資源。您最多可以新增總共 250 個無線閘道和無線裝置資源。若要新增資源，請在 AWS IoT 主控台中的網路分析器詳細資訊頁面選擇 Resources (資源) 索引標籤，然後選擇 Add resources (新增資源)。然後，您可以使用新資源更新組態，並啟動追蹤訊息來查看其他資源的更新訊息。

如需有關透過編輯組態設定和新增資源來更新網路分析器組態的詳細資訊，請參閱 新增資源並更新網路分析器組態。

使用網路分析器，針對您的多播群組和 FUOTA 任務進行偵錯和疑難排解

您可以監控的無線資源包括 LoRaWAN 裝置、LoRaWAN 閘道和多播群組。您也可以使用網路分析器，針對 FUOTA 任務的任何問題進行偵錯和疑難排解。您也可以在 FUOTA 任務進行時監控並追蹤與設定、資料傳輸和狀態查詢相關的訊息。

若要監控 FUOTA 任務，如果任務包含多播群組，您必須同時將多播群組和該群組中的裝置新增至網路分析器組態。您也必須啟動框架資訊和多播框架資訊，以追蹤在 FUOTA 任務進行時與多播群組和裝置交換的單播和多播上行和下行訊息。

若要監控多播群組，您可以將它們新增至網路分析器組態，並使用多播框架資訊，針對傳送至這些群組的多播下行訊息進行疑難排解。如需針對嘗試加入群組 (其中使用單播通訊) 的裝置進行疑難排解，您

也必須在網路分析器組態中包括這些裝置。若要僅監控與群組中裝置的單播通訊，請啟動無線裝置的框架資訊。此方法可確保針對多播群組和加入該群組的裝置進行全方位的監控和診斷。

下列各節描述如何使用網路分析器，針對您的多播群組和 FUOTA 任務進行偵錯和疑難排解。

主題

針對僅包含裝置的 FUOTA 任務進行偵錯

您可以使用網路分析器，針對只有 LoRaWAN 裝置新增至任務的 FUOTA 任務進行偵錯。如需將裝置新增至 FUOTA 任務的相關資訊，請參閱 將裝置和多點傳送群組新增至 FUOTA 任務，並排定 FUOTA 工作階段。。若要針對 FUOTA 任務進行偵錯，請執行下列步驟：

1. 啟動無線裝置的框架資訊來建立網路分析器組態，以便您可以在任務進行時監控與裝置交換的 FUOTA 上行和下行訊息。
2. 將 FUOTA 任務中的裝置新增至網路分析器組態，方法為使用其無線裝置識別碼。
3. 啟動追蹤訊息，開始接收網路分析器組態中裝置的追蹤訊息。

在追蹤訊息資訊的 `applicationCommandType` 欄中，您將開始接收與資料傳輸和分段設定相關的單播下行訊息。

> **Note**
>
> 如果在追蹤訊息表格中看不到 `applicationCommandType` 欄，您可以調整設定以在表格中顯示此欄。

您也可以在 WirelessMetadata > ApplicationInfo 下的 JSON 日誌訊息中看到 `applicationCommandType` 和其他詳細訊息。

針對具有多播群組的 FUOTA 任務進行偵錯

您可以使用網路分析器，針對具有多播群組且 LoRaWAN 裝置新增至群組的 FUOTA 任務進行偵錯。如需將裝置新增至 FUOTA 任務的相關資訊，請參閱 將裝置和多點傳送群組新增至 FUOTA 任務，並排定 FUOTA 工作階段。。若要針對 FUOTA 任務進行偵錯，請執行下列步驟：

1. 啟動無線裝置和多播群組的框架資訊和多播框架資訊設定，以建立網路分析器組態。
2. 將 FUOTA 任務中的多播群組新增至網路分析器組態，方法為使用其多播群組識別碼。透過啟用多播框架資訊，您可以針對在 FUOTA 任務進行時傳送至群組的韌體資料訊息和 FUOTA 狀態查詢訊息進行偵錯。
3. 將多播群組中的裝置新增至網路分析器組態，方法為使用其無線裝置識別碼。透過啟動框架資訊，您可以監控在 FUOTA 任務進行期間直接與裝置交換的上行和下行訊息。
4. 啟動追蹤訊息，開始接收網路分析器組態中裝置和多播群組的追蹤訊息。

然後，您可以檢視追蹤訊息，並對其進行偵錯，方法為使用追蹤訊息表格的 `applicationCommandType` 欄，並使用 JSON 日誌訊息中的詳細資訊，如 針對僅包含裝置的 FUOTA 任務進行偵錯 中所述。

針對嘗試加入多播群組的裝置進行偵錯

您可以使用網路分析器，針對嘗試加入多播群組的裝置進行偵錯 如需將裝置新增至多播群組的相關資訊，請參閱 建立多點傳送群組以及將裝置新增至群組。若要針對多播群組進行偵錯，請執行下列步驟：

1. 啟動無線裝置的框架資訊來建立網路分析器組態。
2. 將您要監控的裝置新增至網路分析器組態，方法為使用其無線裝置識別碼。
3. 啟動追蹤訊息，開始接收網路分析器組態中裝置的追蹤訊息。
4. 在針對群組中的裝置啟動追蹤訊息之後，開始將裝置與多播群組建立關聯。

針對多播群組工作階段進行偵錯

您可以使用網路分析器，針對多播群組工作階段進行偵錯。如需詳細資訊，請參閱排定將下行訊息傳送至多點傳送群組中的裝置。若要針對多播群組工作階段進行偵錯，請執行下列步驟：

1. 啟動多播群組的多播框架資訊來建立網路分析器組態。
2. 將您要監控的多播群組新增至網路分析器組態，方法為使用其多播群組識別碼。

3. 多播工作階段開始之前，請啟動追蹤訊息，開始接收多播群組工作階段的追蹤訊息。
4. 透過檢視追蹤訊息表格中顯示的訊息和 JSON 日誌訊息，啟動多播群組工作階段並監控狀態。

在追蹤訊息表格中，`MulticastAddr` 將會顯示在 `DevAddr` 欄中。在 JSON 日誌訊息中，您可以檢視詳細訊息，例如在 WirelessMetadata > ApplicationInfo 下的 `MulticastGroupId`。

AWS IoT Core for LoRaWAN 和介面 VPC 端點 (AWS PrivateLink)

您可以透過 Virtual Private Cloud (VPC) 中的介面 VPC 端點 (AWS PrivateLink) 直接連接至 AWS IoT Core for LoRaWAN，而非透過公有網際網路進行連接。使用 VPC 介面端點時，VPC 和 AWS IoT Core for LoRaWAN 之間的通訊會完全在 AWS 網路內安全地進行。

AWS IoT Core for LoRaWAN 支援採用 AWS PrivateLink 技術的 Amazon Virtual Private Cloud 介面端點。每個 VPC 端點皆會由一個或多個在您的 VPC 子網路上具私有 IP 地址的彈性網路界面來表示。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的介面 VPC 端點 (AWS PrivateLink)。

如需 VPC 和端點的詳細資訊，請參閱什麼是 Amazon VPC。

如需 AWS PrivateLink 的詳細資訊，請參閱 AWS PrivateLink 和 VPC 端點。

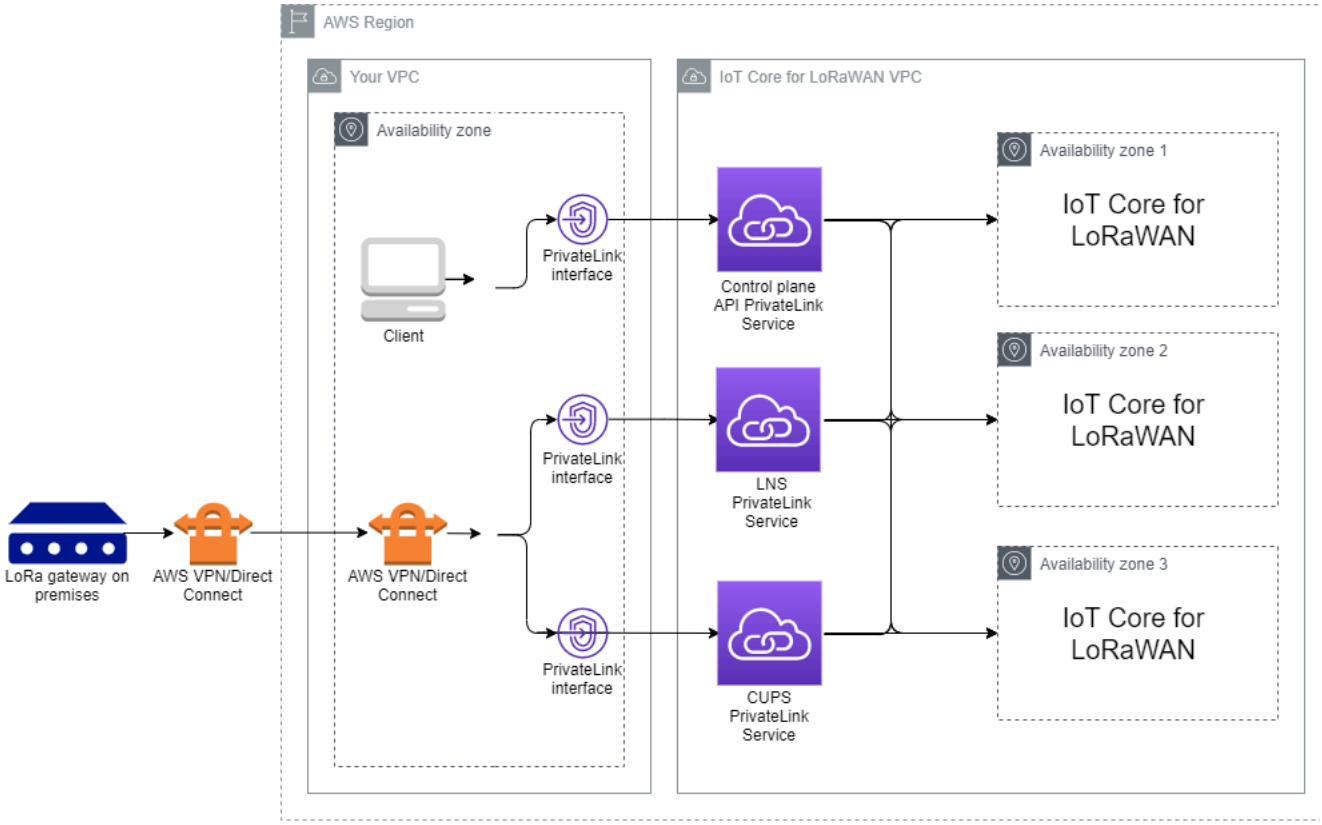
AWS IoT Wireless VPC 端點的考量事項

設定 AWS IoT Wireless 的介面 VPC 端點前，請務必檢閱《Amazon VPC 使用者指南》中的介面端點屬性和限制。

AWS IoT Wireless 支援從您的 VPC 呼叫其所有 API 動作。AWS IoT Wireless 不支援 VPC 端點政策。根據預設，您可透過端點完整存取 AWS IoT Wireless。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。

AWS IoT Core for LoRaWAN PrivateLink 架構

下方圖表顯示 AWS IoT Core for LoRaWAN 的 Privatelink 架構。此架構使用 Transit Gateway 和 Route 53 解析程式，在 VPC、AWS IoT Core for LoRaWAN 和內部部署環境之間共用 AWS PrivateLink 介面端點。設定與 VPC 介面端點的連線時，您會發現更詳細的架構圖。



AWS IoT Core for LoRaWAN 端點

AWS IoT Core for LoRaWAN 有三個公有端點。每個公有端點都有對應的 VPC 介面端點。公有端點可以分類為控制平面和資料平面端點。如需這些端點的相關資訊，請參閱 AWS IoT Core for LoRaWAN API 端點。

- 控制平面 API 端點

  您可以使用控制平面 API 端點與 AWS IoT Wireless API 互動。您可以使用 AWS PrivateLink，從 Amazon VPC 中託管的用戶端存取這些端點。

- 資料平面 API 端點

  資料平面 API 端點是 LoRaWAN 網路伺服器 (LNS) 以及組態與更新伺服器 (CUPS) 端點，您可以使用這些端點與 AWS IoT Core for LoRaWAN LNS 和 CUPS 端點互動。這些端點可以從 LoRa 閘道內部部署中存取，方法為使用 AWS VPN 或 AWS Direct Connect。將您的閘道加入至 AWS IoT Core for LoRaWAN 時，您可以取得這些端點。如需詳細資訊，請參閱將閘道新增至 AWS IoT Core for LoRaWAN。

主題

- 加入 AWS IoT Core for LoRaWAN 控制平面 API 端點
- 加入 AWS IoT Core for LoRaWAN 資料平面 API 端點

加入 AWS IoT Core for LoRaWAN 控制平面 API 端點

您可以使用 AWS IoT Core for LoRaWAN 控制平面 API 端點與 AWS IoT Wireless API 互動。例如，您可使用此端點來執行 SendDataToWirelessDevice API，將資料從 AWS IoT 傳送至 LoRaWAN 裝置。如需詳細資訊，請參閱 AWS IoT Core for LoRaWAN 控制平面 API 端點。

您可以使用 Amazon VPC 中託管的用戶端，存取 AWS PrivateLink 提供的控制平面端點。您可以使用這些端點，透過 Virtual Private Cloud (VPC) 中的介面端點連接至 AWS IoT Wireless API，而不是透過公有網際網路進行連接。

若要加入控制平面端點：

- 建立您的 Amazon VPC 和子網路
- 在子網路中啟動 Amazon EC2 執行個體
- 建立 Amazon VPC 介面端點
- 測試您與介面端點的連線

建立您的 Amazon VPC 和子網路

在可以連接到介面端點之前，您必須建立 VPC 和子網路。然後，將在您的子網路中啟動 EC2 執行個體，您可以使用該執行個體連接到介面端點。

若要建立您的 VPC：

1. 導覽至 Amazon VPC 主控台的 VPC (VPC) 頁面，然後選擇 Create VPC (建立 VPC)。
2. 在 Create VPC (建立 VPC) 頁面上：
   - 輸入 VPC Name tag - optional (VPC 名稱標籤 - 選用) 的名稱 (例如，**`VPC-A`**)。
   - 在 IPv4 CIDR 區塊中輸入 VPC 的 IPv4 位址範圍 (例如，**`10.100.0.0/16`**)。
3. 保留其他欄位的預設值，然後選擇 Create VPC (建立 VPC)。

若要建立您的子網路：

1. 導覽至 Amazon VPC 主控台的 Subnets (子網路) 頁面，然後選擇 Create subnet (建立子網路)。

2. 在 Create subnet (建立子網路) 頁面上：
   - 針對 VPC ID，選擇您稍早建立的 VPC (例如，VPC-A)。
   - 輸入 Subnet name (子網路名稱) 的名稱(例如，**Private subnet**)。
   - 為您的子網路選擇 Availability Zone (可用區域)。
   - 在 IPv4 CIDR block (IPv4 CIDR 區塊) 中輸入子網路的 IP 地址區塊 (例如，**10.100.0.0/24**)。
3. 若要建立子網路並將其新增至 VPC，請選擇 Create subnet (建立子網路)。

如需詳細資訊，請參閱使用 VPC 和子網路。

在子網路中啟動 Amazon EC2 執行個體

若要啟動您的 EC2 執行個體：

1. 導覽至 Amazon EC2 主控台，然後選擇 Launch Instance (啟動執行個體)。
2. 針對 AMI，選擇 Amazon Linux 2 AMI (HVM), SSD Volume Type (Amazon Linux 2 AMI (HVM)，SSD 磁碟區類型)，然後選擇 t2 micro 執行個體類型。若要設定執行個體詳細資訊，請選擇 Next (下一步)。
3. 在 Configure Instance Details (設定執行個體詳細資訊) 頁面中：
   - 針對 Network (網路)，選擇您稍早建立的 VPC (例如，VPC-A)。
   - 針對 Subnet (子網路)，選擇您稍早建立的子網路 (例如，**Private subnet**)。
   - 針對 IAM role (IAM 角色)，選擇角色 AWSIoTWirelessFullAccess 將完整存取政策授予 AWS IoT Core for LoRaWAN。如需詳細資訊，請參閱 AWSIoTWirelessFullAccess 政策摘要。
   - 針對 Assume Private IP (採用私有 IP)，使用 IP 地址，例如，10.100.0.42。
4. 選擇 Next: Add Storage (下一步：新增儲存體)，然後選擇 Next：Add tags (下一步：新增標籤)。您可以選擇性地新增任何標籤，與您的 EC2 執行個體建立關聯。選擇 Next: Configure Security Group (下一步：設定安全群組)。
5. 在 Configure Security Group (設定安全群組) 頁面中，設定要允許的安全群組：
   - 將來源的 All TCP (所有 TCP) 開啟為 10.200.0.0/16。
   - 將來源的 All ICMP - IPV4 (所有 ICMP - IPV4) 開啟為 10.200.0.0/16。
6. 若要檢閱執行個體詳細資訊並啟動 EC2 執行個體，請選擇 Review and Launch (檢閱並發佈)。

如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體入門。

建立 Amazon VPC 介面端點

您可以為 VPC 建立 VPC 端點，然後可以透過 EC2 API 來存取該端點。若要建立端點：

1. 導覽至 VPC Endpoints (端點) 主控台，然後選擇 Create Endpoint (建立端點)。
2. 在 Create Endpoint (建立端點) 頁面中，指定下列資訊。
   - 選擇服務類別的 AWS 服務。
   - 針對 Service Name (服務名稱)，輸入關鍵字 **iotwireless** 進行搜尋。在顯示的 iotwireless 服務清單中，請為您的區域選擇控制平面 API 端點。端點的格式為 com.amazonaws.*region*.iotwireless.api。
   - 針對 VPC 和 Subnets (子網路)，選擇要在其中建立端點的 VPC，以及要在其中建立端點網路的可用區域 (AZ)。

     > **Note**
     >
     > iotwireless 服務可能無法支援所有可用區域。

   - 針對 Enable DNS name (啟用 DNS 名稱)，選擇 Enable for this endpoint (為此端點啟用)。

     選擇此選項會自動解析 DNS，並在 Amazon Route 53 Public Data Plane 中建立路由，以便您稍後用來測試連線的 API 將通過 Privatelink 端點。
   - 針對 Security group (安全群組)，選擇要與端點網路介面建立關聯的安全群組。
   - 您可以選擇性地新增或移除標籤。標籤是您用來與端點建立關聯的名稱值對。
3. 若要建立 VPC 端點，請選擇 Create endpoint (建立端點)。

測試您與介面端點的連線

您可以使用 SSH 存取 Amazon EC2 執行個體，然後使用 AWS CLI 以連接至 Privatelink 介面端點。

在您連接至介面端點之前，請遵循在 Linux 上安裝、更新和解除安裝 AWS CLI 第 2 版中描述的指示來下載最新的 AWS CLI 版本。

下列範例顯示如何使用 CLI 來測試您與介面端點的連線。

```
aws iotwireless create-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com  \
    --name='test-privatelink'
```

下列顯示執行命令的範例。

```
Response:
{
 "Arn": "arn:aws:iotwireless:region:acct_number:ServiceProfile/1a2345ba-4c5d-67b0-ab67-e0c8342f2857",
 "Id": "1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
}
```

同樣地，您可以執行下列命令，來取得服務設定檔資訊或列出所有服務設定檔。

```
aws iotwireless get-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
    --id="1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
```

下列顯示 list-device-profiles 命令的範例。

```
aws iotwireless list-device-profiles \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
```

加入 AWS IoT Core for LoRaWAN 資料平面 API 端點

AWS IoT Core for LoRaWAN 資料平面端點包含下列端點。將您的閘道新增至 AWS IoT Core for LoRaWAN 時，您可以取得這些端點。如需詳細資訊，請參閱將閘道新增至 AWS IoT Core for LoRaWAN。

- LoRaWAN 網路伺服器 (LNS) 端點

  LNS 端點的格式為 *account-specific-prefix*.lns.lorawan.*region*.amazonaws.com。您可以使用此端點，來建立用於交換 LoRa 上行和下行訊息的連線。

- 組態與更新伺服器 (CUPS) 端點

  CUPS 端點的格式為 *account-specific-prefix*.cups.lorawan.*region*.amazonaws.com。您可以將此端點用於閘道的憑證管理、遠端組態和韌體更新。

如需詳細資訊，請參閱使用 CUPS 和 LNS 通訊協定。

若要尋找適用於 AWS 帳戶 和 Region 的資料平面 API 端點，請使用這裡顯示的 get-service-endpoint CLI 命令或 GetServiceEndpoint REST API。如需詳細資訊，請參閱 AWS IoT Core for LoRaWAN 資料平面 API 端點。

您可以連接內部部署 LoRaWAN 閘道，以便與 AWS IoT Core for LoRaWAN 端點通訊。若要建立此連線，首先使用 VPN 連線，將內部部署閘道連接至 VPC 中的 AWS 帳戶。然後，您可以與 AWS IoT Core for LoRaWAN VPC 中 Privatelink 提供的資料平面介面端點通訊。

下列顯示如何加入這些端點。

- 建立 VPC 介面端點和私有託管區域
- 使用 VPN 將 LoRa 閘道連接至您的 AWS 帳戶

建立 VPC 介面端點和私有託管區域

AWS IoT Core for LoRaWAN 有兩個資料平面端點：組態與更新伺服器 (CUPS) 端點和 LoRaWAN 網路伺服器 (LNS) 端點。建立與這兩個端點的 Privatelink 連線的設定程序相同，因此我們基於說明目的使用 LNS 端點。

對於您的資料平面端點，LoRa 閘道首先會連接到 Amazon VPC 中的 AWS 帳戶，然後連接到 AWS IoT Core for LoRaWAN VPC 中的 VPC 端點。

連接到端點時，DNS 名稱可以在一個 VPC 內進行解析，但無法跨多個 VPC 進行解析。若要在建立端點時停用私有 DNS，請停用 Enable DNS name (啟用 DNS 名稱) 設定。您可以使用私有託管區域，以提供您想要 Route 53 如何針對 VPC 回應 DNS 查詢的相關資訊。若要與內部部署環境共用 VPC，您可以使用 Route 53 解析程式來協助混合 DNS。

若要完成此程序，請執行以下步驟：

- 建立 Amazon VPC 和子網路
- 建立 Amazon VPC 介面端點
- 設定私有託管區域
- 設定 Route 53 傳入解析程式
- 後續步驟

建立 Amazon VPC 和子網路

您可以重複使用您在加入控制平面端點時建立的 Amazon VPC 和子網路。如需相關資訊，請參閱 建立您的 Amazon VPC 和子網路。

建立 Amazon VPC 介面端點

您可以為 VPC 建立 VPC 端點，這與您為控制平面端點建立一個端點的方式類似。

1. 導覽至 VPC Endpoints (端點) 主控台，然後選擇 Create Endpoint (建立端點)。
2. 在 Create Endpoint (建立端點) 頁面中，指定下列資訊。
   - 選擇服務類別的 AWS 服務。
   - 針對 Service Name (服務名稱)，輸入關鍵字 **lns** 進行搜尋。在顯示的 `lns` 服務清單中，請為您的區域選擇 LNS 資料平面 API 端點。端點的格式為 `com.amazonaws.region.lorawan.lns`。

     > **Note**
     >
     > 如果您是針對 CUPS 端點遵循此程序，請搜尋 `cups`。端點的格式為 `com.amazonaws.region.lorawan.cups`。

   - 針對 VPC 和 Subnets (子網路)，選擇要在其中建立端點的 VPC，以及要在其中建立端點網路的可用區域 (AZ)。

     > **Note**
     >
     > `iotwireless` 服務可能無法支援所有可用區域。

   - 針對 Enable DNS name (啟用 DNS 名稱)，確定未選取 Enable for this endpoint (為此端點啟用)。

     透過不選取此選項，您可以停用 VPC 端點的私有 DNS，並改用私有託管區域。
   - 針對 Security group (安全群組)，選擇要與端點網路介面建立關聯的安全群組。
   - 您可以選擇性地新增或移除標籤。標籤是您用來與端點建立關聯的名稱值對。
3. 若要建立 VPC 端點，請選擇 Create endpoint (建立端點)。

設定私有託管區域

在建立 Privatelink 端點之後，您會在端點的 Details (詳細資訊) 標籤中看到 DNS 名稱清單。您可以使用其中一個 DNS 名稱來設定私有託管區域。DNS 名稱的格式為 `vpce-xxxx.lns.lorawan.region.vpce.amazonaws.com`。

建立私有託管區域

若要建立私有託管區域：

1. 導覽至 Route 53 Hosted zones (託管區域) 主控台，然後選擇 Create hosted zone (建立託管區域)。
2. 在 Create hosted zone (建立託管區域) 頁面中，指定下列資訊。
   - 針對 Domain name (網域名稱)，請為您的 LNS 端點 **`lns.lorawan.region.amazonaws.com`** 輸入完整服務名稱。

     > **Note**
     >
     > 如果您是針對 CUPS 端點遵循此程序，請輸入 **`cups.lorawan.region.amazonaws.com`**。

   - 針對 Type (類型)，選擇 Private Hosted Zone (私有託管區域)。
   - 或者，您可以新增或移除要與託管區域建立關聯的標籤。
3. 若要建立私有託管區域，請選擇 Create hosted zone (建立託管區域)。

如需詳細資訊，請參閱建立私有託管區域。

在建立了私有託管區域之後，您可以建立記錄，告訴 DNS 您想要流量路由至該網域的方式。

建立記錄

在建立了私有託管區域之後，您可以建立記錄，告訴 DNS 您想要流量路由至該網域的方式。若要建立記錄：

1. 在顯示的託管區域清單，選擇您先前建立的私有託管區域，然後選擇 Create record (建立記錄)。
2. 使用精靈方法來建立記錄。如果主控台呈現 Quick create (快速建立) 方法，請選擇 Switch to wizard (切換至精靈)。
3. 為 Routing policy (路由政策) 選擇 Simple Routing (簡易路由)，然後選擇 Next (下一步)。
4. 在 Configure records (設定記錄) 頁面中，選擇 Define simple record (定義簡易記錄)。
5. 在 Define simple record (定義簡易記錄) 頁面中：
   - 針對 Record name (記錄名稱)，輸入 AWS 帳戶編號的別名。您可以在加入閘道時取得此值，或使用 `GetServiceEndpoint` REST API 取得此值。
   - 針對 Record type (紀錄類型)，將值保留為 `A - Routes traffic to an IPv4 address and some AWS resources`。

- 針對 Value/Route traffic to (值/路由流量至)，選擇 Alias to VPC endpoint (VPC 端點的別名)。接著，選擇您的 Region (區域)，然後從顯示的端點清單中選擇您先前建立的端點，如建立 Amazon VPC 介面端點所述。

6. 選擇 Define simple record (定義簡易記錄) 來建立您的記錄。

設定 Route 53 傳入解析程式

若要與內部部署環境共用 VPC 端點，Route 53 解析程式可以用來協助混合 DNS。傳入解析程式可讓您將流量從內部部署網路路由至資料平面端點，無需通過公有網際網路。若要傳回服務的私有 IP 地址值，請在與 VPC 端點相同的 VPC 中建立 Route 53 解析程式。

當您建立傳入解析程式時，只須指定 VPC 和先前在可用區域 (AZ) 中建立的子網路。Route 53 解析程式會使用此資訊來自動指派 IP 地址，以將流量路由至每個子網路。

若要建立傳入解析程式：

1. 導覽至 Route 53 Inbound endpoints (傳入端點) 主控台，然後選擇 Create inbound endpoint (建立傳入端點)。

> **Note**
>
> 確定您使用的是您在建立端點和私有託管區域時所使用的同一個 AWS 區域。

2. 在 Create inbound endpoint (建立傳入端點) 頁面中，指定下列資訊。
   - 輸入 Endpoint name (端點名稱) 的名稱(例如，**`VPC_A_Test`**)。
   - 針對 VPC in the region (區域中的 VPC)，選擇您在建立 VPC 端點時使用的同一個 VPC。
   - 設定 Security group for this endpoint (此端點的安全群組)，以允許內部部署網路的傳入流量。
   - 針對 IP 地址，選擇 Use an IP address that is selected automatically (使用自動選取的 IP 地址)。
3. 選擇 Submit (提交) 來建立您的傳入解析程式。

針對這個範例，讓我們假設 IP 地址 `10.100.0.145` 和 `10.100.192.10` 指派給路由流量的傳入 Route 53 解析程式。

後續步驟

您已建立私有託管區域和傳入解析程式，以路由 DNS 項目的流量。您現在可以使用 Site-to-Site VPN 或 Client VPN 端點。如需詳細資訊，請參閱使用 VPN 將 LoRa 閘道連接至您的 AWS 帳戶。

使用 VPN 將 LoRa 閘道連接至您的 AWS 帳戶

若要將閘道內部部署連接至 AWS 帳戶，您可以使用 Site-to-Site VPN 連線或 Client VPN 端點。

在可以連接您的內部部署閘道之前，您必須先建立 VPC 端點，並設定私有託管區域和傳入解析程式，以便來自閘道的流量不會通過公有網際網路。如需詳細資訊，請參閱建立 VPC 介面端點和私有託管區域。

Site-to-Site VPN 端點

如果您沒有閘道硬體，或想要使用不同的 AWS 帳戶 測試 VPN 連線，則可以使用 Site-to-Site VPN 連線。您可以使用 Site-to-Site VPN，從相同的 AWS 帳戶 或您可能在不同 AWS 區域 中使用的另一個 AWS 帳戶 連接至 VPC 端點。

> **Note**
>
> 如果您有閘道硬體，而且想要設定 VPN 連線，我們建議您改用 Client VPN。如需指示，請參閱 用戶端 VPN 端點。

若要設定 Site-to-Site VPN：

1. 在您要從中設定連線的網站中建立另一個 VPC。對於 `VPC-A`，您可以重複使用先前建立的 VPC。若要建立另一個 VPC (例如，`VPC-B`)，請使用未與您先前所建立 VPC 之 CIDR 區塊重疊的 CIDR 區塊。

   如需設定 VPC 的相關資訊，請遵循 AWS 設定 Site-to-Site VPN 連線中所述的指示。

   > **Note**
   >
   > 文件中所述的 Site-to-Site VPN 方法會使用 OpenSWAN 進行 VPN 連線，而 VPN 連線只支援一個 VPN 通道。如果您針對 VPN 使用不同的商務軟體，您或許可以在網站之間設定兩個通道。

2. 在設定 VPN 連線之後，請從您的 AWS 帳戶 新增傳入解析程式的 IP 地址來更新 `/etc/resolv.conf` 檔案。您可以針對名稱伺服器使用此 IP 地址。如需如何取得此 IP 地址的相關資訊，請參閱 設定 Route 53 傳入解析程式。在這個範例中，我們可以使用您在建立 Route 53 解析程式時所指派的 IP 地址 `10.100.0.145`。

   ```
   options timeout:2 attempts:5
   ```

```
; generated by /usr/sbin/dhclient-script
search region.compute.internal
nameserver 10.100.0.145
```

3. 我們現在可以使用 nslookup 命令，測試 VPN 連線是否使用 AWS PrivateLink 端點，而不是通過公有網際網路。下列顯示執行命令的範例。

```
nslookup account-specific-prefix.lns.lorawan.region.amazonaws.com
```

下列顯示執行命令的範例輸出，其中顯示私有 IP 地址，指出已建立與 AWS PrivateLink LNS 端點的連線。

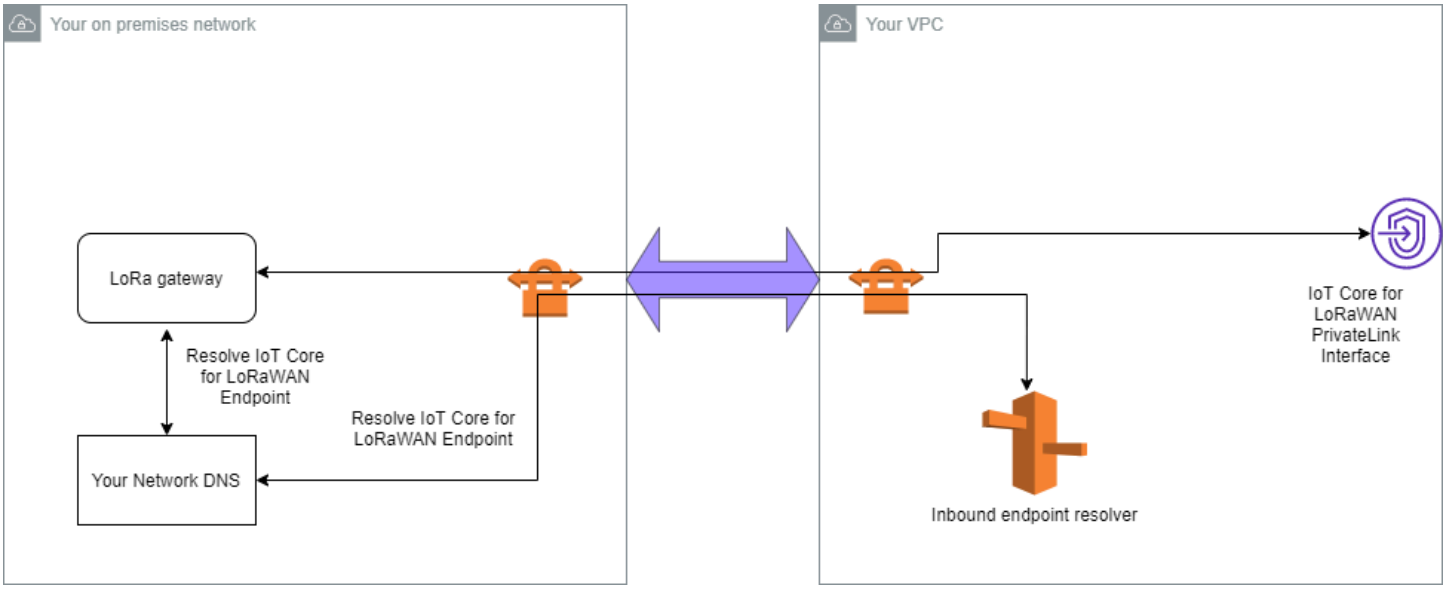
```
Server: 10.100.0.145
Address: 10.100.0.145

Non-authoritative answer:
Name: https://xxxxx.lns.lorawan.region.amazonaws.com
Address: 10.100.0.204
```

如需使用 Site-to-Site VPN 連線的相關資訊，請參閱 Site-to-Site VPN 的運作方式。

用戶端 VPN 端點

AWS Client VPN 是以用戶端為基礎的受管 VPN 服務，能讓您安全地存取 AWS 資源，以及內部部署網路中的資源。下列顯示用戶端 VPN 服務的架構。



若要建立與 Client VPN 端點的 VPN 連線：

1. 遵循 AWS Client VPN 入門中所述的指示來建立 Client VPN 端點。
2. 使用該路由器的存取 URL (例如 192.168.1.1) 登入您的內部部署網路 (例如 Wi-Fi 路由器)，然後尋找根名稱和密碼。
3. 遵循閘道文件中的指示來設定 LoRaWAN 閘道，然後將閘道新增至 AWS IoT Core for LoRaWAN。如需如何新增閘道的相關資訊，請參閱 將閘道加入 AWS IoT Core for LoRaWAN。
4. 檢查閘道的韌體是否為最新版本。如果韌體過期，您可以遵循內部部署網路中提供的指示來更新閘道的韌體。如需詳細資訊，請參閱使用 CUPS 服務搭配 AWS IoT Core for LoRaWAN 來更新閘道韌體。
5. 檢查 OpenVPN 是否已啟用。如果已啟用，請跳至下一個步驟，在內部部署網路內設定 OpenVPN 用戶端。如果尚未啟用，請遵循 OpenVPN for OpenWrt 安裝指南中的指示。

   > **Note**
   >
   > 對於此範例，我們使用 OpenVPN。您可以使用其他 VPN 用戶端 (例如 AWS VPN 或 AWS Direct Connect)，來設定 Client VPN 連線。

6. 根據來自用戶端組態的資訊，以及如何使用 LuCi 來使用 OpenVPN 用戶端，設定 OpenVPN 用戶端。
7. 透過 SSH 連接到您的內部署網路，並更新 /etc/resolv.conf 檔案，方法為在 AWS 帳戶 (10.100.0.145) 中新增傳入解析程式的 IP 地址。
8. 對於要使用 AWS PrivateLink 連接到端點的閘道流量，請將閘道的第一個 DNS 項目取代為傳入解析程式的 IP 地址。

如需使用 Site-to-Site VPN 連線的相關資訊，請參閱 Client VPN 入門。

連接至 LNS 和 CUPS VPC 端點

下列顯示如何測試您與 LNS 和 CUPS VPC 端點的連線。

測試 CUPS 端點

若要從 LoRa 閘道測試 AWS PrivateLink 與 CUPS 端點的連線，請執行下列命令：

```
curl -k -v -X POST https://xxxx.cups.region.iotwireless.iot:443/update-info
     --cacert cups.trust --cert cups.crt --key cups.key --header "Content-Type:
 application/json"
```

```
    --data '{
             "router": "xxxxxxxxxxxx",
             "cupsUri": "https://xxxx.cups.lorawan.region.amazonaws.com:443",
             "cupsCredCrc":1234, "tcCredCrc":552384314
            }'
     –output cups.out
```

測試 LNS 端點

若要測試 LNS 端點，首先佈建將使用無線閘道的 LoRaWAN 裝置。然後，您可以新增裝置並執行「聯結」程序，然後您就可以開始傳送上行訊息。

適用於 Amazon Sidewalk 的 AWS IoT Core

適用於 Amazon Sidewalk 的 AWS IoT Core 提供雲端服務，您可以使用這些服務將 Sidewalk 終端裝置連線至 AWS 雲端 並使用其他 AWS 服務。

Amazon Sidewalk 是安全的共用網路，可讓您社區中的裝置連線並保持連線。Amazon Sidewalk 在 Sidewalk 終端裝置和 Sidewalk 閘道之間，以及 Sidewalk 閘道和 Sidewalk 雲端之間傳輸資料。

存取適用於 Amazon Sidewalk 的 AWS IoT Core

您可以使用主控台或 AWS IoT Wireless API 操作將 Sidewalk 終端裝置加入 AWS IoT。裝置加入後，其訊息便會傳送至 AWS IoT Core。然後，您可以開始在使用來自 Amazon Sidewalk 終端裝置之資料的 AWS 雲端上開發商業應用程式。

使用主控台

若要加入 Sidewalk 終端裝置，請登入AWS Management Console並瀏覽至 AWS IoT 主控台上的裝置頁面。裝置加入後，您可以在 IoT 主控台的此頁面上檢視和管理裝置。

使用 API 或 CLI

您可以使用 AWS IoT Wireless API 操作加入 Sidewalk 和 LoRaWAN 裝置。AWS SDK 支援用於建置 AWS IoT Core 的 AWS IoT Wireless API。如需詳細資訊，請參閱 AWS 開發套件與工具組。

您可以使用 AWS CLI 執行命令來加入和管理 Sidewalk 終端裝置。如需詳細資訊，請參閱 AWS IoT Wireless CLI 參考。

適用於 Amazon Sidewalk 的 AWS IoT Core 區域和端點

Amazon Sidewalk 僅適用於`us-east-1` AWS 區域。適用於 Amazon Sidewalk 的 AWS IoT Core 為此區域中的控制平面和資料平面 API 端點提供支援。專屬於您 AWS 帳戶的資料平面 API 端點。如需詳細資訊，請參閱《AWS 一般參考》中的 AWS IoT Wireless 服務端點。

適用於 Amazon Sidewalk 的 AWS IoT Core 擁有的配額可用於在裝置和 AWS 雲端 之間傳輸的資料，且具有適用於 AWS IoT Wireless API 操作的最大 TPS。如需詳細資訊，請參閱《AWS 一般參考》中的 AWS IoT Wireless 配額。

適用於 Amazon Sidewalk 的 AWS IoT Core 定價

在註冊 AWS 時，您可以透過 AWS 免費方案免費開始使用適用於 Amazon Sidewalk 的 AWS IoT Core。

如需一般產品概觀和定價的詳細資訊，請參閱 AWS IoT Core 定價。

什麼是適用於 Amazon Sidewalk 的 AWS IoT Core？

使用適用於 Amazon Sidewalk 的 AWS IoT Core，您可以將 Amazon Sidewalk 終端裝置加入 AWS IoT，方便管理和監控。它也能管理將裝置資料傳送至其他 AWS 服務 的目的地。

適用於 Amazon Sidewalk 的 AWS IoT Core 的功能

使用適用於 Amazon Sidewalk 的 AWS IoT Core，您可以：

- 使用 AWS IoT 主控台、適用於 Amazon Sidewalk 的 AWS IoT Core API 操作或 AWS CLI 命令，將 Sidewalk 終端裝置加入 AWS IoT。
- 利用 AWS 雲端 提供的功能。
- 建立目的地，以使用 AWS IoT 規則來處理傳入承載訊息並與其他 AWS 服務互動。
- 啟用事件通知以接收有關事件的訊息，例如 Sidewalk 終端裝置的佈建或註冊時間，或者下行訊息是否已成功傳遞至您的裝置。
- 即時記錄和監控您的 Sidewalk 終端裝置、取得有用的見解，以及識別和疑難排解錯誤。
- 將 Sidewalk 終端裝置與 AWS IoT 物件建立關聯，以協助將裝置的表示形式儲存在雲端上。AWS IoT 中的物件可讓您更輕鬆地搜尋和管理您的功能，以及存取其他 AWS IoT Core 功能。

以下主題將協助您了解 Amazon Sidewalk 和適用於 Amazon Sidewalk 的 AWS IoT Core。

主題

- 什麼是 Amazon Sidewalk？
- 適用於 Amazon Sidewalk 的 AWS IoT Core 如何運作

什麼是 Amazon Sidewalk？

Amazon Sidewalk 是安全的社區網絡，它使用 Amazon Sidewalk Bridges，例如相容的 Amazon Echo 和 Ring 裝置，為 IoT 裝置提供雲端連線。Amazon Sidewalk 可在家中以及其他地方使用 Bluetooth LE 進行短距離通訊，以及使用 900MHz 頻率的 LoRa 和 FSK 無線電通訊協定，以涵蓋更長的距離。

Amazon Sidewalk 啟用時，此網路可以支援您社區中的其他 Sidewalk 終端裝置，並可以用於感應您的環境的應用程式，例如感應您的環境。Amazon Sidewalk 可協助裝置連線並保持連線。

Amazon Sidewalk 的功能

以下是 Amazon Sidewalk 的功能：

- Amazon Sidewalk 使用包括環狀和特定 Echo 裝置的 Sidewalk 閘道，建立低頻寬網路。使用閘道，您可以共用部分網際網路頻寬，然後將其用於將終端裝置連接到網路。
- Amazon Sidewalk 提供具有多層加密和安全性的安全聯網機制。
- Amazon Sidewalk 提供一種簡單的機制來啟用或停用 Sidewalk 的參與。

Amazon Sidewalk 概念

以下是 Amazon Sidewalk 的一些關鍵概念。

Sidewalk 閘道

Sidewalk 閘道 (或 Amazon Sidewalk 橋接器) 可在 Sidewalk 終端裝置和雲端之間路由資料。閘道是支援 SubG-CSS (非同步、LDR)、SubG-FSK (同步、HDR) 或用於 Sidewalk 通訊的 Bluetooth LE 的 Amazon 裝置，例如 Echo 裝置或環形泛光燈攝影機。Sidewalk 閘道會與 Sidewalk 社區共用部分網際網路頻寬，以便為一組支援 Sidewalk 的裝置提供連線能力。

Sidewalk 終端裝置

Sidewalk 終端裝置可連接至 Sidewalk 閘道，在 Amazon Sidewalk 上漫遊。終端裝置是低頻寬、低功耗的智慧型產品，例如支援 Sidewalk 的燈或門鎖。

> **Note**
>
> 某些 Sidewalk 閘道也可以充當終端裝置。

Sidewalk 網路伺服器

由 Amazon 操作的 Sidewalk 網路伺服器會驗證傳入封包，並將上行和下行訊息路由到所需的目的地，同時保持 Sidewalk 網路時間同步。

進一步了解 Amazon Sidewalk

如需 Amazon Sidewalk 的詳細資訊，請參閱下列網頁：

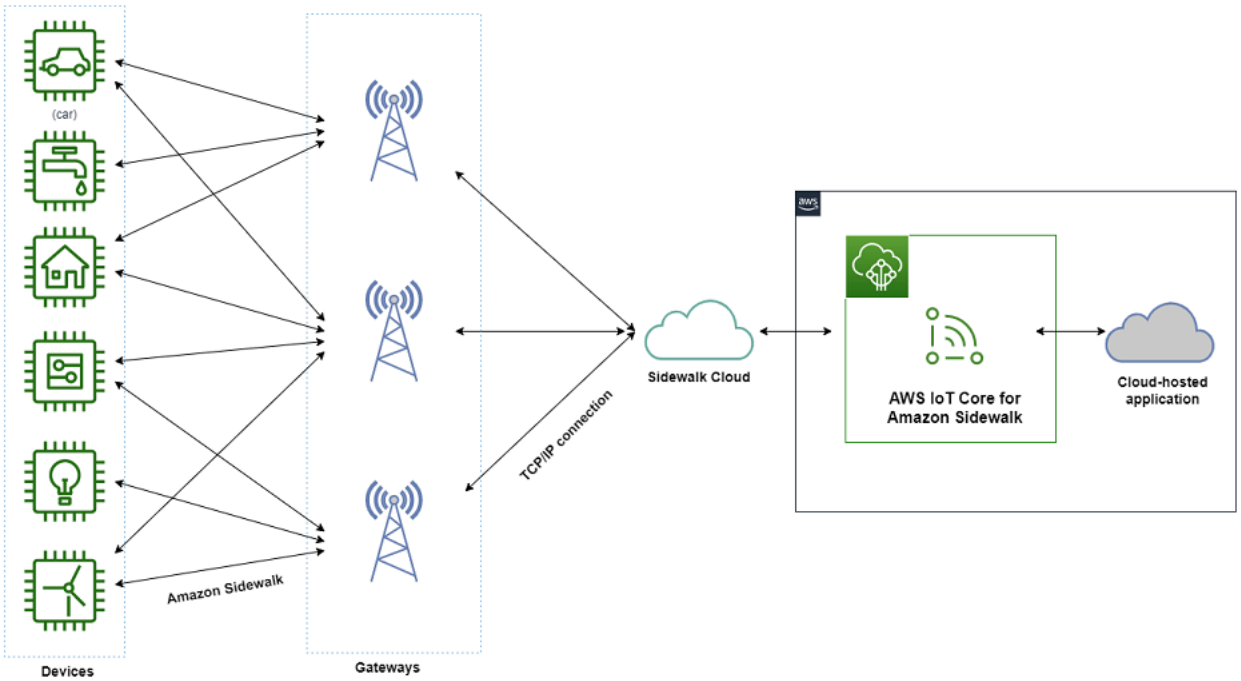
- Amazon Sidewalk
- Amazon Sidewalk 文件
- 適用於 Amazon Sidewalk 的 AWS IoT Core

適用於 Amazon Sidewalk 的 AWS IoT Core 如何運作

使用適用於 Amazon Sidewalk 的 AWS IoT Core，您可以將 Amazon Sidewalk 終端裝置加入 AWS IoT，方便管理和監控。它也管理將裝置資料傳送至其他 AWS 服務的目的地。

適用於 Amazon Sidewalk 的 AWS IoT Core 提供雲端服務，您可以使用這些服務將 Sidewalk 終端裝置連線至 AWS 雲端 並使用其他 AWS 服務。您也可以使用適用於 Amazon Sidewalk 的 AWS IoT Core 來管理 Sidewalk 裝置，以及監控和建置應用程式。

Sidewalk 終端裝置可透過 Sidewalk 閘道與 AWS IoT Core 進行通訊。適用於 Amazon Sidewalk 的 AWS IoT Core 會管理 AWS IoT Core 需要的服務和裝置政策，以便管理 Sidewalk 終端裝置和閘道並與之通訊。它也管理將裝置資料傳送至其他 AWS 服務的目的地。



使用適用於 Amazon Sidewalk 的 AWS IoT Core 開始進行

您可以使用 AWS IoT 主控台、適用於 Amazon Sidewalk 的 AWS IoT Core API 或 AWS CLI 來建立和加入 Sidewalk 終端裝置，以及將其連線至 Sidewalk 網路。如需開始使用 Amazon Sidewalk 以及將終端裝置加入 AWS IoT 的相關資訊，請參閱下列主題。

- 適用於 AWS IoT Core 的 Amazon Sidewalk 入門

  本主題逐步介紹加入 Sidewalk 終端裝置的先決條件、說明使用感應器監控應用程式的工作流程，並提供如何使用 AWS CLI 命令加入裝置的概觀。

- 連線至適用於 Amazon Sidewalk 的 AWS IoT Core

  本節說明加入工作流程簡介中的不同步驟，並引導您使用主控台以及 API 操作加入終端裝置。您也會連線裝置，並檢視在裝置和適用於 Amazon Sidewalk 的 AWS IoT Core 之間交換的訊息。

- 使用適用於 Amazon Sidewalk 的 AWS IoT Core 大量佈建裝置

  本節提供詳細的逐步教學課程，說明如何使用適用於 Amazon Sidewalk 的 AWS IoT Core 大量佈建您的 Sidewalk 終端裝置。您會學到大量佈建工作流程，以及如何加入大量 Sidewalk 裝置。

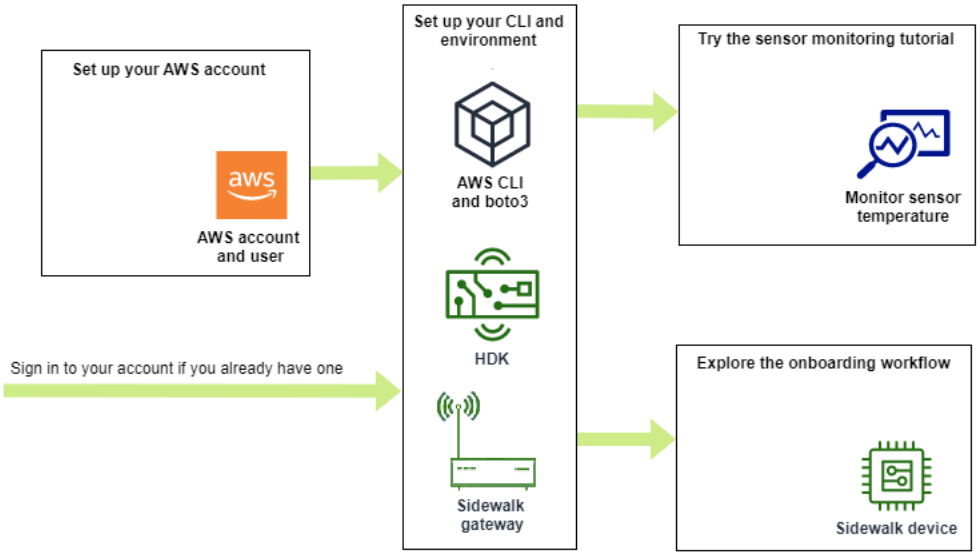
進一步了解適用於 Amazon Sidewalk 的 AWS IoT Core

如需適用於 Amazon Sidewalk 的 AWS IoT Core 的詳細資訊，請參閱下列網頁：

- Amazon Sidewalk
- Amazon Sidewalk 文件
- 適用於 Amazon Sidewalk 的 AWS IoT Core

適用於 AWS IoT Core 的 Amazon Sidewalk 入門

本節說明如何開始將您的 Sidewalk 終端裝置連線至適用於 Amazon Sidewalk 的 AWS IoT Core。此會說明如何將終端裝置連接至 Amazon Sidewalk，並在之間傳遞訊息。您還會了解 Sidewalk 範例應用程式，以及如何使用適用於 Amazon Sidewalk 的 AWS IoT Core 執行感測器監控的概觀。範例應用程式為您提供儀表板，以檢視並監控感測器溫度的變化。



下列主題將協助您開始使用適用於 Amazon Sidewalk 的 AWS IoT Core。

主題

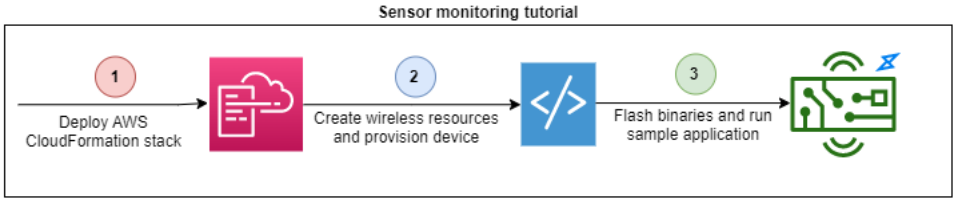
- 試用感測器監控教學課程
- 加入 Sidewalk 裝置簡介

試用感測器監控教學課程

本節提供 GitHub 上 Amazon Sidewalk 範例應用程式的概觀，其展示如何監控感測器的溫度。於本教學課程中，您會使用指令碼，以程式設計方式建立所需的無線資源、佈建終端裝置並重新整理二進位檔案，然後將終端裝置連接至應用程式。使用 AWS CLI 和 Python 命令的指令碼會建立 AWS

CloudFormation 堆疊和無線資源，然後重新整理二進位檔案，並將應用程式部署至您的硬體開發套件 (HDK)。

下圖顯示執行範例應用程式並將 Sidewalk 終端裝置連接至應用程式時所涉及的步驟。如需本教學課程的詳細說明，包括先決條件和組態，請參閱 GitHub 中的 README 文件。



加入 Sidewalk 裝置簡介

本節說明如何將您的 Sidewalk 終端裝置加入適用於 Amazon Sidewalk 的 AWS IoT Core。如要加入裝置，請先新增您的 Sidewalk 裝置，接著佈建並註冊您的裝置，然後將硬體連接至雲端應用程式。執行本教學課程之前，請先檢閱並完成 安裝 Python 和 AWS CLI。

下列步驟示範如何將 Sidewalk 終端裝置加入適用於 Amazon Sidewalk 的 AWS IoT Core 並進行連線。若您要使用 AWS CLI 加入裝置，則可參閱本節中所提供的範例命令。如需使用 AWS IoT 主控台加入裝置的相關資訊，請參閱 連線至適用於 Amazon Sidewalk 的 AWS IoT Core。

> **Important**
>
> 如要執行整個加入工作流程，您還需要佈建及註冊您的終端裝置，並連接您的硬體開發套件 (HDK)。如需詳細資訊，請參閱《Amazon Sidewalk 文件》中的佈建和註冊您的終端裝置。

主題

- 步驟 1：將您的 Sidewalk 裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core
- 步驟 2：建立您 Sidewalk 終端裝置的目的地
- 步驟 3：佈建和註冊終端裝置
- 步驟 4：連接至 Sidewalk 終端裝置並交換訊息

步驟 1：將您的 Sidewalk 裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core

以下是將 Sidewalk 終端裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core 的步驟概觀。儲存您取得之裝置設定檔及所建立無線裝置的相關資訊。您將使用此資訊來佈建並註冊終端裝置。如需這些步驟的詳細資訊，請參閱 將您的裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core。

1. 建立裝置設定檔

   建立一個包含 Sidewalk 裝置共用組態的裝置設定檔。建立設定檔時，請將設定檔的 *name* 指定為英數字串。如要建立設定檔，請移至 AWS IoT 主控台中設定檔中樞的 Sidewalk 索引標籤，並選擇建立設定檔，或使用 CreateDeviceProfile API 操作或 create-device-profile CLI 命令，如此範例所示。

   ```
   // Add your device profile using a name and the sidewalk object.
   aws iotwireless create-device-profile --name sidewalk_profile --sidewalk {}
   ```

2. 建立您的 Sidewalk 終端裝置

   使用適用於 Amazon Sidewalk 的 AWS IoT Core 建立您的 Sidewalk 終端裝置。指定目的地名稱及從先前步驟中所取得的裝置設定檔 ID。如要新增設定檔，請移至 AWS IoT 主控台中設定檔中樞的 Sidewalk 索引標籤，並選擇佈建裝置，或使用 CreateWirelessDevice API 操作或 create-wireless-device CLI 命令，如此範例所示。

   > **Note**
   >
   > 請為您的 AWS 帳戶 和 AWS 區域 指定一個唯一的目的地名稱。您會使用將目的地新增至適用於 Amazon Sidewalk 的 AWS IoT Core 時的相同目的地名稱。

   ```
   // Add your Sidewalk device by using the device profile ID.
   aws iotwireless create-wireless-device --type "Sidewalk" --name sidewalk_device \
       --destination-name SidewalkDestination \
       --sidewalk DeviceProfileId="12345678-234a-45bc-67de-e8901234f0a1"
   ```

3. 取得裝置設定檔和無線裝置資訊

   以 JSON 格式取得裝置設定檔和無線裝置資訊。JSON 將會包含裝置詳細信訊、裝置憑證、私有金鑰、DeviceTypeId 和 Sidewalk 製造序號 (SMSN) 的相關資訊。

   - 若您使用 AWS IoT 主控台，則可使用裝置中樞的 Sidewalk 索引標籤，為您的 Sidewalk 終端裝置下載合併的 JSON 檔案。
   - 若您正在使用 API 操作，請將從 API 操作 GetDeviceProfile 和 GetWirelessDevice 取得的回應儲存為個別的 JSON 檔案，例如 *device_profile.json* 和 *wireless_device.json*。

```
// Store device profile information as a JSON file.
aws iotwireless get-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" > device_profile.json

// Store wireless device information as a JSON file.
aws iotwireless get-wireless-device --identifier-type WirelessDeviceId \
    --identifier "23456789-abcd-0123-bcde-fabc012345678" > wireless_device.json
```

步驟 2：建立您 Sidewalk 終端裝置的目的地

以下是將目的地新增至適用於 Amazon Sidewalk 的 AWS IoT Core 的步驟概觀。使用 AWS Management Console、AWS IoT Wireless API 操作或 AWS CLI，執行下列步驟來建立 AWS IoT 規則和目的地。然後，您可以連接到硬體平台，並檢視和交換訊息。如需本節中用於 AWS CLI 範例的範例 IAM 角色和 AWS IoT 規則，請參閱 為您的目的地建立 IAM 角色和 IoT 規則。

1. 建立 IAM 角色

   建立 IAM 角色，將傳送資料至 AWS IoT 規則的許可授予適用於 Amazon Sidewalk 的 AWS IoT Core。如要建立該角色，請使用 CreateRole API 操作或 create-role CLI 命令。您可將該角色命名為 *SidewalkRole*。

   ```
   aws iam create-role --role-name lambda-ex \
       --assume-role-policy-document file://lambda-trust-policy.json
   ```

2. 建立一個目的地規則

   建立將要處理裝置資料的 AWS IoT 規則名稱，並指定要發佈訊息的主題。連接至硬體平台後，您將會觀察此主題的訊息。請使用 AWS IoT Core API 操作 CreateTopicRule，或 AWS CLI 命令 create-topic-rule，來建立一個目的地規則。

   ```
   aws iot create-topic-rule --rule-name Sidewalkrule \
       --topic-rule-payload file://myrule.json
   ```

3. 建立目的地

   建立一個將您的 Sidewalk 裝置與 IoT 規則相關聯的目的地，其會處理該規則以與其他 AWS 服務 搭配使用。您可使用 AWS IoT 主控台的目的地中樞、或 CreateDestination API 操作或 create-destination CLI 命令來新增目的地。

```
aws iotwireless create-destination --name SidewalkDestination \
    --expression-type RuleName --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

步驟 3：佈建和註冊終端裝置

使用 Python 命令，您可以佈建和註冊您的終端裝置。佈建指令碼會使用您取得的裝置 JSON 資料，來產生一個製造二進位影像，然後再於硬體主機板上加以重新整理。接著，您可註冊終端裝置以連接至硬體平台。如需詳細資訊，請參閱《Amazon Sidewalk 文件》中的佈建和註冊您的終端裝置。

> **Note**
>
> 註冊 Sidewalk 終端裝置時，您的閘道必須選擇加入 Amazon Sidewalk，且閘道和裝置必須在彼此的範圍內。

步驟 4：連接至 Sidewalk 終端裝置並交換訊息

註冊您的終端裝置後，則可接著連接您的終端裝置並開始交換訊息和裝置資料。

1. 連接您的 Sidewalk 終端裝置

   將 HDK 連接至您的電腦，並依照廠商說明文件提供的指示連接至您的 HDK。如需詳細資訊，請參閱《Amazon Sidewalk 文件》中的佈建和註冊您的終端裝置。

2. 檢視和交換訊息

   使用 MQTT 用戶端來訂閱指定於規則中的主題，並檢視所收到的訊息。您還可使用 `SendDataToWirelessDevice` API 操作或 `send-data-to-wireless-device` CLI 命令，將下行訊息傳送至您的裝置，並驗證連線狀態。

   (選用) 您可啟用訊息傳遞狀態事件，檢查是否已成功接收下行訊息。

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

連線至適用於 Amazon Sidewalk 的 AWS IoT Core

本節展示如何加入 Sidewalk 終端裝置，然後將您的裝置連接至 Sidewalk 網路。其描述了您於加入教學課程中所執行的步驟，如 加入 Sidewalk 裝置簡介 中所述。您將了解如何使用 AWS IoT 主控台和適用於 Amazon Sidewalk 的 AWS IoT Core API 操作來加入裝置。您還會了解執行這些作業的 AWS CLI 命令。

必要條件

如要將終端裝置和目的地新增至適用於 Amazon Sidewalk 的 AWS IoT Core，您必須設定 AWS 帳戶。如要使用 AWS IoT Wireless API 或 AWS CLI 命令執行這些操作，您還必須設定 AWS CLI。如需有關先決條件與設定的更多資訊，請參閱 安裝 Python 和 AWS CLI。

> **Note**
>
> 如要執行佈建和註冊終端裝置及連線至硬體開發套件 (HDK) 的完整加入工作流程，您還必須設定 Sidewalk 閘道和 HDK。如需詳細資訊，請參閱《Amazon Sidewalk 文件》中的設定硬體開發套件 (HDK) 和設定 Sidewalk 閘道。

描述您的 Sidewalk 資源

在您開始使用並建立資源之前，建議您考慮 Sidewalk 終端裝置、裝置設定檔和目的地的命名慣例。適用於 Amazon Sidewalk 的 AWS IoT Core 會為您建立的資源指派一個唯一識別碼。但是，您可為其提供更多描述性的名稱，新增描述或新增選用標籤，協助其進行識別和管理。

> **Note**
>
> 目的地名稱建立後便無法變更。使用一個對您 AWS 帳戶 和 AWS 區域 的唯一名稱。

如需詳細資訊，請參閱描述您的 AWS IoT Wireless 資源。

主題

- 將您的裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core
- 為您的 Sidewalk 終端裝置新增目的地
- 連接您的 Sidewalk 裝置並檢視上行中繼資料格式

將您的裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core

建立無線裝置之前，請先建立裝置設定檔。裝置設定檔定義 Sidewalk 裝置的裝置功能及其他參數。單一裝置設定檔可與多個裝置建立關聯。

建立裝置設定檔後，當您擷取設定檔的相關資訊時，其會傳回一個 `DeviceTypeId`。當您佈建終端裝置時，將會使用此 ID、裝置憑證、應用程式伺服器公有金鑰和 SMSN。

如何建立和新增您的裝置

1. 建立您 Sidewalk 終端裝置的裝置設定檔。將要用於您 Sidewalk 裝置的設定檔名稱指定為英數字串。設定檔將有助於識別要與其關聯的裝置。
   - (主控台) 新增您的 Sidewalk 裝置時，您也可建立新的設定檔。這樣有助於快速將您的裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core，並將其與設定檔建立關聯。
   - (API) 透過指定設定檔名稱和 Sidewalk 物件 `sidewalk {}` 來使用 `CreateDeviceProfile` API 操作。API 回應將包含設定檔 ID 和 ARN (Amazon Resource Name)。
2. 將您的無線裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core。指定目的地名稱，並選擇您於上一個步驟中所建立的裝置設定檔。
   - (主控台) 新增 Sidewalk 裝置時，請輸入目的地名稱，然後選擇您所建立的設定檔。
   - (API) 使用 `CreateWirelessDevice` API 操作。指定目的地名稱及先前所取得的裝置設定檔 ID。

   無線裝置參數

參數	描述	備註
目的地名稱	目的地名稱，其用於描述處理其他 AWS 服務 將會使用之裝置資料的 AWS IoT 規則。	若您尚未建立目的地，則可提供任何字串值。適用於 Amazon Sidewalk 的 AWS IoT Core 將在建立裝置時建立一個空白的目的地，您後續可於新增目的地時進行更新。
裝置設定檔	您先前建立的裝置設定檔。	–

3. 取得包含佈建終端裝置所需資訊的 JSON 檔案。
   - (主控台) 從您所建立 Sidewalk 裝置的詳細資訊頁面下載此檔案。

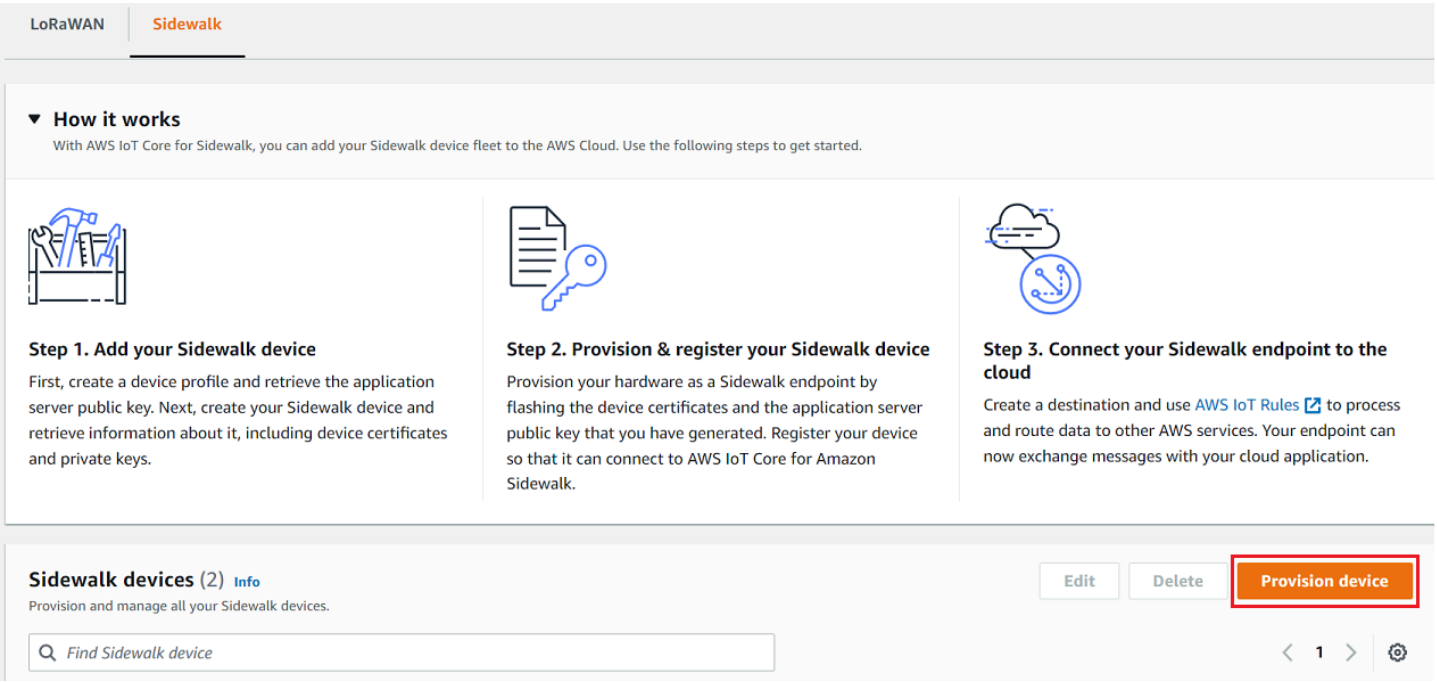
- (API) 使用 `GetDeviceProfile` 和 `GetWirelessDevice` API 操作擷取有關您裝置設定檔和無線裝置的資訊。將 API 回應資訊儲存為 JSON 檔案，例如 *`device_profile.json`* 和 *`wireless_device.json`*。

新增您的裝置設定檔和 Sidewalk 終端裝置

本節展示如何建立裝置設定檔。此外也將說明如何使用 AWS IoT 主控台和 AWS CLI，將 Sidewalk 終端裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core。

新增您的 Sidewalk 裝置 (主控台)

如要使用 AWS IoT 主控台新增您的 Sidewalk 裝置，請移至裝置中樞的 Sidewalk 索引標籤，選擇佈建裝置，然後執行下列步驟。



1. 指定裝置詳細資訊

   指定您 Sidewalk 裝置的組態資訊。您還可建立新的裝置設定檔，或為您的 Sidewalk 裝置選擇現有的設定檔。

   a. 指定裝置名稱和選用說明。該說明的長度最多可達 2,048 個字元。您可在建立裝置後編輯這些欄位。

   b. 選擇要與 Sidewalk 裝置建立關聯的裝置設定檔。若您有任何現有的裝置設定檔，則可選擇您的設定檔。如要建立新的設定檔，請選擇建立新設定檔，接著輸入該設定檔的名稱。

> **Note**
>
> 如要將標籤附加至裝置設定檔，在建立設定檔之後，請移至設定檔中樞，然後編輯您的設定檔以新增此資訊。

c. 指定將訊息從您的裝置路由至其他 AWS 服務 的目的地名稱。若您尚未建立目的地，請移至目的地中樞，來建立目的地。接著，您可選擇您 Sidewalk 裝置的目的地。如需詳細資訊，請參閱為您的 Sidewalk 終端裝置新增目的地。

d. 選擇下一步，繼續新增您的 Sidewalk 裝置。

2. 將 Sidewalk 裝置與 AWS IoT 物件產生關聯 (選用)

   您可選擇將 Sidewalk 裝置與 AWS IoT 物件產生關聯。IoT 物件是 AWS IoT 裝置登錄檔中的項目。物件可讓您更輕鬆地搜尋和管理您的裝置。將物件與您的裝置相關連可讓您的裝置存取其他 AWS IoT Core 功能。

   如要將裝置與物件建立關聯，請選擇自動物件註冊。

   a. 為您要與 Sidewalk 裝置建立關聯的 IoT 物件輸入唯一名稱。物件名稱區分大小寫，且在 AWS 帳戶 和 AWS 區域 中必須是唯一的。

   b. 為您的 IoT 物件提供任何其他組態，例如使用物件類型，或可用來從物件清單中篩選的可搜尋屬性。

   c. 選擇下一步，並驗證 Sidewalk 裝置的相關資訊，然後選擇建立。

新增您的 Sidewalk 裝置 (CLI)

如要新增您的 Sidewalk 裝置，並下載將用來佈建您 Sidewalk 裝置的 JSON 檔案，請執行下列 API 操作。

主題

- 步驟 1：建立裝置設定檔
- 步驟 2：新增您的 Sidewalk 裝置

步驟 1：建立裝置設定檔

如要於 AWS 帳戶 中建立一個裝置設定檔，請使用 `CreateDeviceProfile` API 操作或 `create-device-profile` CLI 命令。當您建立裝置設定檔時，請指定名稱並提供任何選用標籤為名稱/值組。

例如，下列命令會建立 Sidewalk 裝置的裝置設定檔。

```
aws iotwireless create-device-profile \
    --name sidewalk_profile --sidewalk {}
```

執行此命令會傳回 Amazon Resource Name (ARN) 和裝置設定檔 ID 作為輸出。

```
{
    "DeviceProfileArn": "arn:aws:iotwireless:us-
east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

步驟 2：新增您的 Sidewalk 裝置

如要將您的 Sidewalk 裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core 帳戶，請使用 CreateWirelessDevice API 操作或 create-wireless-device CLI 命令。當您建立裝置時，除了您 Sidewalk 裝置的選用名稱和描述之外，請指定下列參數。

> **Note**
>
> 若您想要建立 Sidewalk 裝置與 AWS IoT 物件的關聯，請使用 AssociateWirelessDeviceWithThing API 操作或 associate-wireless-device-with-thing CLI 命令。

下列命令顯示建立 Sidewalk 裝置的範例：

```
aws iotwireless create-wireless-device \
     --cli-input-json "file://device.json"
```

下列顯示 device.json 檔案的內容。

device.json 的內容

```
{
  "Type": "Sidewalk",
  "Name": "SidewalkDevice",
  "DestinationName": "SidewalkDestination",
```

```
  "Sidewalk": {
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
    }
}
```

執行此命令會傳回裝置 ID 與 Amazon Resource Name (ARN) 作為輸出。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/23456789-abcd-0123-bcde-fabc012345678",
    "Id": "23456789-abcd-0123-bcde-fabc012345678"
}
```

取得裝置 JSON 檔案，進行佈建

將 Sidewalk 裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core 後，請下載包含佈建終端裝置所需資訊的 JSON 檔案。您可以使用 AWS IoT 主控台或 AWS CLI 來擷取此資訊。如需有關如何佈建裝置的詳細資訊，請參閱《Amazon Sidewalk 文件》中的佈建和註冊您的終端裝置。

取得 JSON 檔案 (主控台)

如要取得 JSON 檔案，以佈建您的 Sidewalk 裝置：

1. 請移至 Sidewalk 裝置中樞。
2. 選擇您新增至適用於 Amazon Sidewalk 的 AWS IoT Core 的裝置，以檢視其詳細資訊。
3. 在您新增裝置的詳細資料頁面中選擇下載裝置 JSON 檔案，以取得 JSON 檔案。

   將會下載 `certificate.json` 檔案，其包含佈建終端裝置所需的資訊。下列顯示範本 JSON 檔案。其包含裝置憑證、私有金鑰、Sidewalk 製造序號 (SMSN) 和 `DeviceTypeID`。

   ```
   {
     "p256R1": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbH ... DANKkOKoNT3bUGz+/f/pyTE+xMRdIUBZ1Bw==",
     "eD25519": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbHD ... UiZmntHiUr1GfkTOFMYqRB+Aw==",
     "metadata": {
       "devicetypeid": "fe98",
       "applicationDeviceArn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/897ce68e-3ca2-4ed0-85a2-30b0666c4052",
       "applicationDeviceId": "897ce68e-3ca2-4ed0-85a2-30b0666c4052",
       "smsn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A4A",
   ```

```
    "devicePrivKeyP256R1":
 "3e704bf8d319b3a475179f1d68c60737b28c708f845d0198f2d00d00c88ee018",
    "devicePrivKeyEd25519":
 "17dacb3a46ad9a42d5c520ca5f47f0167f59ce54d740aa13918465faf533b8d0"
  },
  "applicationServerPublicKey":
 "5ce29b89c2e3ce6183b41e75fe54e45f61b8bb320efbdd2abd7aefa5957a316b"
}
```

在 Sidewalk 裝置的詳細資訊頁面中，您也會看到下列相關資訊：

- 裝置 ID、其 Amazon Resource Name (ARN)，及裝置與之關聯的任何 AWS IoT 物件的相關詳細資訊。
- 裝置設定檔和目的地詳細資訊。
- 從裝置所收到最後一則上行訊息的時間。
- 指出您裝置是否已佈建或註冊的狀態。

取得 JSON 檔案 (CLI)

如要使用適用於 Amazon Sidewalk 的 AWS IoT Core API 或 AWS CLI 取得 JSON 檔案來佈建 Sidewalk 終端裝置，請將擷取裝置設定檔和無線裝置相關資訊的 API 回應以 JSON 檔案形式暫存為 *wireless_device.json* 和 *device_profile.json* 等檔案。您將會使用它們佈建您的 Sidewalk 裝置。

下列顯示如何擷取 JSON 檔案。

主題

- 步驟 1：取得裝置設定檔資訊作為 JSON 檔案
- 步驟 2：取得 Sidewalk 裝置資訊作為 JSON 檔案

步驟 1：取得裝置設定檔資訊作為 JSON 檔案

使用 `GetDeviceProfile` API 操作或 `get-device-profile` CLI 命令，以取得您為適用於 Amazon Sidewalk 的 AWS IoT Core 新增至帳戶之裝置設定檔的相關資訊。如要擷取裝置設定檔的相關資訊，請指定設定檔 ID。

接著，API 將會傳回與所指定識別符與裝置 ID 相符之裝置設定檔的資訊。您可將此回應資訊另存為檔案，並為其命名，如 *device_profile.json*。

下列顯示範例 CLI 命令：

```
aws iotwireless get-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" > device_profile.json
```

執行此命令會傳回裝置設定檔的參數、應用程式伺服器公有金鑰，和 DeviceTypeID。下列顯示 JSON 檔案，其包含來自 API 的範例回應資訊。如需有關 API 回應中參數的詳細資訊，請參閱 GetDeviceProfile。

GetDeviceProfile API 回應 (***device_profile.json*** 的內容)

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Name": "Sidewalk_profile",
    "LoRaWAN": null,
    "Sidewalk":
    {
        "ApplicationServerPublicKey":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234",
        "DAKCertificateMetadata": [
            {
                "DeviceTypeId: "fe98",
                "CertificateId": "43564A6D2D50524F544F54595045",
                "FactorySupport": false,
                "MaxAllowedSignature": 1000
            }
        ],
        "QualificationStatus": false
    }
}
```

步驟 2：取得 Sidewalk 裝置資訊作為 JSON 檔案

使用 GetWirelessDevice API 操作或 get-wireless-device CLI 命令，以取得您為適用於 Amazon Sidewalk 的 AWS IoT Core 新增至帳戶之 Sidewalk 裝置的相關資訊。如要取得終端裝置的相關資訊，請提供您在新增裝置時取得的無線裝置識別符。

接著，API 將會傳回與所指定識別符及裝置 ID 相符之裝置的資訊。將此回應資訊儲存為 JSON 檔案。為檔案提供一個有意義的名稱，例如 *wireless_device.json*。

下列顯示使用 CLI 執行命令的範例：

```
aws iotwireless get-wireless-device --identifier-type WirelessDeviceId \
    --identifier "23456789-abcd-0123-bcde-fabc012345678" > wireless_device.json
```

執行此命令會傳回裝置詳細資訊、裝置憑證、私有金鑰和 Sidewalk 製造序號 (SMSN)。下列會顯示執行此命令的範例輸出。如需有關 API 回應中參數的詳細資訊，請參閱 GetWirelessDevice。

GetWirelessDevice API 回應 (***wireless_device.json*** 的內容)

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/23456789-
abcd-0123-bcde-fabc012345678",
    "Id": "23456789-abcd-0123-bcde-fabc012345678",
    "DestinationName": "SidewalkDestination",
    "Type": "Sidewalk",
    "Sidewalk": {
        "CertificateId": "4C7438772D50524F544F54595045",
        "DeviceCertificates": [
            {
                "SigningAlg": "Ed25519",

 "Value": "hDdkJw9L2uMCORjImjMHqzNR6nYYh6QKncSl5GthQNl7NKe4ounb5UMQtLjnm7zOUPYOqghCeVOLCBUiQe2Z
F+GeltcafZcFKhS+O5NPcVNR/fHYaf/cn5iUbRwlz/T
+ODXvGdwkBkgDyFgoUJgn7JdzFjaneE5qzTWXUbL79i1sXToGGjP8hiD9jJhidPWhIswleydAWgO1OZGA4CjzIaSGVM1Vta
uMMBfgAeL8Tdv5LkFIPIB3ZX9zt8zzmAuFRzI4MuNjWfIDnOF6AKu37WWU6/
QYhZoQrW9D/wndiCcsRGl+ANn367r/HE02Re4DOiCfs9f2rjc4LT1LKt7g/KW2ii+W
+9HYvvY0bBAI+AHx6Cx4j+djabTsvrgW2k6NU2zUSM7bdDP3z2a2+Z4WzBji/jYwt/
OP8rpsy5Ee4ywXUfCsfQ0rKOr0zay6yh27p3I3MZle2oCO4JIlqK0VbIQqsXzSSyp6XXS0lhmuGugZ1AAADGz
+gFBeX/ZNN8VJwnsNfgzj4me1HgVJdUo4W9kvx9cr2jHWkC3Oj/bdBTh1+yBjOC53yHlQK/
l1GHrEWiWPPnE434LRxnWkwr8EHD4oieJxC8fkIxkQfj+gHhU79Z
+oAAYAAAzsnf9SDIZPoDXF0TdC9POqTgld0oXDl2XPaVD4CvvLearrOSlFv+lsNbC4rgZn23MtIBM/7YQmJwmQ
+FXRup6Tkubg1hpz04J/09dxg8UiZmntHiUr1GfkTOFMYqRB+Aw=="
            },
            {
                "SigningAlg": "P256r1",
                "Value": "hDdkJw9L2uMCORjImjMHqzNR6nYYh6QKncSl5GthQNmHmGU8a
+SOqDXWwDNt3VSntpbTTQl7cMIusqweQo+JPXXWElbGh7eaxPGz4ZeF5yM2cqVNUrQr1lX/6lZ
+OLuycrFrLzzB9APi0NIMLqV/Rt7XJssHQs2RPcT1ul/2XVpa6ztULJeQi2JwhTb/k48wbh/EvafG/
ibrIBIx9v7/
dwGRAPKHq7Uwb9hHnhpa8qNOUtjeUdIwJNh9vCBFX9s22t4PdortoFxbXo9C149PDDD4wqUHJGYlCsVX/
Sqqjf7Aug3h5dwdYN6cDgsuuiOm0+aBcXBGpkh7OxVxlwXkIP
+11dt23TkrSUKd0B01sc9Mc/0yEBCzx5RutKBwsefzyOl4vQX3AHgV7oD/XV73THMgGiDxQ55CPaaxN/
```

```
pm791VkQ76BSZaBeF+Su6tg0k/
eQneklt8Du5uqkyBHVxy8MvxsBIMZ73vIFwUrLHjDeq3+nOOyQqSBMnrHKU2mAwN3zb2LolwjPkKNOh1+NNnv99L2pBcNCn
+BgewzYNdWrXyKkp4O3ZDa4f+5SVWvbY5eyDDXcohvz/
OcCtuRjAkzKBCvIjBDnCv1McjVdCO3+utizGntfhAo1RZstnOoRkgVF2WuMT9IrUmzYximuTXUmWtjyFSTqgNBZwHWUTlMm
csC4HPTKr3dazdvEkhwGAAAIFByCjSp/5WHc4AhsyjMvKCsZQiKgiI8ECwjfXBaSZdY4zYsRlO3FC428H1atrFChFCZT0Bq
+vAUJiP8XqiEdXeqf2mYMJ5ykoDpwkve/cUQfPpjzFQlQfvwjBwiJDANKkOKoNT3bUGz+/f/pyTE
+xMRdIUBZ1Bw=="
            }
        ],
        "DeviceProfileId": "0ff5b0c6-f149-4498-af34-21993acd52a7",
        "PrivateKeys": [
            {
                "SigningAlg": "Ed25519",

 "Value": "2c24d4572327f23b9bef38097137c29224a9e979081b3d90124ac9dfa477934e"
            },
            {
                "SigningAlg": "P256r1",

 "Value": "38d526f29cfaf142f596deca187bd809ef71bc13435eedc885b63bb825d63def"
            }
        ],

 "SidewalkManufacturingSn": "843764270F4BDAE3023918C89A3307AB3351EA761887A40A9DC4A5E46B6140D9",
        "Status": "PROVISIONED"
    },

    ...

}
```


後續步驟

暫時儲存 JSON 檔案 *wireless_device.json* 和 *device_profile.json*，因為您會在下一個步驟中使用這些檔案來佈建和註冊您的終端裝置以連線至硬體平台。如需詳細資訊，請參閱《Amazon Sidewalk 文件》中的佈建和註冊您的終端裝置。

為您的 Sidewalk 終端裝置新增目的地

使用 AWS IoT 規則來處理資料和裝置訊息，並將其路由至其他服務。您還可定義規則來處理從裝置接收的二進位訊息，並將訊息轉換為其他格式，讓其他服務可輕鬆加以使用。目的地會將您的 Sidewalk 終端裝置與規則建立關聯，而此規則會處理裝置資料以傳送至其他 AWS 服務。

如何建立和使用目的地

1. 建立目的地的 AWS IoT 規則 和 IAM 角色。AWS IoT 規則指定將處理裝置資料並將其路由以供其他 AWS 服務 和您應用程式使用的規則。IAM 角色授予存取規則的權限。
2. 使用 `CreateDestination` API 操作，建立您 Sidewalk 裝置的目的地。指定目的地名稱、規則名稱、角色名稱和任何選用的參數。API 會傳回目的地的唯一識別碼，您可於將終端裝置新增至適用於 Amazon Sidewalk 的 AWS IoT Core 時加以指定。

下列展示如何建立目的地，及目的地的 AWS IoT 規則和 IAM 角色。

主題

- 建立您 Sidewalk 裝置的目的地
- 為您的目的地建立 IAM 角色和 IoT 規則

建立您 Sidewalk 裝置的目的地

您可從使用目的地中樞或使用 `CreateDestination`，將目的地新增至您的適用於 Amazon Sidewalk 的 AWS IoT Core 帳戶。建立目的地時，請指定：

- 一個用於您 Sidewalk 終端裝置之目的地的唯一名稱。

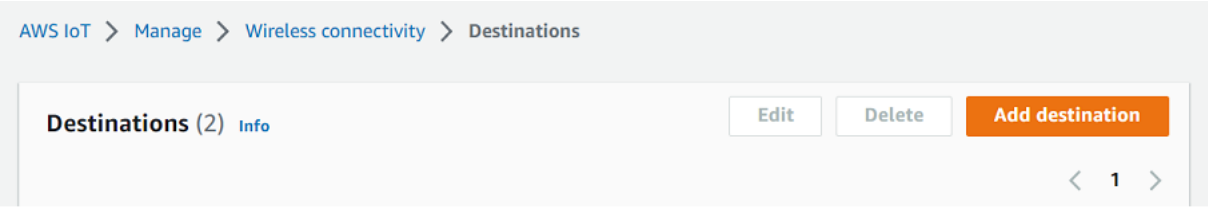
> **Note**
>
> 若您已使用目的地名稱新增裝置，則在建立目的地時必須使用該名稱。如需詳細資訊，請參閱步驟 2：新增您的 Sidewalk 裝置。

- 將要處理裝置資料的 AWS IoT 規則名稱，及訊息將要發佈的主題。
- 授予裝置資料存取規則權限的 IAM 角色。

下列章節說明如何建立您目的地的 AWS IoT 規則和 IAM 角色。

建立目的地 (主控台)

如要使用 AWS IoT 主控台建立目的地，請移至目的地中樞，並選擇新增目的地。



如要處理裝置的資料，請在建立目的地時指定下列欄位，然後選擇新增目的地。

- 目的地詳細資訊

  為您的目的地輸入 Destination name (目的地名稱) 和選用描述。

- 規則名稱

  AWS IoT 規則被設定為評估裝置傳送的訊息並處理裝置的資料。規則名稱會映射至目的地。目的地需要規則來處理接收到的訊息。您可以選擇透過叫用 AWS IoT 規則或發佈至 AWS IoT 訊息代理程式的方式來處理訊息。

  - 如果選擇 Enter a rule name (輸入規則名稱)，請輸入名稱然後選擇 Copy (複製)，來複製您在建立 AWS IoT 規則時將輸入的規則名稱。您可以選擇 Create rule (建立規則) 來立即建立規則，或導覽至 AWS IoT 主控台的規則中樞，並使用該名稱建立規則。

    您也可以輸入規則，再使用 Advanced (進階) 設定來指定主題名稱。主題名稱會在規則叫用期間提供，而且可以使用規則中的 `topic` 運算式存取。如需有關 AWS IoT 規則的更多資訊，請參閱 AWS IoT 規則。

  - 若您選擇發佈至 AWS IoT 訊息代理程式，請輸入主題名稱。然後您可以複製 MQTT 主題名稱，多位訂閱者可以訂閱此主題，以接收發佈至該主題的訊息。如需更多詳細資訊，請參閱 MQTT 主題。

  如需有關目的地之 AWS IoT 規則的詳細資訊，請參閱建立規則來處理 LoRaWAN 裝置訊息。

- 角色名稱

  授予裝置資料許可，以存取在 Rule name (規則名稱) 中命名之規則的 IAM 角色。您可以在主控台中建立新的服務角色，或選取現有的服務角色。如果正在建立新的服務角色，您可以輸入角色名稱 (例如 **`SidewalkDestinationRole`**)，或為 AWS IoT Core for LoRaWAN 保留為空白以產生新的角色名稱。AWS IoT Core for LoRaWAN 會代表您自動建立具有適當許可的 IAM 角色。

建立目的地 (CLI)

如要建立一個目的地，請使用 CreateDestination API 操作或 create-destination CLI 命令。例如，下列命令會建立一個 Sidewalk 終端裝置的目的地。

```
aws iotwireless create-destination --name SidewalkDestination \
    --expression-type RuleName --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

執行此命令會傳回目的地詳細資訊，此包括 Amazon Resource Name (ARN) 和目的地名稱。

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/SidewalkDestination",
    "Name": "SidewalkDestination"
}
```

如需有關建立目的地的詳細資訊，請參閱建立規則來處理 LoRaWAN 裝置訊息。

為您的目的地建立 IAM 角色和 IoT 規則

AWS IoT 規則會將裝置訊息傳送至其他服務。AWS IoT 規則還可處理從 Sidewalk 終端裝置接收的二進位訊息，以供其他服務使用。適用於 Amazon Sidewalk 的 AWS IoT Core 目的地會將無線裝置與規則建立關聯，而此規則會處理裝置訊息資料以傳送至其他服務。一旦適用於 Amazon Sidewalk 的 AWS IoT Core 收到裝置資料，該規則就會對裝置資料採取行動。對於將其資料傳送至相同服務的所有裝置，您可建立可供所有裝置共用的目的地。您還需建立 IAM 角色，授與將資料傳送至規則的權限。

為您的目的地建立 IAM 角色

建立 IAM 角色，將傳送資料至 AWS IoT 規則的許可授予適用於 Amazon Sidewalk 的 AWS IoT Core。如要建立該角色，請使用 CreateRole API 操作或 create-role CLI 命令。您可將該角色命名為 *SidewalkRole*。

```
aws iam create-role --role-name SidewalkRole \
    --assume-role-policy-document '{"Version": "2012-10-17","Statement":
 [{ "Effect": "Allow", "Principal": {"Service": "lambda.amazonaws.com"}, "Action":
 "sts:AssumeRole"}]}'
```

您也可使用 JSON 檔案來定義角色的信任政策。

```
aws iam create-role --role-name SidewalkRole \
```

```
    --assume-role-policy-document file://trust-policy.json
```

下列顯示 JSON 檔案的內容。

trust-policy.json 的內容

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

為您的目的地建立規則

請使用 AWS IoT Core API 操作 CreateTopicRule，或 AWS CLI 命令 create-topic-rule 來建立一個規則。您的目的地將使用主題規則，將從 Sidewalk 終端裝置接收的資料路由至其他 AWS 服務。例如，您可建立一個將訊息傳送至 Lambda 函數的規則動作。您可以定義 Lambda 函數，使其從您的裝置接收應用程式資料，並使用 base64 解碼承載資料，則其他應用程式可以使用該函數。

下列步驟顯示如何建立 Lambda 函數，接著建立將訊息傳送至此函數的主題規則。

1. 建立執行角色和原則

   建立一個 IAM 角色，授予您函數存取 AWS 資源的權限。您也可使用 JSON 檔案來定義角色的信任政策。

   ```
   aws iam create-role --role-name lambda-ex \
       --assume-role-policy-document file://lambda-trust-policy.json
   ```

   下列顯示 JSON 檔案的內容。

   lambda-trust-policy.json 的內容

   ```
   {
   ```

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

2. 建立並測試 Lambda 函數

   執行下列步驟來建立 base64 解碼承載資料的 AWS Lambda 函數。

   a. 撰寫用於解碼負載資料的程式碼。例如，您可使用下列 Python 程式碼範例。指定指令碼的名稱，例如 *base64_decode.py*。

      base64_decode.py 的內容

      ```
      // ---------------------------------------------------------
      // ----- Python script to decode incoming binary payload -----
      // ---------------------------------------------------------
      import json
      import base64

      def lambda_handler(event, context):

          message = json.dumps(event)
          print (message)

          payload_data = base64.b64decode(event["PayloadData"])
          print(payload_data)
          print(int(payload_data,16))
      ```

   b. 將部署套件建立為一個包含 Python 檔案的 zip 檔案，並將其命名為 *base64_decode.zip*。使用 `CreateFunction` API 或 `create-function` CLI 命令，為範例程式碼 *base64_decode.py* 建立 Lambda 函數。

   c.
      ```
      aws lambda create-function --function-name my-function \
      --zip-file fileb://base64_decode.zip --handler index.handler \
      --runtime python3.9 --role arn:aws:iam::123456789012:role/lambda-ex
      ```

您應該會看到下列輸出。建立主題規則時，您將會使用輸出 FunctionArn 中的 Amazon Resource Name (ARN) 值。

```
{
    "FunctionName": "my-function",
    "FunctionArn": "arn:aws:lambda:us-east-1:123456789012:function:my-
function",
    "Runtime": "python3.9",
    "Role": "arn:aws:iam::123456789012:role/lambda-ex",
    "Handler": "index.handler",
    "CodeSha256": "FpFMvUhayLkOoVBpNuNiIVML/tuGv2iJQ7t0yWVTU8c=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "88ebe1e1-bfdf-4dc3-84de-3017268fa1ff",
    ...
}
```

d. 如要從命令列取得某次調用的日誌，請使用具有 invoke 命令的 --log-type 選項。該回應包括 LogResult 欄位，其內含該次調用的 base64 編碼日誌 (最大達 4 KB)。

```
aws lambda invoke --function-name my-function out --log-type Tail
```

您應該會收到 StatusCode 為 200 的回應。如需有關從 AWS CLI 建立並使用 Lambda 函數的詳細資訊，請參閱 Lambda 與 AWS CLI 搭配使用。

3. 建立主題規則

使用 CreateTopicRule API 或 create-topic-rule CLI 命令，建立將訊息傳送至此 Lambda 函數的主題規則。您還可新增一個重新發佈至 AWS IoT 主題的第二個規則動作。將此主題規則命名為 *Sidewalkrule*。

```
aws iot create-topic-rule --rule-name Sidewalkrule \
    --topic-rule-payload file://myrule.json
```

您可使用 myrule.json 檔案，來指定有關規則的更多詳細資訊。例如，下列 JSON 檔案顯示如何重新發佈至 AWS IoT 主題，及如何將訊息傳送至 Lambda 函數。

```
{
```

```
    "sql": "SELECT * ",
    "actions": [
       {
            // You obtained this functionArn when creating the Lambda function
 using the
            // create-function command.
            "lambda": {
                "functionArn": "arn:aws:lambda:us-east-1:123456789012:function:my-
function"
             }
        },
        {
            // This topic can be used to observe messages exchanged between the
 device and
            // AWS IoT Core for Amazon Sidewalk after the device is connected.
             "republish": {
                 "roleArn": "arn:aws:iam::123456789012:role/service-
role/SidewalkRrepublishRole",
                 "topic": "project/sensor/observed"
             }
        }
    ],
}
```

連接您的 Sidewalk 裝置並檢視上行中繼資料格式

於本教學課程中，您將會使用 MQTT 測試用戶端來測試連線，並查看在您終端裝置和 AWS 雲端 之間的訊息交換。如要接收訊息，請於 MQTT 測試用戶端中訂閱在建立目的地的 IoT 規則時所指定的主題。您還可使用 SendDataToWirelessDevice API 操作，將下行訊息從適用於 Amazon Sidewalk 的 AWS IoT Core 傳送至您的裝置。您可啟用訊息傳遞狀態事件通知，確認訊息已傳遞。

> **Note**
>
> 如需連接硬體平台並進行設定的相關資訊，請參閱《Amazon Sidewalk 文件》中的佈建和註冊終端裝置和設定硬體開發套件 (HDK)。

將下行訊息傳送至您的終端設備

使用 SendDataToWirelessDevice API 操作或 send-data-to-wireless-device CLI 指令，將下行訊息從適用於 Amazon Sidewalk 的 AWS IoT Core 傳送至您的 Sidewalk 終端裝置。下列顯示如何執行此命令的範例。承載資料是要傳送的二進位，以 base64 編碼。

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

下列顯示執行此命令的範例輸出，這是傳送至裝置的下行訊息 ID。

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

> **Note**
>
> SendDataToWirelessDevice API 可以傳回訊息 ID，但該訊息可能無法成功傳遞。如要檢查傳送至裝置的訊息狀態，您可為 Sidewalk 帳戶和裝置啟用訊息傳遞狀態事件。如需如何啟用這些事件的詳細資訊，請參閱 Sidewalk 資源的事件通知。如需此事件類型的詳細資訊，請參閱訊息傳遞事件。

檢視從裝置傳送的上行訊息格式

在連接至您的裝置後，您可訂閱主題 (例如，*project/sensor/observed*)，此主題是您在建立目的地規則，並從裝置觀察上行訊息時所指定的。

若您在建立目的地時指定了主題名稱，則可訂閱從您終端裝置監控上行訊息的主題。請移至 AWS IoT 主控台之測試頁面上的 MQTT 測試用戶端，輸入主題名稱 (例如，*project/sensor/observed*)，接著選擇訂閱。

下列範例顯示從 Sidewalk 裝置傳送至 AWS IoT 之上行訊息的格式。WirelessMetadata 包含有關訊息請求的中繼資料。

```
{
```

```
    "PayloadData":"ZjRlNjY1ZWNlNw==",
    "WirelessDeviceId":"wireless_device_id",
    "WirelessMetadata":{
       "Sidewalk":{
          "CmdExStatus":"Cmd",
          "SidewalkId":"device_id",
          "Seq":0,
          "MessageType":"messageType"
       }
     }
}
```

下表顯示上行中繼資料中不同參數的定義。此 *device-id* 是無線裝置的 ID，如 *ABCDEF1234*，而 *messageType* 是從裝置所接收之上行訊息的類型。

Sidewalk 上行中繼資料參數

參數	描述	類型	必要
PayloadData	從無線裝置傳送之訊息承載。	字串	是
WirelessDeviceID	發送資料的無線裝置識別符	字串	是
Sidewalk.CmdExStatus	命令執行時間狀態。回應類型訊息應該包含狀態碼 COMMAND_EXEC_STATUS_SUCCESS 。不過，通知可能未包含狀態碼。	列舉	否
Sidewalk.NackExStatus	回應未確認狀態，可以是 RADIO_TX_ERROR 或 MEMORY_ERROR 。	字串陣列	否

使用適用於 Amazon Sidewalk 的 AWS IoT Core 大量佈建裝置

您可以使用大量佈建，一次將大量終端裝置加入適用於 Amazon Sidewalk 的 AWS IoT Core。大量佈建非常有用，尤其是當您在工廠中製造大量的裝置，並希望將這些裝置加入 AWS IoT 時。如需更多製造裝置的相關詳細資訊，請參閱《Amazon Sidewalk 文件》中的製造 Amazon Sidewalk 裝置。

下列主題會展示大量佈建的運作方式。

- Amazon Sidewalk 大量佈建工作流程

本主題會展示大量佈建及其運作方式的一些關鍵概念。此外還說明了必須執行的步驟，以便將您的 Sidewalk 裝置匯入適用於 Amazon Sidewalk 的 AWS IoT Core 中。

- 使用原廠支援建立裝置設定檔

  本主題說明如何建立裝置設定檔並取得其原廠支援。您還將會了解如何檢索 YubiHSM 金鑰，並將其傳送至您的製造商，以在製造裝置後取得控制日誌。

- 使用匯入任務佈建 Sidewalk 裝置

  本主題展示如何透過建立和使用匯入任務，大量佈建 Sidewalk 裝置。您還將會了解如何更新或刪除匯入任務，並檢視任務中匯入任務和裝置的狀態。

主題

Amazon Sidewalk 大量佈建工作流程

下列各節展示了大量佈建的重要概念及其運作方式。大量佈建所涉及的步驟包括：

1. 使用適用於 Amazon Sidewalk 的 AWS IoT Core 建立裝置設定檔。
2. 向 Amazon Sidewalk 團隊要求 YubiHSM 金鑰，並於原廠支援下更新您的裝置設定檔。
3. 將 YubiHSM 金鑰傳送給您的製造商，讓適用於 Amazon Sidewalk 的 AWS IoT Core 可於裝置製造完成後取得控制日誌。
4. 建立匯入任務，並提供要加入適用於 Amazon Sidewalk 的 AWS IoT Core 之裝置的序號 (SMSN)。

大量佈建的元件

下列概念對您展示了大量佈建的一些關鍵元件，及如何將其用來作為批量佈建 Sidewalk 裝置的一部分。

YubiHSM 金鑰

Amazon 會為您的每個 Sidewalk 產品建立一或多個 HSM (硬體安全模組)。每個 HSM 都有一個唯一的序號，稱為 YubiHSM 金鑰，此列印在硬體模組上。此金鑰可從 Yubico 網頁上購買。

該金鑰對每個 HSM 都是唯一的，且與您使用適用於 Amazon Sidewalk 的 AWS IoT Core 建立的每個裝置設定檔繫結起來。如要取得 YubiHSM 金鑰，請聯絡 Amazon Sidewalk 團隊。若您將 YubiHSM 金鑰傳送給製造商，則在原廠製造 Sidewalk 裝置後，適用於 Amazon Sidewalk 的 AWS IoT Core 將會收到包含裝置序號的控制日誌檔案。接著，其會將此資訊與您的輸入 CSV 檔案進行比較，以將裝置加入 AWS IoT。

裝置證明金鑰 (**DAK**)

當 Sidewalk 終端裝置加入 Sidewalk 網路時，必須使用 Sidewalk 裝置憑證佈建該裝置。用於設定裝置的憑證包括一個私有裝置特定憑證，及與 Sidewalk 憑證鏈相對應的公有裝置憑證。製造 Sidewalk 裝置時，YubiHSM 會簽署裝置憑證。

下列會展示範例 JSON 檔案，其中包含裝置憑證和私有金鑰。如需詳細資訊，請參閱取得裝置 JSON 檔案，進行佈建。

```
{
  "p256R1": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbH ... DANKkOKoNT3bUGz+/f/pyTE
+xMRdIUBZ1Bw==",
  "eD25519": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbHD ... UiZmntHiUr1GfkTOFMYqRB+Aw==",
  "metadata": {
    "devicetypeid": "fe98",

    ...

    "devicePrivKeyP256R1":
 "3e704bf8d319b3a475179f1d68c60737b28c708f845d0198f2d00d00c88ee018",
    "devicePrivKeyEd25519":
 "17dacb3a46ad9a42d5c520ca5f47f0167f59ce54d740aa13918465faf533b8d0"
  },
  "applicationServerPublicKey":
 "5ce29b89c2e3ce6183b41e75fe54e45f61b8bb320efbdd2abd7aefa5957a316b"
}
```

裝置證明金鑰 (DAK) 是您在建立裝置設定檔時取得的一個私有金鑰。此對應於產品憑證，該憑證是核發給每個 Sidewalk 產品的唯一憑證。當您聯絡 Amazon Sidewalk 團隊時，將會收到 Sidewalk 憑證鏈、YubiHSM 金鑰，及使用產品裝置證明金鑰 (DAK) 佈建的 HSM。

您的裝置設定檔也會更新為新的裝置證明金鑰 (DAK)，並啟用原廠支援。裝置設定檔的 DAK 中繼資料資訊會提供詳細資料，例如 DAK 名稱、憑證 ID、APID (公告產品 ID)、是否啟用原廠支援，及 DAK 可簽署的最大簽章數目。

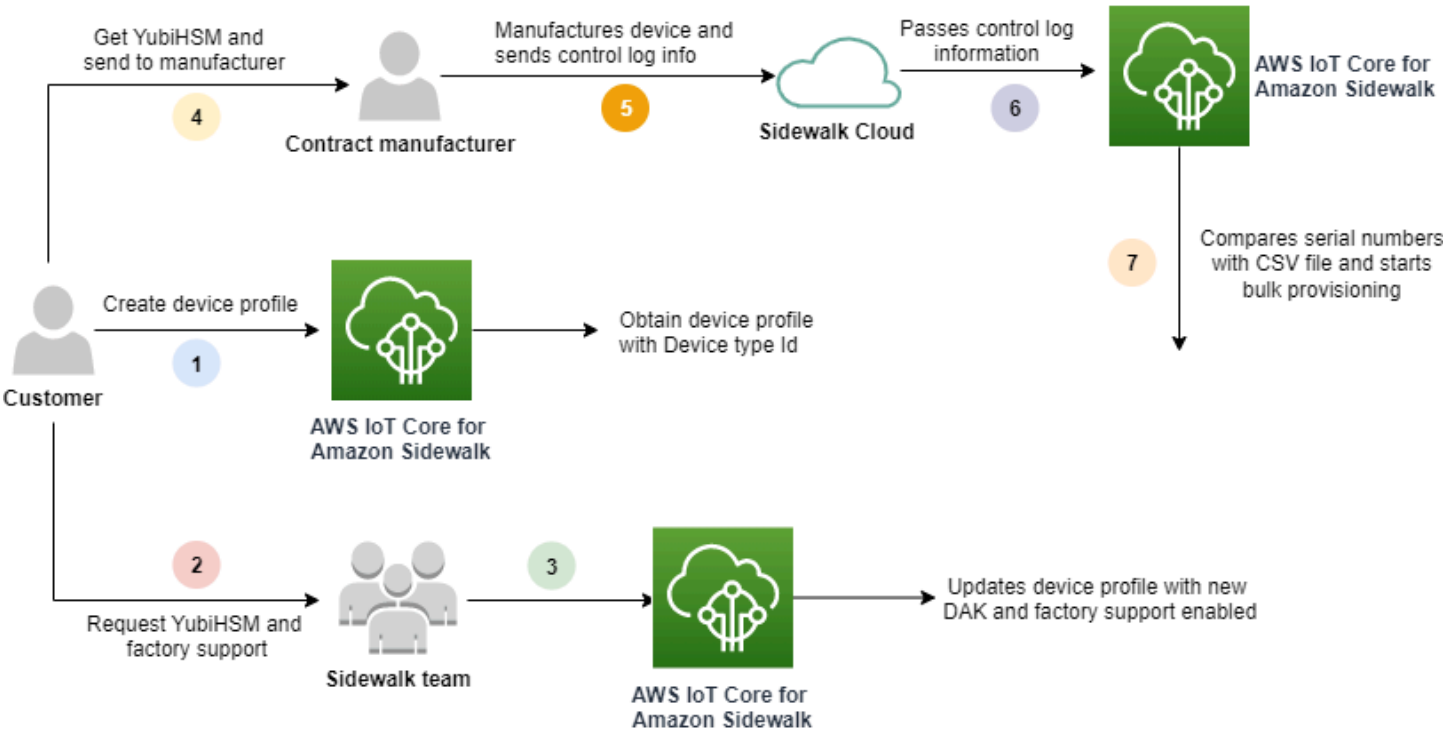
公告產品 ID (**ApId**)

ApId 參數是一個英數字元字串，用來識別公告的產品。當您想要為大量佈建的 Sidewalk 裝置使用給定的裝置設定檔時，必須指定此欄位。適用於 Amazon Sidewalk 的 AWS IoT Core 接著會產生 DAK，並透過 YubiHSM 金鑰將其提供給您。相關的 DAK 資訊將顯示於裝置設定檔中。

如要取得 ApId，於擷取您所建立之裝置設定檔的相關資訊後，請聯絡 Amazon Sidewalk 支援團隊。您可從 AWS IoT 主控台、或使用 GetDeviceProfile API 操作，或 get-device-profile CLI 命令取得裝置設定檔資訊。

大量佈建的運作方式

此流程圖展示大量佈建如何搭配適用於 Amazon Sidewalk 的 AWS IoT Core 運作。



下列程序說明大量佈建程序中的不同步驟。

1. 建立 Sidewalk 裝置的裝置設定檔

   將終端裝置帶至原廠之前，請先建立裝置設定檔。您可以使用此設定檔來佈建個別的裝置，如 新增您的裝置設定檔和 Sidewalk 終端裝置 中所述。

2. 為您的設定檔要求原廠支援

   當您備妥，將終端裝置帶至原廠時，請向 Amazon Sidewalk 團隊要求 YubiHSM 金鑰，並為您的裝置設定檔要求原廠支援。

3. 取得 DAK 和原廠支援的設定檔

   然後，Amazon Sidewalk 支援團隊將會使用產品裝置證明金鑰 (DAK) 和原廠支援來更新您的裝置設定檔。您的裝置設定檔將會自動更新為已公告產品 ID (APID)，及新的 DAK 和憑證資訊，例如憑證 ID。使用此設定檔的 Sidewalk 裝置符合大量佈建使用的資格。

4. 將 YubiHSM 金鑰傳送給製造商 (CM)

   您的終端裝置現在符合資格，因此您可將 YubiHSM 金鑰傳送給合約製造商 (CM)，開始製造程序。如需詳細資訊，請參閱《Amazon Sidewalk 文件》中的製造 Amazon Sidewalk 裝置。

5. 製造裝置並傳送控制日誌和序號

   CM 製造裝置並產生控制日誌。CM 還提供您 CSV 檔案，其中包含要製造的裝置清單及其 Sidewalk 製造序號 (SMSN)。下列程式碼顯示了範例控制日誌。其包含裝置的序號、APID 和公有裝置憑證。

```
{
    "controlLogs": [
    {
        "version": "4-0-1",
        "device":
        {
            "serialNumber": "device1",
            "productIdentifier": {
                "advertisedProductId": "abCD"
             },
             "sidewalkData": {
                "SidewalkED25519CertificateChain": "...",
                "SidewalkP256R1CertificateChain": "..."
             }
         }
      }
   ]
}
```

6. 將控制日誌資訊傳遞給適用於 Amazon Sidewalk 的 AWS IoT Core

   Amazon Sidewalk 雲端會從製造商擷取控制日誌資訊，並將此資訊傳遞給適用於 Amazon Sidewalk 的 AWS IoT Core。接著便可建立裝置及其序號。

7. 檢查序號是否符合，並開始大量佈建

   使用 AWS IoT 主控台或適用於 Amazon Sidewalk 的 AWS IoT Core API 操作 `StartWirelessDeviceImportTask` 時，適用於 Amazon Sidewalk 的 AWS IoT Core 將會對從 Amazon Sidewalk 所取得每個裝置之 Sidewalk 製造序號 (SMSN) 與 CSV 檔案中的對應序號進行比較。若此資訊相符，其會啟動大量佈建程序，並建立要匯入適用於 Amazon Sidewalk 的 AWS IoT Core 的裝置。

使用原廠支援建立裝置設定檔

在大量佈建您的 Amazon Sidewalk 裝置之前，您必須建立一個裝置設定檔，才可連絡 Amazon Sidewalk 支援團隊要求原廠支援。接著，Amazon Sidewalk 支援團隊將會使用新的裝置證明金鑰 (DAK) 更新您的裝置設定檔，並對其新增原廠支援。使用此設定檔的 Sidewalk 裝置隨後便符合資格，可與適用於 Amazon Sidewalk 的 AWS IoT Core 搭配使用，並可加入以進行大量佈建。

下列步驟會展示如何建立原廠支援的裝置設定檔。

1. 建立裝置設定檔

   首先建立一個裝置設定檔。建立設定檔時，請將名稱和選用標籤指定為名稱/值組。如需所需參數、及建立和使用設定檔的詳細資訊，請參閱 如何建立和新增您的裝置。

2. 取得設定檔的原廠支援

   接著取得您裝置設定檔的原廠支援，則使用此設定檔的裝置便可符合資格。若要取得資格，請與 Amazon Sidewalk 團隊一起建立票證。團隊確認後，您將會收到一個 APID (公告產品 ID)，而您的設定檔將以原廠核發的 DAK 進行更新。使用此設定檔的 Sidewalk 終端裝置將會符合資格。

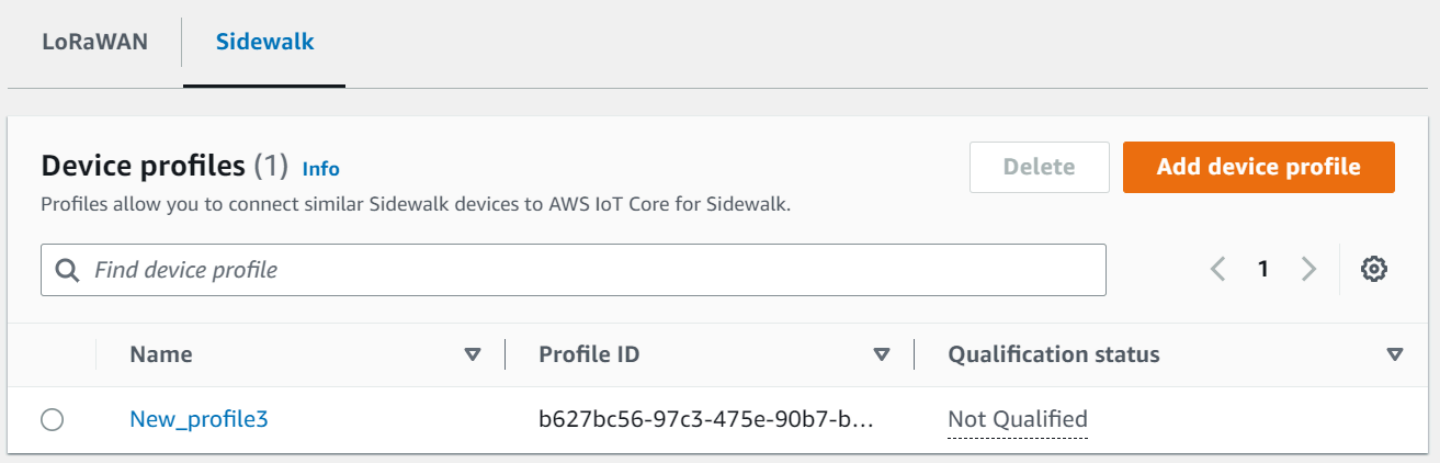
您可使用 AWS IoT 主控台、適用於 Amazon Sidewalk 的 AWS IoT Core API 操作或 AWS CLI 來建立裝置設定檔。

主題

- 建立設定檔 (主控台)
- 建立一個設定檔 (CLI)
- 後續步驟

建立設定檔 (主控台)

如要使用 AWS IoT 主控台建立裝置設定檔，請移至設定檔中樞的 Sidewalk 索引標籤，然後選擇建立設定檔。



如要建立設定檔，請指定下列欄位，然後選擇提交。

- 名稱

  輸入您設定檔的名稱。

- 標籤

  輸入選用索引標籤作為名稱/值組，協助您更輕鬆地識別您的設定檔。索引標籤也可讓您更輕鬆地追蹤帳單費用。

檢視設定檔資訊並限定設定檔

您會看到您建立於設定檔中樞的設定檔。選擇設定檔以檢視其詳細資訊。您會看到下列相關資訊：

- 裝置設定檔名稱和唯一識別碼，及您指定為名稱/值組的任何選用索引標籤。
- 設定檔的應用程式伺服器公有金鑰和裝置類型 ID。
- 資格狀態，此表示您使用的裝置設定檔不受原廠支援。如要限定您的裝置設定檔以使其獲得原廠支援，請聯絡 Amazon Sidewalk 支援。
- 裝置證明金鑰 (DAK) 資訊。一旦您的裝置設定檔符合資格，便會核發一個新的 DAK，您的設定檔會自動更新為新的 DAK 資訊。

建立一個設定檔 (CLI)

如要建立一個裝置設定檔，請使用 CreateDeviceProfile API 操作或 create-device-profile CLI 命令。例如，下列命令會建立 Sidewalk 終端裝置的設定檔。

```
aws iotwireless create-device-profile \
    --name sidewalk_device_profile --sidewalk {}
```

執行此命令會傳回設定檔詳細資訊，此包括 Amazon Resource Name (ARN) 和設定檔的 ID。

```
{
    "DeviceProfileArn": "arn:aws:iotwireless:us-
east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

檢視設定檔資訊並限定設定檔

使用 GetDeviceProfile API 操作或 get-device-profile CLI 命令，以取得您為適用於 Amazon Sidewalk 的 AWS IoT Core 新增至帳戶之裝置設定檔的相關資訊。如要擷取裝置設定檔的相關資訊，請指定設定檔 ID。接著，API 將會傳回與所指定識別符相符之裝置設定檔的資訊。

下列顯示範例 CLI 命令：

```
aws iotwireless get-device-profile \
    --id "12345678-234a-45bc-67de-e8901234f0a1" > device_profile.json
```

執行此命令會傳回裝置設定檔的參數、應用程式伺服器公有金鑰 DeviceTypeId、ApId、資格狀態，及 DAKCertificate 資訊。

於此範例中，資格狀態和 DAK 資訊表示您的裝置設定檔不合格。如要符合您的設定檔資格，請聯絡 Amazon Sidewalk 支援，您的設定檔將會獲一個新的 DAK，沒有裝置限制。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Name": "Sidewalk_profile",
    "LoRaWAN": null,
    "Sidewalk":
    {
```

```
        "ApplicationServerPublicKey":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234",
        "DAKCertificateMetadata": [
            {
                "DeviceTypeId": "fe98",
                "CertificateId": "43564A6D2D50524F544F54595045",
                "FactorySupport": false,
                "MaxAllowedSignature": 1000
            }
        ],
        "QualificationStatus": false
    }
}
```

Amazon Sidewalk 支援團隊確認此資訊後，您將會收到 APID 和原廠支援的 DAK，如下列範例所示。

> **Note**
>
> MaxAllowedSignature 為 -1 表示 DAK 沒有任何裝置限制。如需 DAK 參數的詳細資訊，請參閱 DAKCertificateMetadata。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Name": "Sidewalk_profile",
    "LoRaWAN": null,
    "Sidewalk":
    {
        "ApplicationServerPublicKey":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234",
        "DAKCertificateMetadata": [
            {
                "ApId": "GZBd",
                "CertificateId": "43564A6D2D50524F544F54595045",
                "FactorySupport": true,
                "MaxAllowedSignature": -1
            }
        ],
        "QualificationStatus": true
    }
```

```
}
```

後續步驟

現在您已經建立了具有原廠支援的 DAK 的裝置設定檔，請將您從團隊取得的 YubiHSM 金鑰提供給製造商。然後，您的裝置將於工廠中製造，然後將控制日誌資訊傳送至 Amazon Sidewalk，其包含裝置的序號 (SMSN)。如需此工作流程的相關詳細資訊，請參閱《Amazon Sidewalk 文件》中的製造 Amazon Sidewalk 裝置。

接著，您可將要加入裝置的序號提供給適用於 Amazon Sidewalk 的 AWS IoT Core，藉此大量佈建 Sidewalk 裝置。當適用於 Amazon Sidewalk 的 AWS IoT Core 收到控制日誌時，其會將控制日誌中的序號與您提供的序號進行比較。若序號相符，匯入任務會開始將您的裝置加入適用於 Amazon Sidewalk 的 AWS IoT Core。如需詳細資訊，請參閱使用匯入任務佈建 Sidewalk 裝置。

使用匯入任務佈建 Sidewalk 裝置

本節將說明如何使用 AWS IoT 主控台、適用於 Amazon Sidewalk 的 AWS IoT Core API 操作，或 AWS CLI 大量佈建 Sidewalk 裝置。下列各節說明如何大量佈建您的 Sidewalk 裝置。

主題

Sidewalk 大量佈建的運作方式

下列步驟說明大量佈建的運作方式。

1. 啟動無線裝置匯入任務

   如要大量佈建 Sidewalk 裝置，您必須建立匯入任務，並將要加入裝置的 Sidewalk 製造序號 (SMSN) 提供給適用於 Amazon Sidewalk 的 AWS IoT Core。製造商將控制日誌上傳至 Amazon Sidewalk 後，您在電子郵件中以 CSV 檔案的形式取得裝置的 Sidewalk 製造序號 (SMSN)。如需此工作流程及取得控制日誌方法的相關詳細資訊，請參閱《Amazon Sidewalk 文件》中的製造 Amazon Sidewalk 裝置。

2. 在背景中執行匯入程序

   當適用於 Amazon Sidewalk 的 AWS IoT Core 收到匯入任務請求時，便會開始進行設定，並啟動經常輪詢系統的背景處理程序。一旦背景處理程序收到匯入任務指示，便會開始讀取 CSV 檔案。適用於 Amazon Sidewalk 的 AWS IoT Core 會同時檢查是否已從 Amazon Sidewalk 收到控制日誌。

3. 建立無線裝置記錄

   從 Amazon Sidewalk 收到控制日誌時，適用於 Amazon Sidewalk 的 AWS IoT Core 會檢查控制日誌中的序號是否與 CSV 檔案中的 SMSN 值相符。若序號相符，適用於 Amazon Sidewalk 的 AWS IoT Core 將會開始為與這些序號對應的 Sidewalk 裝置建立無線裝置記錄。加入所有裝置之後，匯入任務會標示為已完成。

Sidewalk 大量佈建的關鍵注意事項

將 Sidewalk 裝置大量佈建至適用於 Amazon Sidewalk 的 AWS IoT Core 時，以下是一些主要考量事項。

- 您必須在建立裝置設定檔的相同 AWS 帳戶 位置，使用 AWS IoT 主控台或適用於 Amazon Sidewalk 的 AWS IoT Core API 操作執行大量佈建。
- 在大量佈建 Sidewalk 裝置之前，您的裝置設定檔必須已包含表示原廠支援的 DAK 資訊。否則，使用 AWS IoT 主控台或大量佈建 API 操作的大量佈建可能會失敗。
- 開始匯入任務後，處理 CSV 檔案、匯入無線裝置並將其加入適用於 Amazon Sidewalk 的 AWS IoT Core 至少需要 10 分鐘或更長時間。
- 一旦啟動後，無線裝置匯入任務將執行 90 天。在此期間，其會檢查是否已從 Amazon Sidewalk 收到控制日誌。若 90 天之前並未收到 Amazon Sidewalk 的控制日誌，則在您檢視任務詳細資訊時，該任務將標記為已完成，並顯示一則訊息，指出該任務已過期。匯入任務中正在等待控制日誌的裝置加入狀態將標記為失敗。
- 當您嘗試更新已建立的匯入任務時，您只能將其他裝置新增至該任務。您可在建立匯入任務之後，及在已新增至匯入任務的裝置上開始任務之前，隨時新增裝置。若更新檔案包含原始匯入任務中已存在的裝置序號，則會忽略這些序號。
- 當您要求更新作業時，將假定與您在建立匯入任務時所使用角色相同的 IAM 角色來存取 Amazon S3 儲存貯體中的 CSV 檔案。
- 只有在任務已成功完成或任務更新失敗時，才可刪除匯入任務。在諸如提供了不正確的 IAM 角色或找不到 Amazon S3 儲存貯體檔案等狀況下，任務可能無法更新。若匯入任務處於 `PENDING` 狀態，則無法進行更新或刪除。

- 您匯入至任務的 CSV 檔案必須使用下個章節中所述的格式。

CSV 檔案格式

您為匯入任務所指定 Amazon S3 儲存貯體中包含的 CSV 檔案必須使用下列格式：

- 第 1 列必須使用關鍵字 `smsn`，此表示要匯入的 CSV 檔案包含要匯入之裝置的 SMSN。
- 第 2 列和之後的列必須包含要加入之裝置的 SMSN。該裝置 SMSN 必須為 64 十六進位字元格式。

此 JSON 檔案會顯示 CSV 檔案格式的範例。

```
smsn
1C1A10B0AC0A200C012BBAC2CBB1B21CB12C0CA2AC1C1BB22CAA01C1B0B01122
B122C2B1121BACA2221001AC1B22012AAC11112C11C2A100C1C2B012A1100C10
02B222C110B0A210B0A0C2C112CCCAC21C1C0B0AA1221AB1022A2CC11B1B1122
C2C021CA1C111CCAB1221C0021C1C2AAA0AA1A2A01ABC10CBAACCA2A0121022A
0CB22C01BBC2CA2C0B11001121ACB2ABB0BB0121C2BA101C012CC2B20C011AC0
```

Sidewalk 大量佈建的使用方式

下列步驟展示如何使用 Amazon Sidewalk 大量佈建。

1. 提供裝置序號

   如要佈建您的 Sidewalk 裝置，則您必須提供要加入裝置的序號。您可以使用下列任一種方法來佈建您的裝置。

   - 使用其 Sidewalk 製造序號 (SMSN) 個別佈建每個裝置。若您想要更快速地測試工作流程並加入您的裝置而無需上傳具有適當 IAM 角色的 CSV 檔案，或等待裝置準備好加入任務時，則此方法很有用。
   - 透過提供包含要在 CSV 檔案中所佈建裝置之 SMSN 的 Amazon S3 儲存貯體 URL，來大量佈建裝置。當您有大量要加入的裝置時，此種方法特別有用。於此狀況下，單獨加入每個裝置可能會很繁瑣。相反地，您只需要提供已上傳到 Amazon S3 儲存貯體之 CSV 檔案的路徑，和 IAM 角色即可存取檔案。
2. 取得匯入任務和裝置加入狀態

   對於您建立的每個匯入任務，您可擷取有關任務加入狀態和新增至該任務裝置之上線狀態的資訊。您還可查看其他狀態資訊，例如任務或裝置加入失敗的原因。如需詳細資訊，請參閱

3. (選用) 更新或刪除匯入任務

   您可以更新或刪除已建立的匯入任務。

   - 在已新增的裝置上開始匯入任務之前，您可隨時更新任務並將其他裝置新增至該任務。適用於 Amazon Sidewalk 的 AWS IoT Core 擔任的 IAM 角色與您建立匯入任務時使用的角色相同。當您建立任務時，請指定包含要新增至任務之裝置序號的新 CSV 檔案。

     > **Note**
     >
     > 當您更新現有的匯入任務時，您僅可將裝置新增至任務。適用於 Amazon Sidewalk 的 AWS IoT Core 會在匯入任務中已有的裝置與您嘗試新增至任務的裝置之間執行聯集操作。若新檔案包含匯入任務中已存在的裝置序號，則會忽略這些序號。

   - 您可刪除已順利完成的匯入任務，或在諸如 IAM 角色資訊不正確，或建立或更新任務時無法使用 S3 儲存貯體檔案等況下，更新失敗的匯入任務。

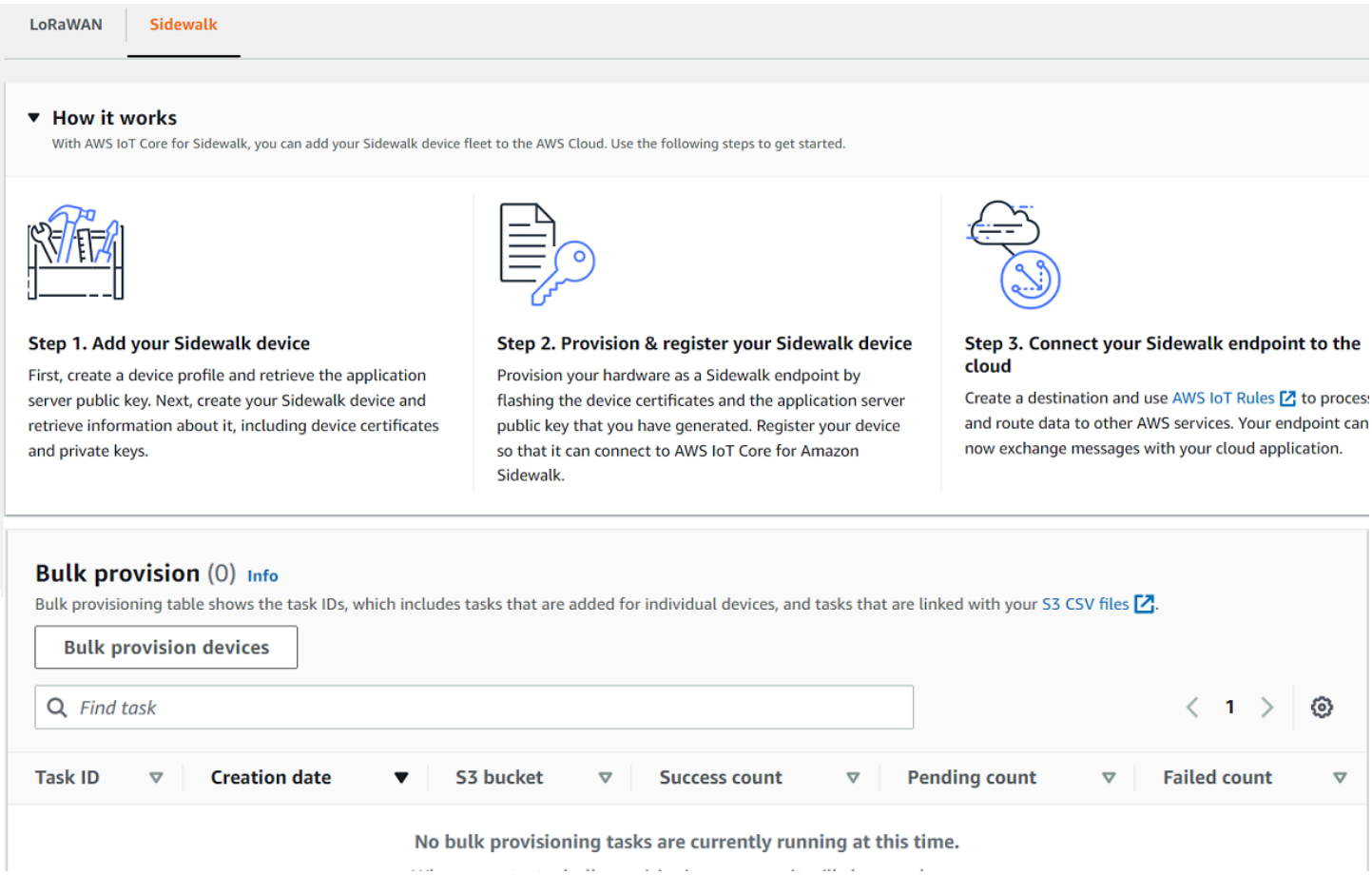
主題

大量佈建 Sidewalk 裝置

本節將說明如何使用 AWS IoT 主控台和 AWS CLI，對適用於 Amazon Sidewalk 的 AWS IoT Core 大量佈建 Sidewalk 裝置。

大量佈建 Sidewalk 裝置 (主控台)

如要使用 AWS IoT 主控台新增您的 Sidewalk 裝置，請移至裝置中樞的 Sidewalk 索引標籤，選擇大量佈建裝置，然後執行下列步驟。



1. 選擇匯入方式

   指定您要將大量加入的裝置匯入適用於 Amazon Sidewalk 的 AWS IoT Core 的方式。

   - 如要使用其 SMSN 佈建個別裝置，請選擇佈建個別的原廠支援裝置。
   - 如要透過提供包含裝置及其 SMS 清單的 CSV 檔案，大量佈建裝置，請選擇使用 S3 儲存貯體。

2. 指定要加入的裝置

   視您選擇將裝置加入的方法而定，新增裝置資訊及其序號。

   a. 若您選擇佈建個別的原廠支援裝置，請指定下列資訊：

      i. 每個要加入之裝置的名稱。該新名稱在您的 AWS 帳戶 和 AWS 區域 中必須是唯一的。

      ii. 其在輸入 SMSN欄位中的 Sidewalk 製造序號 (SMSN)。

      iii. 描述將訊息從裝置路由至其他 AWS 服務 之 IoT 規則的目的地。

   b. 若您選擇使用 S3 儲存貯體：

i. 提供包含 S3 URL 資訊的 S3 儲存貯體目的地資訊。如要提供 CSV 檔案，請選擇瀏覽 S3，然後選擇您想要使用的 CSV 檔案。

適用於 Amazon Sidewalk 的 AWS IoT Core 會自動填入 S3 URL，其為前往 S3 儲存貯體中 CSV 檔案的路徑。路徑的格式為 `s3://bucket_name/file_name`。若要在 Amazon Simple Storage Service 主控台中檢視檔案，請選擇 View (檢視)。

ii. 提供 S3 佈建角色，此可讓適用於 Amazon Sidewalk 的 AWS IoT Core 代表您存取 S3 儲存貯體中的 CSV 檔案。您可以建立新的服務角色或選擇現有角色。

如要建立新角色，您可提供一個角色名稱，或者保留空白，以自動產生隨機名稱。

iii. 提供一個描述將訊息從裝置路由至其他 AWS 服務 之 IoT 規則的目的地。

3. 啟動匯入任務

提供任何選用標籤作為名稱/值組，並選擇提交以開始無線裝置匯入任務。

大量佈建 Sidewalk 裝置 (CLI)

如要將 Sidewalk 裝置加入您的適用於 Amazon Sidewalk 的 AWS IoT Core 帳戶，請使用下列任何 API 操作，這將取決於您要個別新增裝置，還是提供包含於 S3 儲存貯體中的 CSV 檔案。

- 使用 S3 CSV 檔案大量上傳裝置

  如要透過在 S3 儲存貯體中提供 CSV 檔案以大量上傳裝置，請使用 `StartWirelessDeviceImportTask` API 操作或 `start-wireless-device-import-task` AWS CLI 命令。建立任務時，請指定 Amazon S3 儲存貯體中 CSV 檔案的路徑，並指定 IAM 角色，以便將存取 CSV 檔案的許可授予適用於 Amazon Sidewalk 的 AWS IoT Core。

  一旦任務開始執行後，適用於 Amazon Sidewalk 的 AWS IoT Core 將會開始讀取 CSV 檔案，並將檔案中的序號 (SMSN) 與從 Amazon Sidewalk 所收到之控制日誌中的對應資訊進行比較。若序號相符，其將會開始建立與這些序號對應的無線裝置記錄。

  下列命令顯示建立匯入任務的範例：

```
aws iotwireless start-wireless-device-import-task \
    --cli-input-json "file://task.json"
```

  下列顯示 `task.json` 檔案的內容。

task.json 的內容

```
{
    "DestinationName": "Sidewalk_Destination",
    "Sidewalk": {
        "DeviceCreationFile": "s3://import_task_bucket/import_file1",
        "Role": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI"
    }
}
```

執行此命令會傳回匯入任務的 ID 和 ARN。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"
    "Id": "a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"
}
```

- 使用其 SMSN 個別佈建裝置

  如要使用其 SMSN 個別佈建裝置，請使用 StartSingleWirelessDeviceImportTask API 操作或 start-single-wireless-device-import-task AWS CLI 命令。建立任務時，請指定 Sidewalk 目的地和您要加入的裝置序號。

  當序號與從 Amazon Sidewalk 所收到之控制日誌中的對應資訊相符時，會執行任務並建立無線裝置記錄。

  下列命令顯示建立匯入任務的範例：

```
aws iotwireless start-single-wireless-device-import-task \
    --destination-name sidewalk_destination \
    --sidewalk
 '{"SidewalkManufacturingSn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A
```

  執行此命令會傳回匯入任務的 ID 和 ARN。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
    "Id": "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

```
}
```

更新或刪除匯入任務

若您想要將其他裝置新增至匯入任務，您可以更新該任務。若您不再需要某任務或其已失敗，則還可將其刪除。如需有關何時更新或刪除任務的資訊，請參閱 Sidewalk 大量佈建的使用方式。

> **Warning**
>
> 刪除動作為永久性動作，且無法還原。刪除已順利完成的匯入任務將不會移除已使用該任務加入的終端裝置。

如要更新或刪除匯入任務：

- 使用 AWS IoT 主控台

  下列步驟說明如何使用 AWS IoT 主控台來更新或刪除匯入任務。

  如要更新匯入任務：

  1. 請移至 AWS IoT 主控台的 Sidewalk 裝置中樞。
  2. 選擇您要更新的匯入任務，然後選擇編輯。
  3. 提供另一個包含您要新增至任務之裝置序號的 S3 檔案，然後選擇提交。

  如要刪除匯入任務：

  1. 請移至 AWS IoT 主控台的 Sidewalk 裝置中樞。
  2. 選擇您要刪除的任務，然後選擇刪除。

- 使用 AWS IoT Wireless API 或 AWS CLI

  使用下列 AWS IoT Wireless API 操作或 CLI 命令來更新或刪除您的匯入任務。

  - **`UpdateWirelessDeviceImportTask`** API 或 **`update-wireless-device-import-task`** CLI

    此 API 操作會將 Amazon S3 CSV 檔案的內容附加至現有的匯入任務。您僅可新增先前未包含於任務中的裝置序號。

- **DeleteWirelessDeviceImportTask** API 或 **delete-wireless-device-import-task** CLI

  此 API 操作會使用匯入任務 ID 刪除標記為要刪除的匯入任務。

檢視匯入任務和裝置加入狀態

您的無線裝置匯入任務，而您已新增至該任務的 Sidewalk 裝置可能具有下列其中一種狀態訊息。您將會看到這些訊息顯示於 AWS IoT 主控台中，或是您使用任何 AWS IoT Wireless API 操作或 AWS CLI 命令擷取這些任務及其裝置相關資訊的時間。

檢視匯入任務狀態資訊

當您建立匯入任務後，則可檢視您所建立的匯入任務，以及新增至任務之裝置的加入狀態。加入狀態會指出等待加入的裝置數量、已成功加入的裝置數量，及無法加入的裝置數量。

當匯入任務剛剛建立時，待定計數將會顯示一個與新增裝置數量相對應的值。一旦任務啟動並讀取 CSV 檔案以建立無線裝置記錄後，待定計數就會減少，而成功計數也會隨著裝置的成功加入而增加。若有任何裝置無法加入，失敗計數將會增加。

如要檢視匯入任務和裝置加入狀態：

- 使用 AWS IoT 主控台

  於 AWS IoT 主控台的 Sidewalk 裝置中樞中，您可查看您所建立的匯入任務，及裝置加入狀態資訊摘要的計數。若您檢視所建立之任何匯入任務的詳細資訊，則可檢視裝置加入狀態的其他相關資訊。
- 使用 AWS IoT Wireless API 或 AWS CLI

  如要檢視裝置加入狀態，請使用下列任一項 AWS IoT Wireless API 操作或對應的 AWS CLI 命令。
  - **ListWirelessDeviceImportTasks** API 或 **list-wireless-device-import-tasks** CLI

    此 API 操作會傳回有關已新增至您 AWS IoT Wireless 帳戶的所有匯入任務及其狀態的資訊。其還會傳回這些任務中 Sidewalk 裝置加入狀態摘要的計數。
  - **ListDevicesForWirelessDeviceImportTask** API 或 **list-devices-for-wireless-device-import-task** CLI

    此 API 操作會傳回有關指定匯入任務及其狀態的資訊，以及已新增至匯入任務之所有 Sidewalk 裝置及其加入狀態資訊的相關資訊。
  - **GetWirelessDeviceImportTask** API 或 **get-wireless-device-import-task** CLI

此 API 操作會傳回有關指定匯入任務及其狀態的資訊，以及該任務中 Sidewalk 裝置加入狀態摘要的計數。

匯入任務狀態

您於AWS 帳戶 中所建立的匯入任務可能具有下列其中一種狀態訊息。該狀態會指出您的匯入任務是否已開始處理、或已完成或已失敗。您還可使用 AWS IoT 主控台或任何 AWS IoT Wireless API 操作的 `StatusReason` 參數，來擷取其他狀態詳細資訊。

- 初始化

  適用於 Amazon Sidewalk 的 AWS IoT Core 已收到無線裝置匯入任務請求，且正在設定任務。
- 已初始化

  適用於 Amazon Sidewalk 的 AWS IoT Core 已完成匯入任務的設定，並正等待控制日誌到達，以便使用裝置序號 (SMSN) 進行匯入並繼續處理任務。
- 待定

  匯入任務正在佇列中等待處理。適用於 Amazon Sidewalk 的 AWS IoT Core 正在評估處理佇列中的其他任務。
- 完成

  匯入任務已處理且完成。
- 失敗

  匯入任務或裝置任務失敗。您可使用 `StatusReason` 參數來識別匯入任務失敗的原因，例如驗證例外。
- 正在刪除

  匯入任務已標記為刪除，且正在刪除中。

裝置加入狀態

您新增至匯入任務的 Sidewalk 裝置可能具有下列其中一種狀態訊息。該狀態會指出您的裝置是否已準備好加入，或已加入或無法加入。您還可使用 AWS IoT 主控台或 AWS IoT Wireless API 操作 `ListDevicesForWirelessDeviceImportTask` 的 `OnboardingStatusReason` 參數 ，來擷取其他狀態的詳細資訊。

- 已初始化

  適用於 Amazon Sidewalk 的 AWS IoT Core 已完成匯入任務的設定，並正等待控制日誌到達，以便使用裝置序號 (SMSN) 進行匯入並繼續處理任務。

- 待定

  匯入任務正在佇列中等待處理，並開始將您的裝置加入該任務。適用於 Amazon Sidewalk 的 AWS IoT Core 正在評估處理佇列中的其他任務。

- 已加入

  已成功將 Sidewalk 裝置加入匯入任務。

- 失敗

  匯入任務或裝置任務失敗，且 Sidewalk 裝置無法加入任務。您可使用 `OnboardingStatusReason` 參數，擷取有關裝置加入失敗原因的其他詳細資料。

AWS IoT Wireless 中的安全性

雲端安全是 AWS 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足組織最為敏感的安全要求。

安全是 AWS 與您共同肩負的責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端本身的安全 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。第三方稽核人員會定期測試和驗證我們安全性的有效性，作為 AWS 合規計畫的一部分。若要了解適用於 AWS IoT Wireless 的合規計劃，請參閱合規計劃的 AWS 服務範圍。
- 雲端內部的安全 – 您的責任取決於所使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的請求和適用法律和法規。

本文件可幫助您了解如何在使用 AWS IoT Wireless 時套用共同責任模型。文件中將示範如何設定 AWS IoT Wireless 以符合您的安全和合規目標。您也會了解如何使用其他 AWS 服務來協助監控並保護 AWS IoT Wireless 資源。

目錄

AWS IoT Wireless 的資料保護

AWS 共同責任模型適用於 AWS IoT Wireless 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您負責維護在此基礎設施上託管內容的控制權。您也必須負責您所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的相關資訊，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，建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 憑證，並設定個人使用者。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 AWS CloudTrail 設定 API 和使用者活動日誌記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務 內的所有預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需 FIPS 和 FIPS 端點的相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用主控台、API、AWS CLI、AWS SDK 來使用 AWS IoT Wireless 或其他 AWS 服務 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。

AWS IoT Wireless 的資料加密

根據預設，所有傳輸中和靜態的 AWS IoT Wireless 資料都會加密。AWS IoT Wireless 不支援來自 AWS KMS key 的客戶受管 AWS KMS 金鑰。AWS IoT Wireless 只會使用 AWS 擁有的金鑰 來加密資料。

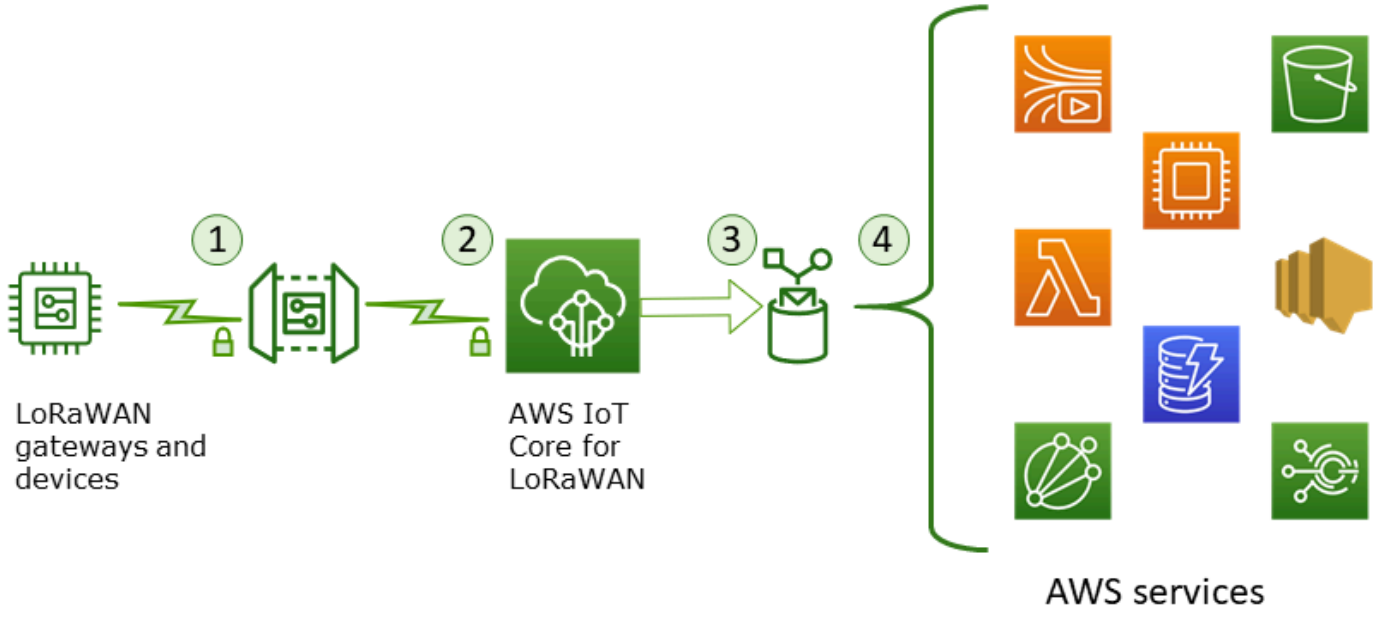
AWS IoT Core for LoRaWAN 的資料和傳輸安全

AWS IoT Core for LoRaWAN 使用以下方法來保護 LoRaWAN 裝置、閘道與 AWS IoT Core for LoRaWAN 之間的資料和通訊：

- 裝置與 LoRaWAN 閘道進行通訊時所遵循的安全最佳實務，如 LoRaWAN 安全白皮書所述。
- AWS IoT Core 用來將閘道連線到 AWS IoT Core for LoRaWAN，並將資料傳送到其他 AWS 服務的安全。如需詳細資訊，請參閱 AWS IoT Core 中的資料保護。

如何在整個系統中保護資料

此圖表顯示連線至 AWS IoT Core for LoRaWAN 的 LoRaWAN 系統中的重要元素，用來呈現資料的整體保護方式。



1. LoRaWAN 無線裝置會在傳輸二進位訊息之前，先使用 AES128 CTR 模式加密該訊息。
2. 閘道與 AWS IoT Core for LoRaWAN 的連線是由 TLS 保護，如 AWS IoT 中的傳輸安全性 所述。AWS IoT Core for LoRaWAN 會解密二進位訊息，並將解密的二進位訊息承載編碼為 base64 字串。
3. 產生的 base64 編碼訊息會當作訊息承載傳送至 AWS IoT 規則，而此規則是在指派給裝置的目的地中加以描述。AWS 內的資料是使用 AWS 擁有的金鑰進行加密。
4. AWS IoT 規則會將訊息資料導向至規則組態中所述的服務。AWS 內的資料是使用 AWS 擁有的金鑰進行加密。

LoRaWAN 裝置和閘道傳輸安全

LoRaWAN 裝置和 AWS IoT Core for LoRaWAN 會存放預先共用的根金鑰。工作階段金鑰是由 LoRaWAN 裝置和 AWS IoT Core for LoRaWAN 遵循通訊協定所衍生。對稱工作階段金鑰用於標準 AES-128 CTR 模式中的加密和解密。4 位元組訊息完整性代碼 (MIC) 也會用來遵循標準 AES-128 CMAC 演算法檢查資料完整性。工作階段金鑰可以使用聯結/重新聯結程序進行更新。

LoRa 閘道的安全實務會在 LoRaWAN 規格中加以描述。LoRa 閘道會使用 `Basics Station` 透過 Web 通訊端連線至 AWS IoT Core for LoRaWAN。AWS IoT Core for LoRaWAN 僅支援 `Basics Station` 2.0.4 版及更新版本。

建立 Web 通訊端連線之前，AWS IoT Core for LoRaWAN 會使用 TLS 伺服器和用戶端身分驗證模式來驗證閘道。為了確保 LoRaWAN 通訊協定的機密性，則會使用 TLS 1.2 版。多種程式設計語言與

作業系統提供 TLS 支援。AWS 內的資料是由特定的 AWS 服務加密。如需有關其他 AWS 服務上資料加密的詳細資訊，請參閱該服務的安全性文件。

AWS IoT Core for LoRaWAN 也會維護組態與更新伺服器 (CUPS)，用來設定並更新用於 TLS 身分驗證的憑證和金鑰。

AWS IoT Wireless 的身分和存取管理

AWS Identity and Access Management (IAM) 是一種 AWS 服務，讓管理員能夠安全地控制對 AWS 資源的存取權。IAM 管理員可以控制誰可以透過身分驗證 (已登入) 和授權 (具有許可) 來使用 AWS IoT Wireless 資源。IAM 是一種您可以免費使用的 AWS 服務。

主題

物件

您使用 AWS Identity and Access Management (IAM) 的方式取決於您在 AWS IoT Wireless 中所執行的工作。

服務使用者：如果您使用 AWS IoT Wireless 執行工作，您的管理員會為您提供所需的憑證和許可。隨著您為了執行工作而使用的 AWS IoT Wireless 功能數量變多，您可能會需要額外的許可。瞭解存取許可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 Wireless 中的某項功能，請參閱 對 AWS IoT Wireless 身分和存取進行故障診斷。

服務管理員：若您在公司負責管理 AWS IoT Wireless 資源，您應該擁有 AWS IoT Wireless 的完整存取權。您的工作是判斷服務使用者應存取的 AWS IoT Wireless 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可搭配 AWS IoT Wireless 使用 IAM 的方式，請參閱 AWS IoT Wireless 如何與 IAM 搭配使用。

IAM 管理員：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 AWS IoT Wireless 存取權的詳細資訊。若要檢視您可以在 IAM 中使用的範例 AWS IoT Wireless 身分型政策，請參閱 AWS IoT Wireless 身分型政策範例。

使用身分驗證

身分驗證是使用身分憑證登入 AWS 的方式。您必須以 AWS 帳戶根使用者、IAM 使用者身分，或擔任 IAM 角色進行 驗證 (登入至 AWS)。

您可以使用透過身分來源 AWS IAM Identity Center 提供的憑證，以聯合身分登入 AWS。(IAM Identity Center) 使用者、貴公司的單一登入身分驗證和您的 Google 或 Facebook 憑證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。您 AWS 藉由使用聯合進行存取時，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。如需登入至 AWS 的相關資訊，請參閱《AWS 登入 使用者指南》中的如何登入您的 AWS 帳戶。

如果您是以程式設計的方式存取 AWS，AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以便使用您的憑證透過密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，您必須自行簽署請求。如需使用建議的方法自行簽署請求的相關資訊，請參閱《IAM 使用者指南》中的簽署 AWS API 請求。

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

AWS 帳戶 根使用者

如果是建立 AWS 帳戶，您會先有一個登入身分，可以完整存取帳戶中所有 AWS 服務 與資源。此身分稱為 AWS 帳戶 根使用者，使用建立帳戶時所使用的電子郵件地址和密碼即可登入並存取。強烈建議您不要以根使用者處理日常作業。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。

IAM 使用者和群組

IAM 使用者是您 AWS 帳戶 中的一種身分，具備單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰)的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需詳細資訊，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。您無法以群組身分登入。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過程變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。若要進一步了解，請參閱《IAM 使用者指南》中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

> **Note**
>
> AWS IoT Wireless 不支援服務角色和服務連結角色。

IAM 角色是您 AWS 帳戶 中的一種身分，具備特定許可。它類似 IAM 使用者，但不與特定的人員相關聯。您可以在 AWS Management Console 中透過切換角色來暫時取得 IAM 角色。您可以透過呼叫 AWS CLI 或 AWS API 操作，或是使用自訂 URL 來取得角色。如需使用角色的方法的相關資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 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，並取得由角色定義的許可。如需有關聯合角色的詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-idp.html中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱《AWS IAM Identity Center 使用者指南》中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳户存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人)存取您帳戶中的資源。角色是授予跨帳户存取權的主要方式。但是，針對某些 AWS 服務，您可以將政策直接連接到資源 (而非使用角色作為代理)。若要了解跨帳戶存取角色和資源型政策間的差異，請參閱《IAM 使用者指南》中的 IAM 角色與資源類型政策的差異。
- 跨服務存取 – 有些 AWS 服務 會使用其他 AWS 服務 中的功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。

- 轉發存取工作階段 (FAS)：當您使用 IAM 使用者或角色在 AWS 中執行動作時，系統會將您視為主體。當您使用某些服務時，您可能會執行一個動作，而該動作之後會在不同的服務中啟動另一個動作。FAS 使用主體的許可呼叫 AWS 服務，搭配請求 AWS 服務 以向下游服務發出請求。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之後，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給 AWS 服務 服務。
- 服務連結角色 – 服務連結角色是一種連結到 AWS 服務 的服務角色類型。服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的 AWS 帳戶 中，並由該服務所擁有。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 – 針對在 EC2 執行個體上執行並提出 AWS CLI 和 AWS API 請求的應用程式，您可以使用 IAM 角色來管理暫時憑證。這是在 EC2 執行個體內儲存存取金鑰的較好方式。如需指派 AWS 角色給 EC2 執行個體並提供其所有應用程式使用，您可以建立連接到執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

如需了解是否要使用 IAM 角色或 IAM 使用者，請參閱《IAM 使用者指南》中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透過建立政策並將其附加到 AWS 身分或資源，在 AWS 中控制存取。政策是 AWS 中的一個物件，當其和身分或資源建立關聯時，便可定義其許可。AWS 會在主體 (使用者、根使用者或角色工作階段)發出請求時評估這些政策。政策中的許可決定是否允許或拒絕請求。大部分政策以 JSON 文件形式儲存在 AWS 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哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授與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。具備該政策的使用者便可以從 AWS Management Console、AWS CLI 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色)的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策則是獨立的政策，您可以將這些政策附加到 AWS 帳戶 中的多個使用者、群組和角色。受管政策包含 AWS 管理政策和客戶管理政策。如需瞭解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

資源型政策

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

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

存取控制清單 (ACL)

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

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

其他政策類型

AWS 支援其他較少見的政策類型。這些政策類型可設定較常見政策類型授與您的最大許可。

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

策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱《IAM 使用者指南》中的 IAM 實體許可範圍。

- 服務控制政策 (SCP) – SCP 是 JSON 政策，可指定 AWS Organizations 中組織或組織單位 (OU) 的最大許可。AWS Organizations 服務可用來分組和集中管理您企業所擁有的多個 AWS 帳戶。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 AWS 帳戶根使用者。如需組織和 SCP 的更多相關資訊，請參閱《AWS Organizations 使用者指南》中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱《IAM 使用者指南》中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。如需瞭解 AWS 在涉及多種政策類型時如何判斷是否允許一項請求，請參閱 IAM 使用者指南中的政策評估邏輯。

AWS IoT Wireless 如何與 IAM 搭配使用

使用 IAM 來管理 AWS IoT Wireless 的存取權之前，您應該先了解可與 AWS IoT Wireless 搭配使用的 IAM 功能有哪些。若要全面了解 AWS IoT Wireless 和其他 AWS 服務如何與 IAM 搭配使用，請參閱《IAM 使用者指南》中的使用 IAM 的 AWS 服務。

您可搭配 AWS IoT Wireless 使用的 IAM 功能

IAM 功能	AWS IoT Wireless 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵	是
ACL	否

IAM 功能	AWS IoT Wireless 支援
ABAC (政策中的標籤)	是
臨時憑證	是
主體許可	是
服務角色	否
服務連結角色	否

主題

- AWS IoT Wireless 身分型政策
- AWS IoT Wireless 內的資源型政策
- 政策動作
- 政策資源
- 條件索引鍵
- 存取控制清單 (ACL)
- ABAC 與 AWS IoT Wireless
- 將臨時憑證與 AWS IoT Wireless 搭配使用
- AWS IoT Wireless 的跨服務主體權限
- 服務角色
- AWS IoT Wireless 的服務連結角色

AWS IoT Wireless 身分型政策

支援身分型政策	是

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要瞭解如何建立身分類型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至附加的使用者或角色。如要瞭解您在 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。

範例

若要檢視 AWS IoT Wireless 身分型政策範例，請參閱 AWS IoT Wireless 身分型政策範例。

AWS IoT Wireless 內的資源型政策

支援以資源基礎的政策	否

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

若要啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源在不同的 AWS 帳戶 中時，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 存取資源的許可。其透過將身分型政策附加到實體來授予許可。不過，如果資源型政策會為相同帳戶中的主體授與存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM 角色與資源型政策有何差異。

政策動作

支援政策動作	是

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

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作的名稱通常會和相關聯的 AWS API 操作相同。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些操作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與執行相關聯操作的許可。

AWS IoT Wireless 中的政策動作會在動作之前使用以下字首：iotwireless:。例如，若要授與某人使用 ListWirelessDevices API 操作列出 AWS 帳戶 中註冊的所有無線裝置的許可，請將 iotwireless:ListWirelessDevices 動作包含在其政策中。政策陳述式必須包含 Action 或 NotAction 元素。AWS IoT Wireless 會定義自己的一組動作來描述您可以使用此服務執行的任務。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示：

```
"Action": [
      "iotwireless:ListMulticastGroups",
      "iotwireless:ListFuotaTasks"
   ]
```

您也可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 Get 文字的所有動作，請包含以下動作：

```
"Action": "iotwireless:Get*"
```

若要查看 AWS IoT Wireless 動作的清單，請參閱《IAM 使用者指南》中的 AWS IoT Wireless 定義的動作。

政策資源

支援政策資源	是

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

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

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

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

AWS IoT Wireless 服務具有以下 ARN：

```
arn:${Partition}:iotwireless:${Region}:${Account}:${Resource}/${Resource-id}
```

如需 ARN 格式的詳細資訊，請參閱 Amazon Resource Name (ARN) 和 AWS 服務命名空間。

例如，若要在您的陳述式中指定網路分析器組態 NAConfig1，請使用以下 ARN：

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:NetworkAnalyzerConfiguration/
NAConfig1"
```

若要指定屬於特定帳戶的所有 FUOTA 任務，請使用萬用字元 (*)：

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:FuotaTask/*"
```

有些 AWS IoT Wireless 動作無法對特定資源執行，例如用來列出資源的動作。在這些情況下，您必須使用萬用字元 (*)。

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

許多 AWS IoT Wireless API 動作都涉及多個資源。例如，AssociateWirelessDeviceWithThing 會將無線裝置與 AWS IoT 物件建立關聯，因此 IAM 使用者必須具有使用裝置和 IoT 物件的許可。若要在單一陳述式中指定多項資源，請使用逗號分隔 ARN。

```
"Resource": [
      "WirelessDevice",
      "thing"
```

若要查看 AWS IoT Wireless 資源類型及其 ARN 的清單，請參閱《IAM 使用者指南》中的 AWS IoT Wireless 定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 AWS IoT Wireless 定義的動作。

條件索引鍵

支援服務特定政策條件索引鍵	是

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

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

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。若您為單一條件索引鍵指定多個值，AWS 會使用邏輯 OR 操作評估條件。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與該 IAM 使用者。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件索引鍵和服務特定的條件索引鍵。若要查看 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。

AWS IoT Wireless 會定義自己的一組條件索引鍵，也支援使用某些全域條件索引鍵。若要查看 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的 AWS 全域條件內容金鑰。若要查看 AWS IoT Wireless 條件索引鍵的清單，請參閱《IAM 使用者指南》中的 AWS IoT Wireless 的條件索引鍵。若要了解您可以搭配哪些動作和資源使用條件索引鍵，請參閱 AWS IoT Wireless 定義的動作。

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。

ABAC 與 AWS IoT Wireless

支援 ABAC (政策中的標籤)	是

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 AWS 中，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色)，以及許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

若要根據標籤控制存取，請使用 aws:ResourceTag/*key-name*、aws:RequestTag/*key-name* 或 aws:TagKeys 條件索引鍵，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索引鍵，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索引鍵，則值為 Partial。

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

您可以將標籤連接到 AWS IoT Wireless 資源，或以請求方式將標籤傳遞到 AWS IoT Wireless。若要根據標籤控制存取，請使用 YOUR-SERVICE-PREFIX: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。如需標記 AWS IoT Wireless 資源的詳細資訊，請參閱 標記您的 AWS IoT Wireless 資源。

將臨時憑證與 AWS IoT Wireless 搭配使用

支援臨時憑證	是

您使用臨時憑證進行登入時，某些 AWS 服務 無法運作。如需詳細資訊，包括那些 AWS 服務 搭配臨時憑證運作，請參閱《IAM 使用者指南》中的可搭配 IAM 運作的 AWS 服務。

如果您使用使用者名稱和密碼之外的任何方法登入 AWS Management Console，則您正在使用臨時憑證。例如，當您使用公司的單一登入(SSO)連結存取 AWS 時，該程序會自動建立臨時憑證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南中的切換至角色 (主控台)。

您可使用 AWS CLI 或 AWS API，手動建立臨時憑證。接著，您可以使用這些臨時憑證來存取 AWS。AWS 建議您動態產生臨時憑證，而非使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

AWS IoT Wireless 的跨服務主體權限

支援轉寄存取工作階段 (FAS)	是

當您使用 IAM 使用者或角色在 AWS 中執行動作時，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用主體的許可呼叫 AWS 服務，搭配請求

AWS 服務 以向下游服務發出請求。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之後，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

支援服務角色	否

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

AWS IoT Wireless 的服務連結角色

支援服務連結角色。	否

服務連結角色是一種連結到 AWS 服務 的服務角色類型。服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的 AWS 帳戶 中，並由該服務所擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

AWS IoT Wireless 身分型政策範例

根據預設，IAM 使用者和角色不具備建立或修改 AWS IoT Wireless 資源的許可。他們也無法使用 AWS Management Console、AWS CLI 或 AWS API 執行任務。IAM 管理員必須建立 IAM 政策，授予使用者和角色在指定資源上執行特定 API 操作的所需許可。管理員接著必須將這些政策連接至需要這些許可的 IAM 使用者或群組。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的在 JSON 標籤上建立政策。

主題

- 政策最佳實務
- 使用 AWS IoT Wireless 主控台
- 允許使用者檢視他們自己的許可
- 執行 AWS IoT Wireless 無線裝置動作所需的許可

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 Wireless 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 受管政策並朝向最低權限許可的目標邁進：如需開始授予許可給使用者和工作負載，請使用 AWS 受管政策，這些政策會授予許可給許多常用案例。它們可在您的 AWS 帳戶 中使用。我們建議您定義特定於使用案例的 AWS 客戶管理政策，以便進一步減少許可。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。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授予對服務動作的存取權，前提是透過特定 AWS 服務 (例如 AWS CloudFormation)使用條件。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 IAM Access Analyzer 政策驗證。
- 需要多重要素驗證 (MFA)：如果存在需要 AWS 帳戶中 IAM 使用者或根使用者的情況，請開啟 MFA 提供額外的安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 IAM 使用者指南中的設定 MFA 保護的 API 存取。

有關 IAM 中最佳實務的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 最佳安全實務。

使用 AWS IoT Wireless 主控台

若要存取 AWS IoT Wireless 主控台，您必須擁有一組最低限度的許可。這些許可必須允許您列出和檢視您 AWS 帳戶中 AWS IoT Wireless 資源的詳細資訊。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 (IAM 使用者或角色) 而言，主控台就無法如預期運作。

為確保那些實體仍可使用 AWS IoT Wireless 主控台，請同時將以下 AWS 受管政策連接到實體。如需更多資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

```
AWSIoTWirelessFullAccess
```

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許其最基本主控台許可。反之，只需允許存取符合您嘗試執行之 API 操作的動作就可以了。

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上，或是使用 AWS CLI 或 AWS API 透過編寫程式的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": "*"
        }
    ]
}
```

執行 AWS IoT Wireless 無線裝置動作所需的許可

您可以在身分型政策中使用條件來控制 AWS IoT Wireless 動作的存取權。此範例會示範如何建立政策，以允許建立和管理裝置。但是，只有在物件標籤 Owner 的值是該使用者的使用者名稱時，才會授予該許可。此政策也會授予在主控台完成此動作的必要許可。

```
{
 "Version": "2012-10-17",
 "Statement": [{
    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
              "iotwireless:CreateWirelessDevice",
              "iotwireless:GetWirelessDevice",
              "iotwireless:ListWirelessDevices",
              "iotwireless:UpdateWirelessDevice",
              "iotwireless:DeleteWirelessDevice"
           ],
    "Resource": "*"
    }
 ]
}
```

此政策中有一個陳述式，會授予使用 `CreateWirelessDevice`、`GetWirelessDevice`、`ListWirelessDevices`、`UpdateWirelessDevic` 和 `DeleteWirelessDevice` 動作的許可。AWS IoT Wireless 會呼叫這些方法來建立和管理您的無線裝置。

此政策不會指定主體元素，因為您不會在身分型政策中指定取得許可的主體。當您將政策連接至使用者時，這名使用者是隱含委託人。當您將許可政策連接至 IAM 角色，該角色的信任政策中所識別的委託人即取得許可。

AWS IoT Wireless 的 AWS 受管政策

若要新增許可給使用者、群組和角色，使用 AWS 受管政策比自己撰寫政策更容易。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 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。如需有關 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。您無法更改 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 支援跨越多項服務之任務職能的受管政策。例如，ReadOnlyAccess 這項 AWS 受管政策提供針對所有 AWS 服務和資源的唯讀存取權限。當服務啟動新功能時，AWS 會為新的操作和資源新增唯讀許可。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：AWSIoTWirelessDataAccess

您可將 `AWSIoTWirelessDataAccess` 政策連接到 IAM 身分。

此政策會授予相關的身分許可，以允許使用 `SendDataToWirelessDevice` API 將資料傳送至 LoRaWAN 和 Sidewalk 裝置。若要在 AWS Management Console 中檢視此政策，請參閱 AWSIoTWirelessDataAccess。

許可詳細資訊

此政策包含以下許可。

- `iotwireless`：擷取 AWS IoT Wireless 資料。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iotwireless:SendDataToWirelessDevice"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

AWS 受管政策：AWSIoTWirelessFullAccess

您可將 AWSIoTWirelessFullAccess 政策連接到 IAM 身分。

此政策會授予相關的身分許可，以允許完整存取所有 AWS IoT Wireless 操作。若要在 AWS Management Console 中檢視此政策，請參閱 AWSIoTWirelessFullAccess。

許可詳細資訊

此政策包含以下許可。

- iotwireless：擷取 AWS IoT Wireless 資料和執行所有 AWS IoT Wireless 操作。

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

AWS 受管政策：AWSIoTWirelessFullPublishAccess

您可將 AWSIoTWirelessFullPublishAccess 政策連接到 IAM 身分。

此政策會授予相關的身分許可，以允許代表您發佈至 AWS IoT 規則的有限存取權。若要在 AWS Management Console 中檢視此政策，請參閱 AWSIoTWirelessFullPublishAccess。

許可詳細資訊

此政策包含以下許可。

- iot：執行取得端點 URL 和發佈至 AWS IoT 規則引擎的操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeEndpoint",
                "iot:Publish"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 受管政策：AWSIoTWirelessLogging

您可將 AWSIoTWirelessLogging 政策連接到 IAM 身分。

此政策會授予相關的身分許可，以允許建立 Amazon CloudWatch Logs 日誌群組，以及將日誌串流至群組。此政策會連接至您的 CloudWatch 記錄角色。若要在 AWS Management Console 中檢視此政策，請參閱 AWSIoTWirelessLogging。

許可詳細資訊

此政策包含以下許可。

- `logs` – 擷取 CloudWatch 日誌。此外，也允許建立 CloudWatch 日誌群組，以及將日誌串流至群組。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/iotwireless*"
        }
    ]
}
```

AWS 受管政策：AWSIoTWirelessReadOnlyAccess

您可將 `AWSIoTLogging` 政策連接到 IAM 身分。

此政策會授予相關的身分許可，以允許透過唯讀方式存取 AWS IoT Wireless 操作。若要在 AWS Management Console 中檢視此政策，請參閱 `AWSIoTWirelessReadOnlyAccess`。

許可詳細資訊

此政策包含以下許可。

- `logs`：執行 AWS IoT Wireless `List` 和 `Get` API 操作。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iotwireless:List*",
                "iotwireless:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 受管政策：AWSIoTWirelessGatewayCertManager

您可將 AWSIoTWirelessGatewayCertManager 政策連接到 IAM 身分。

此政策會授予相關的身分許可，以允許建立、列出和描述 AWS IoT 憑證。若要在 AWS Management Console 中檢視此政策，請參閱 AWSIoTWirelessGatewayCertManager。

許可詳細資訊

此政策包含以下許可。

- iot：執行建立、描述和列出憑證的動作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IoTWirelessGatewayCertManager",
            "Effect": "Allow",
            "Action": [
                "iot:CreateKeysAndCertificate",
                "iot:DescribeCertificate",
                "iot:ListCertificates"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

AWS：AWS IoT Wireless 受管政策更新

檢視自 AWS IoT Wireless 開始追蹤 AWS 管理的政策變更以來的更新詳細資訊。如需自動收到有關此頁面變更的提醒，請前往 AWS IoT Wireless 文件歷史記錄頁面上訂閱 RSS 摘要。

變更	描述	日期
AWS IoT Wireless 已開始追蹤變更	AWS IoT Wireless 已開始追蹤其 AWS 管理的政策的變更。	2022 年 5 月 18 日

對 AWS IoT Wireless 身分和存取進行故障診斷

請使用以下資訊來協助您診斷和修復使用 AWS IoT Wireless 和 IAM 時可能發生的常見問題。

主題

- 我未獲得在 AWS IoT Wireless 中執行動作的授權
- 我想要檢視我的存取金鑰
- 我是管理員，我想要允許其他人存取 AWS IoT Wireless
- 我想要允許 AWS 帳戶外的人存取我的 AWS IoT Wireless 資源

我未獲得在 AWS IoT Wireless 中執行動作的授權

若 AWS Management Console 告知您並未獲得執行動作的授權，您必須聯絡您的管理員以取得協助。您的管理員是提供您使用者名稱和密碼的人員。

以下範例錯誤會在 `mateojackson` IAM 使用者嘗試使用主控台檢視 *`WirelessDevice`* 的詳細資訊，但卻沒有 `YOUR-SERVICE-PREFIX:`*`GetWirelessDevice`* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: YOUR-SERVICE-PREFIX:GetWirelessDevice on resource: my-LoRaWAN-device
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-LoRaWAN-device* 動作存取 YOUR-SERVICE-PREFIX:*GetWirelessDevice* 資源。

我想要檢視我的存取金鑰

在您建立 IAM 使用者存取金鑰後，您可以隨時檢視您的存取金鑰 ID。但是，您無法再次檢視您的私密存取金鑰。若您遺失了密碼金鑰，您必須建立新的存取金鑰對。

存取金鑰包含兩個部分：存取金鑰 ID (例如 AKIAIOSFODNN7EXAMPLE) 和私密存取金鑰 (例如 wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY)。如同使用者名稱和密碼，您必須一起使用存取金鑰 ID 和私密存取金鑰來驗證您的請求。就如對您的使用者名稱和密碼一樣，安全地管理您的存取金鑰。

> **Important**
>
> 請勿將您的存取金鑰提供給第三方，甚至是協助尋找您的標準使用者 ID。透過執行此操作，可能會讓他人永久存取您的 AWS 帳戶。

建立存取金鑰對時，您會收到提示，要求您將存取金鑰 ID 和私密存取金鑰儲存在安全位置。私密存取金鑰只會在您建立它的時候顯示一次。若您遺失了私密存取金鑰，您必須將新的存取金鑰新增到您的 IAM 使用者。您最多可以擁有兩個存取金鑰。若您已有兩個存取金鑰，您必須先刪除其中一個金鑰對，才能建立新的金鑰對。若要檢視說明，請參閱《IAM 使用者指南》中的管理存取金鑰。

我是管理員，我想要允許其他人存取 AWS IoT Wireless

若要允許其他人存取 AWS IoT Wireless，您必須針對需要存取的人員或應用程式建立 IAM 實體 (使用者或角色)。他們將使用該實體的憑證來存取 AWS。接著您必須將政策連接到實體，以便在 AWS IoT Wireless 中授予他們正確的許可。

若要立即開始使用，請參閱《IAM 使用者指南》中的建立您的第一個 IAM 委派使用者及群組。

我想要允許 AWS 帳戶外的人存取我的 AWS IoT Wireless 資源

您可以建立一個角色，讓其他帳戶中的使用者或您的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 AWS IoT Wireless 是否支援這些功能，請參閱 AWS IoT Wireless 如何與 IAM 搭配使用。

- 如需了解如何存取您擁有的所有 AWS 帳戶 所提供的資源，請參閱《IAM 使用者指南》中的將存取權提供給您所擁有的另一個 AWS 帳戶 中的 IAM 使用者。
- 如需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擁有的 AWS 帳戶。
- 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)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 角色與資源型政策的差異。

AWS IoT Wireless 合規驗證

在多個 AWS 合規計劃中，第三方稽核人員會評估 AWS IoT Wireless 的安全與合規情形。這些計劃包括 SOC、PCI、FedRAMP、HIPAA 等等。

如需特定合規計劃範圍內的 AWS 服務清單，請參閱合規計劃範圍內的 AWS 服務。如需一般資訊，請參閱 AWS 合規計劃 。

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱 AWS Artifact 中的下載報告。

您使用 AWS IoT Wireless 時的合規責任取決於資料的敏感度、您的公司的合規目標，以及適用的法律和法規。AWS 提供以下資源協助您處理合規事宜：

- 安全與合規快速入門指南：這些部署指南討論架構考量，並提供在 AWS 上部署以安全及合規為重心之基準環境的步驟。
- HIPAA 安全與合規架構白皮書：本白皮書說明公司可如何運用 AWS 來建立 HIPAA 合規的應用程式。
- AWS 合規資源 – 這組手冊和指南可能適用於您的產業和位置。
- 《AWS Config 開發人員指南》中的使用規則評估資源：AWS Config 可評估資源組態對於內部實務、業界準則和法規的合規狀態。
- AWS Security Hub：此 AWS 服務可供您檢視 AWS 中的安全狀態，可助您檢查是否符合安全產業標準和最佳實務。

AWS IoT Wireless 的復原能力

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援網路連線相互連結。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、容錯能力和可擴展性能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域與可用區域的詳細資訊，請參閱 AWS全球基礎設施。

AWS IoT Wireless 中的基礎設施安全性

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

您可使用 AWS 發佈的 API 呼叫，透過網路存取 AWS IoT Wireless。用戶端必須支援 Transport Layer Security (TLS) 1.0 或更新版本。建議使用 TLS 1.2 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。

使用 Amazon CloudWatch Logs 監控您的 AWS IoT Wireless 資源

監控是維護 AWS IoT Wireless 與其他 AWS 解決方案的可靠性、可用性和效能的重要環節。您可以同時監控 LoRaWAN 和 Sidewalk 裝置，並從裝置加入 AWS IoT Wireless 開始即取得參考訊息和錯誤。

我們強烈建議您從 AWS 解決方案的所有部分收集監控資料，以更容易在發生多點失敗時偵錯。請在一開始先建立可回答下列問題的監視計劃。如果您不確定如何回答這些問題，您仍然可以繼續啟用記錄並建立效能基準。

- 監控目標是什麼？
- 監控哪些資源？
- 監控這些資源的頻率為何？
- 將使用哪些監控工具？
- 誰將執行監控任務？
- 發生問題時應該通知誰？

您的下一步是啟用記錄，並在各個時間點和不同的負載條件下測量效能，以在您的環境中確立 AWS IoT Wireless 正常效能的基準。監控 AWS IoT Wireless 時，請保留歷史監控資料，以便與目前的效能資料進行比較。這可協助您識別正常效能模式和效能異常情況，並策劃解決這些情況的方法。

監控工具

您可以使用下列監控工具來監看 AWS IoT Wireless，在發現問題時回報，並適時自動採取行動：

- Amazon CloudWatch 會即時監控您的 AWS 資源，以及您在 AWS 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，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，您可以讓 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。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/。
- 網路分析器可讓您監控 LoRaWAN 資源，包括 LoRaWAN 裝置和閘道在內，並且可縮短設定連線以開始接收追蹤訊息所需的時間，進而為您提供即時日誌資訊。如需詳細資訊，請參閱使用網路分析器即時監控無線資源機群。

如何使用 Amazon CloudWatch 監控資源

您可以使用 CloudWatch 監控 AWS IoT Wireless，它會收集原始資料並將該資料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 CloudWatch 使用者指南。

若要記錄並監控您的 AWS IoT Wireless 資源，請執行下列步驟：

1. 建立記錄角色以記錄您的 AWS IoT Wireless 資源，如 建立 AWS IoT Wireless 的記錄角色和政策 所述。
2. CloudWatch Logs 主控台中的記錄訊息具有預設日誌層級 ERROR，此層級不太詳細，而且只包含錯誤資訊。如果您想要檢視更多詳細訊息，我們建議您首先使用 CLI 來設定記錄，如 設定 AWS IoT Wireless 資源的記錄 所述。
3. 接下來，您可以透過檢視 CloudWatch Logs 主控台中的日誌項目來監控您的資源。如需詳細資訊，請參閱檢視 CloudWatch AWS IoT Wireless 日誌項目。
4. 您可以使用 Logs groups (記錄群組) 來建立篩選表達式，但我們建議您首先在記錄群組中建立簡單篩選條件並檢視記錄項目，然後移至 CloudWatch Insights 來建立查詢，以根據您正在監控的資源或事件篩選記錄項目。如需詳細資訊，請參閱使用 CloudWatch Insights 來篩選 AWS IoT Wireless 的記錄。

設定 AWS IoT Wireless 的記錄

在您可以監控並記錄 AWS IoT 活動之前，請使用 CLI 或 API 來啟用 AWS IoT Wireless 資源的記錄。

在考慮如何設定 AWS IoT Wireless 記錄時，除非另有指定，否則預設記錄組態會決定記錄 AWS IoT 活動的方式。開始之後，您可能想要取得預設日誌層級為 INFO 的預設記錄。

在檢閱初始記錄之後，您可以將預設日誌層級變更為較不詳細的 ERROR，並在可能需要更多注意的資源上設定更詳細的資源特定日誌層級。您可以隨時更改日誌層級。

下列主題顯示如何設定 AWS IoT Wireless 資源的記錄。

主題

- 建立 AWS IoT Wireless 的記錄角色和政策
- 設定 AWS IoT Wireless 資源的記錄

建立 AWS IoT Wireless 的記錄角色和政策

下列顯示如何建立僅適用於 AWS IoT Wireless 資源的記錄角色。如果您也想要建立 AWS IoT Core 的記錄角色，請參閱 https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html。

建立 AWS IoT Wireless 的記錄角色

啟用記錄功能之前，您必須先建立 IAM 角色和授予代表您監控 AWS IoT Wireless 活動之 AWS 許可的政策。

建立 IAM 角色進行記錄

若要建立 AWS IoT Wireless 記錄角色，請開啟 Roles hub of the IAM console (IAM 主控台的角色中樞)，然後選擇 Create role (建立角色)。

1. 在 Select type of trusted entity (選取信任的實體類型) 下，選擇 Another AWS account (另一個 帳戶)。
2. 在 Account ID (帳戶 ID) 中，輸入您的 AWS 帳戶 ID，然後選擇 Next: Permissions (下一步：許可)。
3. 在搜尋方塊中，輸入 **`AWSIoTWirelessLogging`**。
4. 選取名為 AWSIoTWirelessLogging 之政策旁邊的方塊，然後選擇 Next: Tags (下一步：標籤)。
5. 選擇 下一步：檢閱。
6. 在 Role name (角色名稱) 中，輸入 **`IoTWirelessLogsRole`**，然後選擇 Create role (建立角色)。

編輯 IAM 角色的信任關係

在執行前一個步驟之後顯示的確認訊息中，選擇您已建立的角色名稱 (IoTWirelessLogsRole)。接下來，您將編輯角色以新增下列信任關係。

1. 在角色 IoTWirelessLogsRole 的 Summary (摘要) 區段中，選擇 Trust relationships (信任關係) 標籤，然後選擇 Edit trust relationship (編輯信任關係)。
2. 在 Policy Document (政策文件) 中，變更 `Principal` 屬性以看起來像此範例。

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

在您變更 Principal 屬性之後，完整政策文件應該看起來像此範例。

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

3. 若要儲存您的變更，請選擇 Update Trust Policy (更新信任政策)。

AWS IoT Wireless 的記錄政策

下列政策文件提供了角色政策及信任政策，用以允許 AWS IoT Wireless 代表您將日誌項目提交給 CloudWatch。

> **Note**
>
> 當您建立記錄角色 (IoTWirelessLogsRole) 時，便會自動為您建立 AWS 受管政策文件。

角色政策

下列顯示角色政策文件。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
```

```
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/iotwireless*"
        }
    ]
}
```

僅記錄 AWS IoT Wireless 活動的信任政策

下列顯示僅用於記錄 AWS IoT Wireless 活動的信任政策。

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

如果您已建立 IAM 角色以同時記錄 AWS IoT Core 活動，則政策文件允許記錄這兩個活動。如需建立 AWS IoT Core 記錄角色的相關資訊，請參閱 https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html。

後續步驟

您已了解如何建立記錄角色來記錄 AWS IoT Wireless 資源。根據預設，記錄的日誌層級為 ERROR，所以如果您只想要看到錯誤資訊，請移至 檢視 CloudWatch AWS IoT Wireless 日誌項目，藉由檢視記錄項目來監控您的無線資源。

如果想要記錄項目中有更多資訊，您可以為您的資源或不同的事件類型設定預設日誌層級，例如將日誌層級設定為 INFO。如需設定資源記錄的相關資訊，請參閱 設定 AWS IoT Wireless 資源的記錄。

設定 AWS IoT Wireless 資源的記錄

若要設定 AWS IoT Wireless 資源的記錄，您可以使用 API 或 CLI。當開始監控 AWS IoT Wireless 資源時，您可以使用預設組態。若要這樣做，您可以略過此主題，然後進入 使用 CloudWatch Logs 監控 AWS IoT Wireless 來監控您的記錄。

在開始監控記錄之後，您可以使用 CLI 將日誌層級變更為更詳細選項，例如提供 INFO 和 ERROR 資訊，並啟用更多資源的記錄。

AWS IoT Wireless 資源和日誌層級

在使用 API 或 CLI 之前，請使用下表了解不同的日誌層級，以及您可以設定其記錄的資源。此表格會顯示您在監控資源時於 CloudWatch 記錄中看到的參數。設定資源記錄的方式將決定您在主控台中看到的記錄。

如需範例 CloudWatch 日誌外觀的相關資訊，以及如何使用這些參數來記錄 AWS IoT Wireless 資源的實用資訊，請參閱 檢視 CloudWatch AWS IoT Wireless 日誌項目。

日誌層級和資源

名稱	可能的值	描述
logLevel	INFO、ERROR 或 DISABLED	• ERROR：顯示任何導致操作失敗的錯誤。記錄僅包含 ERROR 資訊。 • INFO：提供物件流的高層級資訊。記錄包含 INFO 和 ERROR 資訊。 • DISABLED：停用所有記錄。
resource	WirelessGateway 或 WirelessDevice	資源的類型，可以是 WirelessGateway 或 WirelessDevice 。
wirelessGatewayType	LoRaWAN	無線閘道的類型，當 resource 為 WirelessGateway 時，一律為 LoRaWAN。
wirelessDeviceType	LoRaWAN 或 Sidewalk	無線裝置的類型，當 resource 為 WirelessDevice 時，可以是 LoRaWAN 或 Sidewalk。
wirelessGatewayId	-	無線閘道的識別符，當 resource 為 WirelessGateway 時。

名稱	可能的值	描述
wirelessDeviceId	-	無線裝置的識別符，當 resource 為 WirelessDevice 時。
event	Join、Rejoin、Registration 、Uplink_data 、Downlink_data 、CUPS_Request 和 Certificate	記錄的事件類型，取決於您所記錄的資源是無線裝置還是無線閘道。如需詳細資訊，請參閱檢視 CloudWatch AWS IoT Wireless 日誌項目。

AWS IoT Wireless 記錄 API

您可以使用下列 API 動作來設定資源的記錄。此表格也會顯示您必須為了使用 API 動作而建立的範例 IAM 政策。下節描述如何使用 API 來設定資源的日誌層級。

記錄 API 動作

API 名稱	描述	範例 IAM 政策

GetLogLevelsByResourceTypes	傳回目前的預設日誌層級，或依資源類型傳回日誌層級，其中可包含無線裝置或無線閘道的記錄選項。	<pre>{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Action": [
		"iotwireless:GetLogLevelsByResourceTypes"
		],
		"Resource": [
		"*"</pre>

API 名稱	描述	範例 IAM 政策
		] }] }

GetResourceLogLevel	傳回特定資源識別符和資源類型的日誌層級覆寫。資源可以是無線裝置或無線閘道。	{
		"Version":
		"2012-10-17",
		"Statement": [
		{
		"Effect":
		"Allow",
		"Action": [
		"iotwireless:GetResourceLogLevel"
		],
		"Resource":
		[
		"arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc537-ab12-cd3a-d00e-1f0e20c1204a",
		]
		}
		]
		}

API 名稱	描述	範例 IAM 政策

PutResourceLogLevel	設定特定資源識別符和資源類型的日誌層級覆寫。資源可以是無線閘道或無線裝置。
	
	Note
	此 API 每個帳戶的日誌層級覆寫限制為 200 個。
		{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Action": [
		"iotwireless:PutResourceLogLevel"
		],
		"Resource": [
		"arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc537-ab12-cd3a-d00e-1f0e20c1204a",
		]
		}
		]
		}

API 名稱	描述	範例 IAM 政策

ResetAllResourceLogLevels	移除所有資源 (包括無線閘道和無線裝置) 的日誌層級覆寫。
	
	Note
	此 API 不會影響使用 `UpdateLogLevelsByResourceTypes` API 所設定的日誌層級。
		<pre>{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Action": [
		"iotwireless:ResetAllResourceLogLevels"
		],
		"Resource": [
		"arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/*",
		"arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/*
		]
		}
		]
		}</pre>

API 名稱	描述	範例 IAM 政策

ResetResourceLogLevel	移除特定資源識別符和資源類型的日誌層級覆寫。資源可以是無線閘道或無線裝置。	{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Action": [
		"iotwireless:ResetResourceLogLevel"
		],
		"Resource": [
		"arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc537-ab12-cd3a-d00e-1f0e20c1204a",
		]
		}
		]
		}

API 名稱	描述	範例 IAM 政策

UpdateLogLevelsByResourceTypes	設定預設日誌層級或依資源類型設定日誌層級。您可以將此 API 用於無線裝置或無線閘道的記錄選項，並控制將在 CloudWatch 中顯示的記錄訊息。
	
	Note
	事件是選用的，而且事件類型繫結至資源類型。如需詳細資訊，請參閱事件和資源類型。
		<pre>{
		    "Version":
		"2012-10-17",
		    "Statement": [
		
		        {
		            "Effect":
		"Allow",
		            "Action": [
		
		"iotwireless:Updat
		eLogLevelsByResour
		ceTypes"
		            ],
		
		            "Resource":
		[
		                "*"
		
		            ]
		        }
		    ]
		}</pre>

使用 CLI 設定資源的日誌層級

本節描述如何使用 API 或 AWS CLI 設定 AWS IoT Wireless 的日誌層級。

在使用 CLI 之前：

- 確定您已針對要執行 CLI 命令的 API 建立 IAM 政策，如先前所述。
- 您需要您想要使用的角色的 Amazon 資源名稱 (ARN)。如果您需要建立要用於記錄的角色，請參閱建立 AWS IoT Wireless 的記錄角色和政策。

為什麼要使用 AWS CLI

依預設，如果建立 IAM 角色 IoTWirelessLogsRole (如 建立 AWS IoT Wireless 的記錄角色和政策 所述)，您會在 AWS Management Console 中看到預設日誌層級為 ERROR 的 CloudWatch 記錄。若要變更所有資源或特定資源的預設日誌層級，請使用 AWS IoT Wireless 記錄 API 或 CLI。

如何使用 AWS CLI

API 動作可以分類為下列類型，取決於您想要設定所有資源還是特定資源的日誌層級：

- API 動作 GetLogLevelsByResourceTypes 和 UpdateLogLevelsByResourceTypes 可以擷取並更新您帳戶中屬於特定類型 (例如無線閘道或 LoRaWAN 或 Sidewalk 裝置) 之所有資源的日誌層級。
- API 動作 GetResourceLogLevel、PutResourceLogLevel 和 ResetResourceLogLevel 可以擷取、更新和重設您使用資源識別符指定之個別資源的日誌層級。
- 對於您已使用 PutResourceLogLevel API 為其指定日誌層級覆寫的所有資源，API 動作 ResetAllResourceLogLevels 可將其日誌層級覆寫重設為 null。

使用 CLI 設定 AWS IoT 的資源特定記錄

> **Note**
>
> 您也可以使用 API 中對應於此處顯示的 CLI 命令的方法，在 AWS API 中執行此程序。

1. 依預設，所有資源的日誌層級都設定為 ERROR。若要為您帳戶中的所有資源設定預設日誌層級，或依資源類型設定日誌層級，請使用 update-log-levels-by-resource-types 命令。下列範例顯示如何建立 JSON 檔案 Input.json，並提供此檔案作為 CLI 命令的輸入。您可以使用此命令來選擇性地停用記錄，或覆寫特定類型的資源和事件的預設日誌層級。

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions":
     [
        {
         "Type": "Sidewalk",
         "LogLevel": "INFO",
         "Events":
          [
            {
              "Event": "Registration",
              "LogLevel": "DISABLED"
```

```
            }
          ]
        },
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "Join",
             "LogLevel": "DISABLED"
            },
            {
             "Event": "Rejoin",
             "LogLevel": "ERROR"
            }
          ]
        }
      ]
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
            }
          ]
        }
      ]
}
```

其中：

WirelessDeviceLogOptions

無線裝置的記錄選項清單。每個記錄選項都包含無線裝置類型 (Sidewalk 或 LoRaWAN)，以及無線裝置事件記錄選項的清單。每個無線裝置事件記錄選項都可以選擇性地包含事件類型及其日誌層級。

WirelessGatewayLogOptions

無線閘道的記錄選項清單。每個記錄選項都包含無線閘道類型 (LoRaWAN)，以及無線閘道事件記錄選項的清單。每個無線閘道事件記錄選項都可以選擇性地包含事件類型及其日誌層級。

DefaultLogLevel

用於所有資源的日誌層級。有效值為：ERROR、INFO 和 DISABLED。預設值為 INFO。

LogLevel

您要用於個別資源類型和事件的日誌層級。這些日誌層級會覆寫預設日誌層級，例如 LoRaWAN 閘道的日誌層級 INFO，以及兩種事件類型的日誌層級 DISABLED 和 ERROR。

請執行下列命令來提供 Input.json 檔案，作為命令的輸入。此命令不會產生任何輸出。

```
aws iotwireless update-log-levels-by-resource-types \
    --cli-input-json Input.json
```

如果您想要移除無線裝置和無線閘道的記錄選項，請執行下列命令。

```
{
    "DefaultLogLevel":"DISABLED",
    "WirelessDeviceLogOptions": [],
    "WireslessGatewayLogOptions":[]
}
```

2. update-log-levels-by-resource-types 命令不會傳回任何輸出。使用 get-log-levels-by-resource-types 命令來擷取資源特定的記錄資訊。此命令會傳回預設日誌層級，以及無線裝置和無線閘道記錄選項。

> **Note**
>
> get-log-levels-by-resource-types 命令無法直接擷取 CloudWatch 主控台中的日誌層級。您可以使用 get-log-levels-by-resource-types 命令，來取得您已使用 update-log-levels-by-resource-types 命令，為資源指定的最新日誌層級資訊。

```
aws iotwireless get-log-levels-by-resource-types
```

當您執行下列命令時，它會傳回您已使用 update-log-levels-by-resource-types 指定的最新記錄資訊。例如，如果您移除無線裝置記錄選項，則執行 get-log-levels-by-resource-types 將傳回此值作為 `null`。

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions": null,
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
            }
          ]
        }
      ]
}
```

3. 若要控制個別無線閘道或無線裝置資源的日誌層級，請使用下列 CLI 指令：

   - put-resource-log-level
   - get-resource-log-level

- reset-resource-log-level

舉例來說，使用這些 CLI 的時機，假設您的帳戶中有大量正在記錄的無線裝置或閘道。如果只想要針對某些無線裝置進行疑難排解，您可以將 DefaultLogLevel 設定為 DISABLED 來停用所有無線裝置的記錄，然後使用 put-resource-log-level，只針對您帳戶中的那些裝置將 LogLevel 設定為 ERROR。

```
aws iotwireless put-resource-log-level \
    --resource-identifier
    --resource-type WirelessDevice
    --log-level ERROR
```

在此範例中，命令只會針對指定的無線裝置資源將日誌層級設定為 ERROR，而且所有其他資源的記錄都會停用。此命令不會產生任何輸出。若要擷取此資訊並驗證是否已設定日誌層級，請使用 get-resource-log-level 命令。

4. 在上一個步驟中，您已偵錯問題並解決了錯誤，因此您可以執行 reset-resource-log-level 命令，將該資源的日誌層級重設為 null。如果您已使用 put-resource-log-level 命令來設定多個無線裝置或閘道資源的日誌層級覆寫 (例如疑難排解多個裝置的錯誤)，則可以使用 reset-all-resource-log-levels 命令，將所有這些資源的日誌層級覆寫重設回 null。

```
aws iotwireless reset-all-resource-log-levels
```

此命令不會產生任何輸出。若要擷取資源的記錄資訊，請執行 get-resource-log-level 命令。

後續步驟

您已了解如何建立記錄角色，以及使用 AWS IoT Wireless API 來設定 AWS IoT Core for LoRaWAN 資源的記錄。接下來，若要了解如何監控記錄項目，請移至 使用 CloudWatch Logs 監控 AWS IoT Wireless。

使用 CloudWatch Logs 監控 AWS IoT Wireless

AWS IoT Core for LoRaWAN 具有超過 50 個依預設會啟用的 CloudWatch 日誌項目。每個記錄項目都會描述事件類型、日誌層級和資源類型。如需詳細資訊，請參閱AWS IoT Wireless 資源和日誌層級。

如何監控您的 AWS IoT Wireless 資源

當啟用 AWS IoT Wireless 進行記錄時，AWS IoT Wireless 會在其透過 AWS IoT 從您的裝置來回傳遞時傳送有關每則訊息的進度事件。依預設，AWS IoT Wireless 日誌項目的預設日誌層級為錯誤。當啟用記錄 (如 建立 AWS IoT Wireless 的記錄角色和政策 所述) 時，您會在 CloudWatch 主控台中看到預設日誌層級為 ERROR 的訊息。藉由使用此日誌層級，訊息只會顯示您正在使用之所有無線裝置和閘道資源的錯誤資訊。

如果您想要記錄顯示其他資訊，例如日誌層級為 INFO 的資訊，或停用部分裝置的記錄，並僅顯示部分裝置的記錄訊息，則您可以使用 AWS IoT Wireless 記錄 API。如需詳細資訊，請參閱使用 CLI 設定資源的日誌層級。

您也可以建立篩選表達式，僅顯示必要的訊息。

在您可以檢視主控台中的 AWS IoT Wireless 記錄之前

若要使 /aws/iotwireless 記錄群組出現在 CloudWatch 主控台中，您必須已完成下列動作。

- 在 AWS IoT Wireless 中啟用登入。如需如何在 AWS IoT Wireless 中啟用記錄的詳細資訊，請參閱 設定 AWS IoT Wireless 的記錄。
- 已執行 AWS IoT Wireless 操作來寫入一些日誌項目。

若要更有效地建立和使用篩選表達式，建議您嘗試使用 CloudWatch Insights，如下列主題所述。我們也建議您依照主題在此處顯示的順序，遵循這些主題進行。這將協助您首先使用 CloudWatch Log groups (記錄群組)，來了解不同類型的資源、其事件類型，以及您可以用來檢視主控台中記錄項目的日誌層級。然後，您可以了解如何使用 CloudWatch Insights 來建立篩選表達式，從資源中取得更多實用資訊。

主題

- 檢視 CloudWatch AWS IoT Wireless 日誌項目
- 使用 CloudWatch Insights 來篩選 AWS IoT Wireless 的記錄

檢視 CloudWatch AWS IoT Wireless 日誌項目

在設定了 AWS IoT Wireless 的記錄 (如 建立 AWS IoT Wireless 的記錄角色和政策 所述) 並寫入一些日誌項目之後，您可以執行下列步驟，在 CloudWatch 主控台中檢視記錄項目。

在 CloudWatch Log 群組主控台中檢視 AWS IoT 記錄

在 CloudWatch 主控台 中，CloudWatch 記錄出現在名為 /aws/iotwireless 的記錄群組中。如需 CloudWatch Logs 的詳細資訊，請參閱 CloudWatch Logs。

在 CloudWatch 主控台中檢視您的 AWS IoT 記錄

導覽至 CloudWatch 主控台，然後在導覽窗格中選擇 Log groups (記錄群組)。

1. 在 Filter (篩選條件) 文字方塊中，輸入 **/aws/iotwireless**，然後選擇 /aws/iotwireless 記錄群組。
2. 若要查看為您帳戶所產生之 AWS IoT Core for LoRaWAN 記錄的完整清單，請選擇 Search all (全部搜尋)。若要查看個別日誌串流，請選擇展開圖示。
3. 若要篩選日誌串流，您也可以在 Filter events (篩選事件)文字方塊中輸入一個查詢。以下是一些可以試試看的查詢：

   - `{ $.logLevel = "ERROR" }`

     使用此篩選條件來尋找日誌層級為 ERROR 的所有記錄，而且您可以展開個別錯誤串流來讀取錯誤訊息，這將協助您解決它們。
   - `{ $.resource = "WirelessGateway" }`

     尋找 WirelessGateway 資源的所有記錄，不管其日誌層級為何。
   - `{ $.event = "CUPS_Request" && $.logLevel = "ERROR" }`

     尋找事件類型為 CUPS_Request 且日誌層級為 ERROR 的所有記錄。

事件和資源類型

下表顯示您會看到其記錄項目的不同類型事件。事件類型也取決於資源類型是無線裝置還是無線閘道。您可以使用資源和事件類型的預設日誌層級，或是指定其中每一個的日誌層級來覆寫預設日誌層級。

以所使用資源為基礎的事件類型

資源	資源類型	事件類型	
無線閘道	LoRaWAN	• CUPS_Request • 憑證	
無線裝置	LoRaWAN	• 聯結 • 重新聯結 • Uplink_Data • Downlink_Data	

資源	資源類型	事件類型	
無線裝置	Sidewalk	• 註冊 • Uplink_Data • Downlink_Data	

下列主題包含這些事件類型的詳細資訊，以及無線閘道和無線裝置的記錄項目。

主題

無線閘道和無線裝置資源的記錄項目

在啟用了記錄之後，您可以檢視無線閘道和無線裝置的記錄項目。下節根據您的資源和事件類型描述各種記錄項目。

無線閘道記錄項目

本節顯示無線閘道資源的一些範例記錄項目，您將在 CloudWatch 主控台中看到它們。這些記錄訊息可以具有事件類型 `CUPS_Request` 或 `Certificate`，並且可以設定為在資源層級或事件層級顯示日誌層級 `INFO`、`ERROR` 或 `DISABLED`。如果您只想要看到錯誤資訊，請將日誌層級設定為 `ERROR`。`ERROR` 記錄項目中的訊息將包含其為何失敗的相關資訊。

無線閘道資源的記錄項目可以根據下列事件類型進行分類：

- CUPS_Request

  在閘道上執行的 LoRa Basics Station 會定期將請求傳送至組態與更新伺服器 (CUPS) 進行更新。對於此事件類型，如果您在為無線閘道資源設定 CLI 時將日誌層級設定為 `INFO`，則在記錄中：

  - 如果事件成功，您就會看到 `logLevel` 為 `INFO` 的記錄訊息。這些訊息將包含有關傳送至閘道之 CUPS 回應的詳細資訊以及閘道詳細資訊。下列顯示此記錄項目的範例。如需記錄項目中 `logLevel` 和其他欄位的詳細資訊，請參閱 AWS IoT Wireless 資源和日誌層級。

    ```
    {
        "timestamp": "2021-05-13T16:56:08.853Z",
        "resource": "WirelessGateway",
        "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
        "wirelessGatewayType": "LoRaWAN",
    ```

```
    "gatewayEui": "fefffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "INFO",
    "message": "Sending CUPS response of total length 3213 to GatewayEui:
 fefffff00000000e2 with  TC Credentials,"
}
```

- 如果發生錯誤，您會看到 `logLevel` 為 `ERROR` 的記錄項目，而且訊息將包含錯誤的詳細資訊。`CUPS_Request` 事件何時可能發生錯誤的範例包括：缺少 CUPS CRC、閘道的 TC Uri 與 AWS IoT Core for LoRaWAN 不符、缺少 `IoTWirelessGatewayCertManagerRole`，或無法取得無線閘道記錄。下列範例顯示缺少 CRC 記錄項目。若要解決此錯誤，請檢查您的閘道設定，以驗證您是否已輸入正確的 CUPS CRC。

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "fefffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "ERROR",
    "message": "The CUPS CRC is missing from the request. Check your gateway setup
 and enter the CUPS CRC,"
}
```

- 憑證

這些記錄項目將協助您檢查無線閘道是否呈現正確的憑證，以驗證與 AWS IoT 的連線。對於此事件類型，如果您在為無線閘道資源設定 CLI 時將日誌層級設定為 `INFO`，則在記錄中：

- 如果事件成功，您就會看到 `logLevel` 為 `INFO` 的記錄訊息。這些訊息將包含憑證 ID 和無線閘道識別符的詳細資訊。下列顯示此記錄項目的範例。如需記錄項目中 `logLevel` 和其他欄位的詳細資訊，請參閱 AWS IoT Wireless 資源和日誌層級。

```
{
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "event": "Certificate",
    "logLevel": "INFO",
    "message": "Gateway connection authenticated.
```

```
    (CertificateId:
 b5942a7aee973eda24314e416889227a5e0aa5ed87e6eb89239a83f515dea17c,
 WirelessGatewayId: 5da85cc8-3361-4c79-8be3-3360fb87abda)"
}
```

- 如果發生錯誤，您會看到 `logLevel` 為 `ERROR` 的記錄項目，而且訊息將包含錯誤的詳細資訊。`Certificate` 事件何時可能發生錯誤的範例包含無效的憑證 ID、無線閘道識別符，或無線閘道識別符與憑證 ID 之間不相符。下列範例顯示 `ERROR`，因為無線閘道識別符無效。若要解決此錯誤，請檢查閘道識別符。

```
{
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "event": "Certificate",
    "logLevel": "INFO",
    "message": "The gateway connection couldn't be authenticated because a
 provisioned gateway associated with the certificate couldn't be found.
    (CertificateId:
 729828e264810f6fc7134daf68056e8fd848afc32bfe8082beeb44116d709d9e)"
}
```

無線裝置記錄項目

本節顯示無線裝置資源的一些範例記錄項目，您將在 CloudWatch 主控台中看到它們。這些記錄訊息的事件類型取決於您使用的是 LoRaWAN 還是 Sidewalk 裝置。每個無線裝置資源或事件類型都可設定為顯示日誌層級 `INFO`、`ERROR` 或 `DISABLED`。

> **Note**
>
> 您的請求不得同時包含 LoRaWAN 和 Sidewalk 無線中繼資料。若要避免此案例的 `ERROR` 記錄項目，請指定 LoRaWAN 或 Sidewalk 無線資料。

LoRaWAN 裝置記錄項目

LoRaWAN 無線裝置的記錄項目可以根據下列事件類型進行分類：

- **`Join` 與 `Rejoin`**

當您新增 LoRaWAN 裝置並將其連接到 AWS IoT Core for LoRaWAN，然後您的裝置才能傳送上行資料時，您必須先完成稱為 `activation` 或 `join procedure` 的程序。如需詳細資訊，請參閱將您的無線裝置新增至 AWS IoT Core for LoRaWAN。

對於此事件類型，如果您在為無線閘道資源設定 CLI 時將日誌層級設定為 `INFO`，則在記錄中：

- 如果事件成功，您就會看到 `logLevel` 為 `INFO` 的記錄訊息。這些訊息將包含有關聯結或重新聯結請求之狀態的詳細資訊。下列顯示此記錄項目的範例。如需記錄項目中 `logLevel` 和其他欄位的詳細資訊，請參閱 AWS IoT Wireless 資源和日誌層級。

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessDevice",
    "wirelessDeviceType": "LoRaWAN",
    "WirelessDeviceId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "devEui": "fefffff00000000e2",
    "event": "Rejoin",
    "logLevel": "INFO",
    "message": "Rejoin succeeded"
}
```

- 如果發生錯誤，您會看到 `logLevel` 為 `ERROR` 的記錄項目，而且訊息將包含錯誤的詳細資訊。`Join` 和 `Rejoin` 事件何時可能發生錯誤的範例包括無效的 LoRaWAN 區域設定，或無效的訊息完整性代碼 (MIC) 檢查。下列範例顯示由於 MIC 檢查而產生的聯結錯誤。若要解決此錯誤，請檢查您是否已輸入正確的根金鑰。

```
{
    "timestamp": "2020-11-24T01:46:50.883481989Z",
    "resource": "WirelessDevice",
    "wirelessDeviceType": "LoRaWAN",
    "WirelessDeviceId": "cb4c087c-1be5-4990-8654-ccf543ee9fff",
    "devEui": "58a0cb000020255c",
    "event": "Join",
    "logLevel": "ERROR",
    "message": "invalid MIC. It's most likely caused by wrong root keys."
}
```

- Uplink_Data 與 Downlink_Data

事件類型 Uplink_Data 用於 LoRaWAN 或 Sidewalk 裝置向 AWS IoT 傳送承載時由 AWS IoT Wireless 產生的訊息。事件類型 Downlink_Data 用於與從 AWS IoT 傳送至無線裝置之下行訊息相關的訊息。

對於此事件類型，如果您在為無線裝置設定 CLI 時將日誌層級設定為 INFO，則在記錄中，您將看到：

- 如果事件成功，您就會看到 logLevel 為 INFO 的記錄訊息。這些訊息將包含所傳送之上行或下行訊息狀態的詳細資訊，以及無線裝置識別符。下列顯示 Sidewalk 裝置的此記錄項目的範例。如需記錄項目中 logLevel 和其他欄位的詳細資訊，請參閱 AWS IoT Wireless 資源和日誌層級。

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "5371db88-d63d-481a-868a-e54b6431845d",
    "wirelessDeviceType": "Sidewalk",
    "event": "Downlink_Data",
    "logLevel": "INFO",
    "messageId": "8da04fa8-037d-4ae9-bf67-35c4bb33da71",
    "message": "Message delivery succeeded.  MessageId: 8da04fa8-037d-4ae9-
bf67-35c4bb33da71. AWS IoT Core: {\"message\":\"OK\",\"traceId\":\"038b5b05-a340-
d18a-150d-d5a578233b09\"}"
}
```

- 如果發生錯誤，您會看到 logLevel 為 ERROR 的記錄項目，而且訊息將包含錯誤的詳細資訊，這將協助您解決此錯誤。Registration 事件何時可能發生錯誤的範例包括：身分驗證問題、無效或太多請求、無法加密或解密承載，或無法使用指定的 ID 找到無線裝置。下列範例顯示在處理訊息時遇到的許可錯誤。

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "cb4c087c-1be5-4990-8654-ccf543ee9fff",
    "wirelessDeviceType": "LoRaWAN",
    "event": "Uplink_Data",
    "logLevel": "ERROR",
    "message": "Cannot assume role MessageId:
 ef38877f-3454-4c99-96ed-5088c1cd8dee.
    Access denied: User: arn:aws:sts::005196538709:assumed-role/
DataRoutingServiceRole/6368b35fd48c445c9a14781b5d5890ed is not authorized
 to perform: sts:AssumeRole on resource: arn:aws:iam::400232685877:role/
ExecuteRules_Role\tstatus code: 403, request id: 471c3e35-f8f3-4e94-b734-
c862f63f4edb"
```

```
}
```

Sidewalk 裝置記錄項目

Sidewalk 裝置的記錄項目可以根據下列事件類型進行分類：

- **`Registration`**

  這些日誌項目會協助您監控正要向 AWS IoT Wireless 註冊之任何 Sidewalk 裝置的狀態。對於此事件類型，如果您在為無線裝置資源設定 CLI 時將日誌層級設定為 `INFO`，則在記錄中，您將看到 `logLevel` 為 `INFO` 和 `ERROR` 的記錄訊息。這些訊息將包含註冊進度 (從開始到完成) 的詳細資訊。`ERROR` 記錄訊息將包含如何疑難排解裝置註冊問題的相關資訊。

  下列顯示日誌層級為 `INFO` 的記錄訊息範例。如需記錄項目中 `logLevel` 和其他欄位的詳細資訊，請參閱 AWS IoT Wireless 資源和日誌層級。

  ```
  {
      "resource": "WirelessDevice",
      "wirelessDeviceId": "8d0b2775-e19b-4b2a-a351-cb8a2734a504",
      "wirelessDeviceType": "Sidewalk",
      "event": "Registration",
      "logLevel": "INFO",
      "message": "Successfully completed device registration. Amazon SidewalkId =
   2000000002"
  }
  ```

- Uplink_Data 與 Downlink_Data

  Sidewalk 裝置的事件類型 `Uplink_Data` 和 `Downlink_Data` 類似於 LoRaWAN 裝置的對應事件類型。如需詳細資訊，請參閱先前針對 LoRaWAN 裝置記錄項目所述的 Uplink_Data 與 Downlink_Data 一節。

後續步驟

您已了解如何檢視資源的日誌項目，以及您在啟用 AWS IoT Wireless 的記錄之後可在 CloudWatch 主控台中檢視的不同日誌項目。儘管您可以使用 Log groups (記錄群組) 來建立篩選串流，但我們建議您使用 CloudWatch Insights 來建立和使用篩選串流。如需詳細資訊，請參閱使用 CloudWatch Insights 來篩選 AWS IoT Wireless 的記錄。

使用 CloudWatch Insights 來篩選 AWS IoT Wireless 的記錄

儘管您可以使用 CloudWatch Logs 來建立篩選表達式，但我們建議您使用 CloudWatch Insights，根據您的應用程式更有效地建立和使用篩選表達式。

建議您首先使用 CloudWatch Log groups (記錄群組)，來了解不同類型的資源、其事件類型，以及您可以用來在主控台中檢視記錄項目的日誌層級。然後，您可以使用此頁面上某些篩選表達式的範例作為參考，為 AWS IoT Wireless 資源建立自己的篩選條件。

在 CloudWatch Logs 洞察主控台中檢視 AWS IoT 記錄

在 CloudWatch 主控台 中，CloudWatch 記錄出現在名為 /aws/iotwireless 的記錄群組中。如需 CloudWatch Logs 的詳細資訊，請參閱 CloudWatch Logs。

在 CloudWatch 主控台中檢視您的 AWS IoT 記錄

導覽至 CloudWatch 主控台，然後在導覽窗格中選擇 Logs Insights (記錄洞察)。

1. 在 Filter (篩選條件) 文字方塊中，輸入 **`/aws/iotwireless`**，然後選擇 `/aws/iotwireless` 記錄洞察。
2. 若要查看記錄群組的完整清單，請選擇 Select log group(s) (選取記錄群組)。若要查看 AWS IoT Wireless 的記錄群組，請選擇 `/aws/iotwireless`。

您現在可以開始輸入查詢以篩選記錄群組。下列各節包含一些實用查詢，將協助您取得有關資源指標的洞察。

建立實用查詢來篩選並取得 AWS IoT Wireless 的洞察

您可以使用篩選表達式，搭配 CloudWatch Insights 來顯示其他實用記錄資訊。下列顯示一些範例查詢：

僅顯示特定資源類型的記錄

您可以建立一個查詢，協助您僅顯示特定資源類型 (例如 LoRaWAN 閘道或 Sidewalk 裝置) 的記錄。例如，若要篩選記錄以僅顯示 Sidewalk 裝置的訊息，您可以輸入下列查詢並選擇 Run query (執行查詢)。若要儲存此查詢，請選擇 Save (儲存)。

```
fields @message
| filter @message like /Sidewalk/
```

在執行查詢之後，您會在 Logs (記錄) 標籤中看到結果，其中顯示與您帳戶中 Sidewalk 裝置相關的記錄時間戳記。如果先前發生與 Sidewalk 裝置相關的事件，您也會看到橫條圖，其中顯示事件發生的時間。如果展開 Logs (日誌) 標籤中的其中一個結果，下列會顯示一個範例。或者，如果想要疑難排解與 Sidewalk 裝置相關的錯誤，您可以新增另一個篩選條件，將日誌層級設定為 ERROR 並僅顯示錯誤資訊。

```
Field           Value
@ingestionTime      1623894967640
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-715adccfb34170214ec2f6667ddfa13cb5af2c3ddfc52fbeee0e554a2e780bed
@message      {
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "Sidewalk",
                    "devEui": "fefffff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Successfully sent downlink message. Amazon SidewalkId =
 2000000006, Sequence number = 0"
                    }
@timestamp          1623894967640
devEui            fefffff000000011a
event         Downlink_Data
logLevel            INFO
message           Successfully sent downlink message. Amazon SidewalkId = 2000000006,
 Sequence number = 0
messageId     7e752a10-28f5-45a5-923f-6fa7133fedda
resource      WirelessDevice
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  Sidewalk
```

顯示特定訊息或事件

您可以建立一個查詢，協助您顯示特定訊息，並在事件發生時進行觀察。例如，如果您想查看何時從 LoRaWAN 無線裝置傳送下行訊息，您可以輸入下列查詢並選擇 Run query (執行查詢)。若要儲存此查詢，請選擇 Save (儲存)。

```
filter @message like /Downlink message sent/
```

在執行查詢之後，您會在 Logs (記錄) 標籤中看到結果，其中顯示下行訊息成功傳送至無線裝置時的時間戳記。您也會看到長條圖，其中顯示下行訊息的傳送時間，如果先前有下行訊息傳送到無線裝置的話。如果展開 Logs (日誌) 標籤中的其中一個結果，下列會顯示一個範例。或者，如果未傳送下行訊息，您可以修改查詢，僅顯示未傳送訊息時的結果，以便您可以偵錯問題。

```
Field           Value
@ingestionTime      1623884043676
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-42d0e6d09ba4d7015f4e9756fcdc616d401cd85fe3ac19854d9fbd866153c872
@message      {
                    "timestamp": "2021-06-16T22:54:00.770493863Z",
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "LoRaWAN",
                    "devEui": "fefffff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Downlink message sent. MessageId:
 7e752a10-28f5-45a5-923f-6fa7133fedda"
                    }
@timestamp          1623884040858
devEui            fefffff000000011a
event         Downlink_Data
logLevel            INFO
message           Downlink message sent. MessageId:
 7e752a10-28f5-45a5-923f-6fa7133fedda
messageId     7e752a10-28f5-45a5-923f-6fa7133fedda
resource      WirelessDevice
timestamp     2021-06-16T22:54:00.770493863Z
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  LoRaWAN
```

後續步驟

您已了解如何使用 CloudWatch Insights，透過建立查詢來篩選記錄訊息，以取得更多實用資訊。您可以結合先前描述的某些篩選條件，並根據您正在監控的資源設計自己的篩選條件。如需使用 CloudWatch Insights 的詳細資訊，請參閱使用 CloudWatch Insights 分析記錄資料。

在使用了 CloudWatch Insights 來建立查詢之後，如果您已儲存這些查詢，則可以視需要載入並執行已儲存的查詢。或者，如果您在 CloudWatch Logs Insights (記錄洞察) 主控台中按一下 History (歷史記錄) 按鈕，則可以檢視先前執行的查詢，並視需要重新執行它們，或建立其他查詢來進一步修改它們。

AWS IoT Wireless 事件通知

AWS IoT Wireless 可以發佈訊息，通知加入 AWS IoT Core 的 LoRaWAN 和 Sidewalk 裝置事件。例如，在佈建或註冊帳戶中的 Sidewalk 裝置時，您可以收到事件的通知。

如何將事件通知到資源

只要發生特定事件，就會發佈事件通知。例如，佈建 Sidewalk 裝置時會產生事件。每個事件都會觸發傳送一則事件通知。事件通知會透過具 JSON 承載的 MQTT 發佈。承載內容取決於事件的類型。

> **Note**
>
> 至少會發佈一次事件通知。事件通知也有可能發佈超過一次。無法保證事件通知的順序。

事件和資源類型

下表顯示您會接收通知的不同類型事件。事件類型取決於資源類型是無線裝置、無線閘道還是 Sidewalk 帳戶。您也可以在資源層級為資源啟用事件，這些事件適用於特定類型的所有資源，或用於選定資源，如下一區段所述。如需不同事件類型的詳細資訊，請參閱 LoRaWAN 資源的事件通知 和 Sidewalk 資源的事件通知。

以資源為基礎的事件類型

資源	資源類型	事件類型	
無線裝置	LoRaWAN	聯結	
	Sidewalk	• 裝置註冊狀態 • 近距離	
無線閘道	LoRaWAN	連線狀態	
Sidewalk 帳戶	Sidewalk	• 裝置註冊狀態 • 近距離	

接收無線事件通知的政策

若要接收事件通知，裝置必須使用適當的政策，以允許該裝置連接至 AWS IoT 裝置閘道，並訂閱 MQTT 事件主題。您也必須訂閱合適的主題篩選條件。

以下為接收各種無線事件通知所需的政策範例。

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "iot:Subscribe",
            "iot:Receive"
        ],
        "Resource":[
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/join/*",
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/
connection_status/*"
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/
device_registration_state/*",
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/proximity/*"
        ]
    }]
}
```

無線事件的 MQTT 主題格式

為了傳送無線資源的事件通知給您，AWS IoT 會使用以貨幣符號 ($) 開頭的 MQTT 預留主題。您可以發佈和訂閱這些預留主題。不過，您無法建立以貨幣符號開頭的新主題。

> **Note**
>
> MQTT 主題轉屬於您的 AWS 帳戶，並且使用格式 `arn:aws:iotwireless:aws-region:AWS-account-ID:topic/Topic`。如需詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 主題。

無線裝置的預留 MQTT 主題使用以下格式：

- 資源層級主題

這些主題適用於您的 AWS 帳戶 中已加入 AWS IoT Wireless 的所有特定類型資源。

```
$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/resources
```

- 識別符層級主題

  這些主題適用於您的 AWS 帳戶 中已加入 AWS IoT Wireless 的選定特定類型資源，由資源識別符指定。

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/
  {resourceIdentifierType}/{resourceID}/{id}
  ```

如需更多關於資源和識別符層級主題的資訊，請參閱 事件組態。

下列資料表顯示各種事件的 MQTT 主題範例：

事件和 MQTT 主題

事件	MQTT 主題	備註

Sideside 裝置註冊狀態	• 資源層級主題
	`$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/wireless_devices`
	• 識別符層級主題
	`$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/{resourceType}/`
		• `{eventType}` 可以是 `registered` 或 `provisioned`
		• `{resourceType}` 可以是 `sidewalk_accounts` 或 `wireless_devices`
		• `{resourceID}` 是 `sidewalk_accounts` 的 `amazon_id` 和 `wireless_devices` 的 `wireless_device_id`

事件	MQTT 主題	備註
	{resourceID}/{id}	
Sidewalk 接近	• 資源層級主題 $aws/iotwireless/events/proximity/{eventType}/sidewalk/wireless_devices • 識別符層級主題 $aws/iotwireless/events/proximity/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}	• {eventType} 可以是 beacon_discovered 或 beacon_lost • {resourceType} 可以是 sidewalk_accounts 或 wireless_devices • {resourceID} 是 sidewalk_accounts 的 amazon_id 和 wireless_devices 的 wireless_device_id

事件	MQTT 主題	備註

LoRaWAN 加入	• 資源層級主題
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices`
	• 識別符層級主題
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices/{resourceID}/{id}`
		• `{eventType}` 可以是 `join_req_0_received` 或 `join_req_2_received` 或 `join_accepted`
		• `{resourceID}` 可以是 `wireless_device_id` 或 `dev_eui`

LoRaWAN 閘道連接狀態	• 資源層級主題
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways`
	• 識別符層級主題
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways/{resourceID}/{id}`
		• `{eventType}` 可以是 `connected` 或 `disconnected`
		• `{resourceID}` 可以是 `wireless_gateway_id` 或 `gateway_eui`

如需不同事件的詳細資訊，請參閱 LoRaWAN 資源的事件通知 和 Sidewalk 資源的事件通知。

如果已經訂閱這些主題，系統會在訊息發佈至其中一個事件通知主題時通知您。如需詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 保留主題。

無線事件的定價

如需訂閱事件和接收通知的定價資訊，請參閱 AWS IoT Core 定價。

為無線資源啟用事件

在預留主題的訂閱者可以接收訊息之前，您必須先啟用事件通知。若要執行此項操作，您可以使用 AWS Management Console 或者 AWS IoT Wireless API 或 AWS CLI。

事件組態

您可以將事件設定為向屬於特定類型的所有資源或個別無線資源寄送通知。資源類型可以是無線閘道、Sidewalk 合作夥伴帳戶或無線裝置，無線裝置可以是 LoRaWAN 或 Sidewalk 裝置。可為無線裝置啟用之事件類型的相關資訊，請參閱 LoRaWAN 資源的事件類型 和 Sideside 資源的事件類型。

所有 資源

您可以啟用事件，讓屬於特定資源類型之 AWS 帳戶 中的所有資源接收通知。例如，您可以啟用一個事件，針對您使用 AWS IoT Core for LoRaWAN 加入的所有 LoRaWAN 閘道，向您通知其連線狀態的變更。監控這些事件將幫助您收到通知，例如資源機群中的某些 LoRaWAN 閘道中斷連線，或者如果您的 AWS 帳戶 中有數個 Sidewalk 裝置信標遺失。

個別資源

您也可以將個別的 LoRaWAN 和 Sidewalk 資源加入您的事件組態，並為其啟用通知。這會幫助您監控特定類型的個別資源。例如，您可以將所選的 LoRaWAN 和 Sidewalk 裝置加入組態，並接收有關這些資源的加入或裝置註冊狀態事件通知。

必要條件

您的 LoRaWAN 或 Sidewalk 資源必須具有允許其接收事件通知的適當政策。如需詳細資訊，請參閱接收無線事件通知的政策。

使用 AWS Management Console 啟用通知

若要從主控台啟用事件訊息，請前往 AWS IoT 主控台的 Settings (設定) 索引標籤，然後前往 LoRaWAN and Sidewalk event notification (LoRaWAN 和 Sidewalk 事件通知) 區段。

您可以針對屬於特定資源類型之 AWS 帳戶 中的所有資源啟用通知，並進行監控。

若要啟用所有資源的通知

1. 在 LoRaWAN and Sidewalk event notification (LoRaWan 和 Sidewalk 事件通知) 區段，前往 All resources (所有資源) 索引標籤，選擇 Action (動作)，然後選擇 Manage events (管理事件)。
2. 啟用您想要監控的事件，然後選擇 Update events (更新事件)。如果您不想再監控某些事件，請選擇 Action (動作) 並選擇 Manage events (管理事件)，然後停用這些事件。

您也可以針對屬於特定資源類型之 AWS 帳戶 中的個別資源啟用通知，並進行監控。

啟用個別資源的通知

1. 在 LoRaWAN and Sidewalk event notification (LoRaWan 和 Sidewalk 事件通知) 區段，選擇 Action (動作)，然後選擇 Add resources (新增資源)。
2. 選擇您要接收通知的資源和事件：

   a. 選擇是否要監控 LoRaWAN resources (LoRaWAN 資源) 或 Sidewalk resources (Sidewalk 資源) 的事件。

   b. 取決於資源類型，您可以選擇要為資源啟用的事件。然後，您可以訂閱這些事件並接收通知。如果選擇：

      - LoRaWAN resources (LoRaWAN 資源)：您可以對您的 LoRaWAN 裝置啟用 join (加入) 事件，或對您的 LoRaWAN 閘道啟用 connection status (連線狀態) 事件。
      - Sidewalk 資源：您可以對您的 Sidewalk 合作夥伴帳戶和 Sidewalk 裝置同時啟用 device registration state (裝置註冊狀態) 或 proximity (接近) 事件。
3. 取決於您選擇的資源類型和事件，選擇要監控的無線裝置或閘道。您可以針對所有資源選擇最多共 250 個資源。
4. 選擇 Submit (提交) 加入您的資源。

您新增的資源會與其 MQTT 主題共同顯示在主控台的 LoRaWAN and Sidewalk event notification (LoRaWan 和 Sidewalk 事件通知) 區段中所屬資源類型的索引標籤。

- LoRaWAN join (LoRaWAN 加入) 事件和 Sidewalk 裝置的事件會顯示在主控台的 Wireless devices (無線裝置) 區段。
- LoRaWAN 閘道的 Connection status (連線狀態) 事件會顯示在 Wireless gateways (無線閘道) 區段。

- Sidewalk 帳戶的 Device registration state (裝置註冊狀態) 和 proximity (接近) 事件會顯示在 Sidewalk accounts (Sidewalk 帳戶) 索引標籤。

使用 MQTT 用戶端訂閱主題

取決於您是為所有資源或個別資源類型啟用事件，您啟用的事件會顯示在主控台，其 MQTT 主題位於 All resources (所有資源) 索引標籤或指定資源類型的索引標籤。

- 如果選擇 MQTT 主題之一，您可以前往 MQTT 用戶端訂閱這些主題並接收訊息。
- 如果加入多個事件，可以訂閱多個事件主題並接收其通知。若要訂閱多個主題，請選擇您的主題，然後選擇 Action (動作)，並選擇 Subscribe (訂閱)。

使用 AWS CLI 啟用通知

您可以設定事件並將資源加入組態，方法是使用 AWS IoT Wireless API 或 AWS CLI。

啟用所有資源的通知

您可以針對 AWS 帳戶 中屬於特定資源類型的所有資源啟用通知，並使用 UpdateEventConfigurationByResourceTypes API 或 update-event-configuration-by-resource-types CLI 命令加以監控。例如：

```
aws iotwireless update-event-configuration-by-resource-types \
   --cli-input-json input.json
```

input.json 的內容

```
{
   "DeviceRegistrationState": {
      "Sidewalk": {
         "AmazonIdEventTopic": "Enabled"
      }
   },
   "ConnectionStatus": {
      "LoRaWAN": {
         "WirelessGatewayEventTopic": "Enabled"
      }
   }
}
```

> **Note**
>
> 所有引號 (") 都會與反斜線 (\) 一起逸出。

可以呼叫 GetEventConfigurationByResourceTypes API 或使用 `get-event-configuration-by-resource-types` CLI 命令來取得目前的事件組態。例如：

```
aws iotwireless get-event-configuration-by-resource-types
```

啟用個別資源的通知

若要透過使用 API 或 CLI 來新增個別資源到事件組態，並控制要發佈的事件類型，請呼叫 UpdateResourceEventConfiguration API 或使用 `update-resource-event-configuration` CLI 命令。例如：

```
aws iotwireless update-resource-event-configuration \
   --identifer 1ffd32c8-8130-4194-96df-622f072a315f \
   --identifier-type WirelessDeviceId \
   --cli-input-json input.json
```

input.json 的內容

```
{
   "Join": {
      "LoRaWAN": {
         "DevEuiEventTopic": "Disabled"
      },
      "WirelessDeviceIdEventTopic": "Enabled"
   }
}
```

> **Note**
>
> 所有引號 (") 都會與反斜線 (\) 一起逸出。

可以呼叫 GetResourceEventConfiguration API 或使用`get-resource-event-configuration` CLI 命令來取得目前的事件組態。例如：

```
aws iotwireless get-resource-event-configuration \
    --identifier-type WirelessDeviceId \
    --identifier 1ffd32c8-8130-4194-96df-622f072a315f
```

列出事件組態

您也可以使用 AWS IoT Wireless API 或 AWS CLI 列出至少啟用一個事件主題的事件組態。若要列出組態，請使用 ListEventConfigurations API 操作或使用 `list-event-configurations` CLI 命令。例如：

```
aws iotwireless list-event-configurations --resource-type WirelessDevice
```

LoRaWAN 資源的事件通知

您可以使用 AWS Management Console 或 AWS IoT Wireless API 操作，通知自己有關 LoRaWAN 裝置和閘道的事件。如需事件通知以及如何啟用事件通知的相關資訊，請參閱 AWS IoT Wireless 事件通知 和 為無線資源啟用事件。

LoRaWAN 資源的事件類型

您可以為 LoRaWAN 資源啟用的事件包括：

- 通知您 LoRaWAN 裝置加入事件的加入事件。裝置加入 AWS IoT Core for LoRaWAN，或者收到類型 0 或類型 2 的重新加入請求時，您會收到通知。
- LoRaWAN 閘道的連線狀態變更為已連線或中斷連線時，連線狀態事件會通知您。

以下部分包含有關 LoRaWAN 資源事件的更多資訊：

主題

- LoRaWAN 加入事件
- 連線狀態事件

LoRaWAN 加入事件

AWS IoT Core for LoRaWAN 可以發佈訊息，通知加入 AWS IoT 的 LoRaWAN 裝置事件。如果收到類型 0 或類型 2 的加入或重新加入請求，並且裝置已使用 AWS IoT Core for LoRaWAN 加入，加入事件會通知您。

加入事件的運作方式

當您使用 AWS IoT Core for LoRaWAN 將您的 LoRaWAN 裝置加入時，AWS IoT Core for LoRaWAN 會對您的裝置以 AWS IoT Core for LoRaWAN 執行加入程序。您的裝置便會啟用以供使用，並且可以傳送一則上行訊息，指示其可供使用。裝置加入後，您的裝置和 AWS IoT Core for LoRaWAN 之間可以交換上行和下行鏈路訊息。如需加入裝置的相關資訊，請參閱 將裝置加入 AWS IoT Core for LoRaWAN。

您可以啟用事件，以在您的裝置加入 AWS IoT Core for LoRaWAN 時通知您。如果加入事件失敗、收到類型 0 或類型 2 的重新加入請求以及接受加入時，您也會收到通知。

啟用 LoRaWAN 加入事件

在 LoRaWAN 加入預留主題的訂閱者可以接收訊息之前，您必須先從 AWS Management Console 或透過使用 API 或 CLI 啟用事件通知。您可以為 AWS 帳戶 中的所有 LoRaWAN 資源或選定資源啟用這些事件。如需如何啟用這些事件的詳細資訊，請參閱 為無線資源啟用事件。

LoRaWAN 事件的 MQTT 主題格式

LoRaWAN 裝置的預留 MQTT 主題使用以下格式。如果您訂閱了這些主題，則所有已註冊到 AWS 帳戶 的 LoRaWAN 裝置可以接收通知：

- 資源層級主題

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices
  ```

- 識別符主題

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices/
  {resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必須是 `join`。

{eventType}

{eventType} 可以是：

- `join_req_received`

- rejoin_req_0_received
- rejoin_req_2_received
- join_accepted

{resourceID}

{resourceID} 可以是 dev_eui 或 wireless_device_id。

例如，您可以訂閱以下主題，以便在 AWS IoT Core for LoRaWAN 接受來自您裝置的加入請求時接收事件通知。

$aws/iotwireless/events/join/join_accepted/lorawan/wireless_devices/wireless_device_id/{id}

您也可以使用 + 萬用字元同時訂閱多個主題。此 + 萬用字元會比對包含該字元的層級中的任何字串，例如下列主題：

$aws/iotwireless/events/join/join_req_received/lorawan/wireless_devices/wireless_device_id/+

> **Note**
>
> 您不能使用萬用字元 # 訂閱預留主題。

如需有關在訂閱主題時使用 + 萬用字元的詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 主題篩選器。

LoRaWAN 加入事件的訊息承載

以下說明 LoRaWAN 加入事件的訊息承載。

```
{
 // General fields
    "eventId": "string",
    "eventType": "join_req_received|rejoin_req_0_received|rejoin_req_2_received|
join_accepted",
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",

 // Event-specific fields
```

```
    "LoRaWAN": {
        "DevEui": "string",

        // The fields below are optional indicating that it can be a null value.
        "DevAddr": "string",
        "JoinEui": "string",
        "AppEui": "string",
    }
}
```

承載包含以下屬性：

eventId

由 AWS IoT Core for LoRaWAN (字串) 產生的唯一事件 ID。

eventType

發生的事件類型。可以是下列其中一個值：

- `join_req_received`：這個欄位會顯示 EUI 參數 `JoinEui` 或 `AppEui`
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`：這個欄位會顯示 `NetId` 和 `DevAddr`。

wirelessDeviceId

LoRaWAN 裝置的 ID。

timestamp

事件發生時的 Unix 時間戳記。

DevEui

在裝置標籤或裝置文件中可找到的裝置唯一識別符。

DevAddr 和 EUI (選填)

這些欄位是選填的裝置位址和 EUI 參數 `JoinEUI` 或 `AppEUI`。

連線狀態事件

AWS IoT Core for LoRaWAN 可以發佈訊息，通知加入 AWS IoT 的 LoRaWAN 閘道連線狀態事件。LoRaWAN 閘道的連線狀態變更為已連線或中斷連線時，連線狀態事件會通知您。

連線狀態事件如何運作

將您的閘道加入 AWS IoT Core for LoRaWAN 後，可以將閘道連線到 AWS IoT Core for LoRaWAN 並驗證其連線狀態。您的閘道連線狀態變更為已連線或中斷連線時，這個事件會通知您。如需更多關於將閘道加入並連接到 AWS IoT Core for LoRaWAN 的資訊，請參閱 將閘道加入 AWS IoT Core for LoRaWAN 和 連接您的 LoRaWAN 閘道並驗證其連線狀態。

LoRaWAN 閘道的 MQTT 主題格式

LoRaWAN 閘道的預留 MQTT 主題使用以下格式。如果您訂閱了這些主題，則所有已註冊到 AWS 帳戶 的 LoRaWAN 閘道可以接收通知：

- 對於資源層級主題：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways
  ```

- 對於識別符主題：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/
  wireless_gateways/{resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必須是 `connection_status`。

{eventType}

{eventType} 可以是 `connected` 或 `disconnected`。

{resourceID}

{resourceID} 可以是 `gateway_eui` 或 `wireless_gateway_id`。

例如，您可以訂閱以下主題，以在所有閘道都已連線 AWS IoT Core for LoRaWAN 時接收事件通知：

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/{id}
```

您也可以使用 + 萬用字元同時訂閱多個主題。此 + 萬用字元會比對包含該字元的層級中的任何字串，例如下列主題：

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/+
```

> **Note**
>
> 您不能使用萬用字元 # 訂閱預留主題。

如需有關在訂閱主題時使用 + 萬用字元的詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 主題篩選器。

連線狀態事件的訊息承載

以下說明連線狀態事件的訊息承載。

```
{
 // General fields
    "eventId": "string",
    "eventType": "connected|disconnected",
    "WirelessGatewayId": "string",
    "timestamp": "timestamp",

 // Event-specific fields
    "LoRaWAN": {
        "GatewayEui": "string"
    }
}
```

承載包含以下屬性：

eventId

由 AWS IoT Core for LoRaWAN (字串) 產生的唯一事件 ID。

eventType

發生的事件類型。可以是 `connected` 或 `disconnected`。

wirelessGatewayId

LoRaWAN 閘道的 ID。

timestamp

事件發生時的 Unix 時間戳記。

GatewayEui

在閘道標籤或閘道文件中可找到的閘道唯一識別符。

Sidewalk 資源的事件通知

您可以使用 AWS Management Console 或 AWS IoT Wireless API 操作，通知自己有關 Sidewalk 裝置和合作夥伴帳戶的事件。如需事件通知以及如何啟用事件通知的相關資訊，請參閱 AWS IoT Wireless 事件通知 和 為無線資源啟用事件。

Sideside 資源的事件類型

您可以為 Sidewalk 資源啟用的事件包括：

- 裝置事件：通知您 Sidewalk 裝置狀態發生變更，例如裝置已註冊且可供使用。
- 接近事件：在 AWS IoT Wireless 接收來自 Amazon Sidewalk 的通知，指出發現或遺失信標時通知您的事件。

以下部分包含有關 Sidewalk 資源事件的更多資訊：

主題

- 裝置註冊狀態事件
- 接近事件

裝置註冊狀態事件

裝置註冊狀態事件會在裝置註冊狀態發生變更時發佈事件通知，例如已佈建或註冊 Sidewalk 裝置時。這些事件會提供關於裝置從佈建到註冊時所經歷的不同狀態資訊。

裝置註冊狀態事件如何運作

在使用 Amazon Sidewalk 和 AWS IoT Wireless 加入 Sidewalk 裝置時，AWS IoT Wireless 會執行 `create` 操作，並將 Sidewalk 裝置新增至 AWS 帳戶。然後裝置會進入已佈建狀態，而 `eventType`

會變成 provisioned。如需加入裝置的相關資訊，請參閱 適用於 AWS IoT Core 的 Amazon Sidewalk 入門。

裝置已完成 provisioned 之後，Amazon Sidewalk 會執行 register 操作，透過 AWS IoT Wireless 註冊 Sidewalk 裝置。註冊程序會啟動，其中加密和工作階段金鑰設定為 AWS IoT。註冊裝置後，eventType 會變成 registered，裝置即可供使用。

裝置 registered 後，Sidewalk 可以傳送請求以 deregister 您的裝置。AWS IoT Wireless 會完成該請求，並將裝置狀態改回 provisioned。如需裝置狀態的詳細資訊，請參閱 DeviceState。

啟用裝置註冊狀態事件通知

在裝置註冊狀態預留主題的訂閱者可以接收訊息之前，您必須先從 AWS Management Console 或透過使用 API 或 CLI 啟用事件通知。您可以為 AWS 帳戶 中的所有 Sidewalk 資源或選定資源啟用這些事件。如需如何啟用這些事件的詳細資訊，請參閱 為無線資源啟用事件。

裝置註冊狀態事件的 MQTT 主題格式

若要通知裝置註冊狀態事件，您可以訂閱以美元 ($) 符號開頭的 MQTT 預留主題。如需詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 主題。

Sidewalk 裝置註冊狀態事件的預留 MQTT 主題使用以下格式：

- 對於資源層級主題：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 對於識別符主題：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必須是 device_registation_state。

{eventType}

{eventType} 可以是 provisioned 或 registered。

{resourceType}

{resourceType} 可以是 `sidewalk_accounts` 或 `wireless_devices`。

{resourceID}

{resourceID} 在 `sidewalk_accounts` 的 {resourceType} 為 `amazon_id`，在 `wireless_devices` 的 {resourceType} 為 `wireless_device_id`。

您也可以使用 + 萬用字元同時訂閱多個主題。此 + 萬用字元會比對包含該字元的層級中的任何字串。例如，如果您想要收到所有可能事件類型 (`provisioned` 和 `registered`) 的通知，以及註冊到特定 Amazon ID 的所有裝置的通知，您可以使用下列主題篩選條件：

```
$aws/iotwireless/events/device_registration_state/+/sidewalk/
sidewalk_accounts/amazon_id/+
```

> **Note**
>
> 您不能使用萬用字元 # 訂閱預留主題。如需有關主題篩選器的詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 主題篩選器。

裝置註冊狀態事件的訊息承載

啟用裝置註冊狀態事件通知之後，事件通知會透過具 JSON 承載的 MQTT 發佈。這些事件包含下方的範例承載：

```
{
    "eventId": "string",
    "eventType": "provisioned|registered",
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",

    // Event-specific fields
    "operation": "create|deregister|register",
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

承載包含以下屬性：

eventId

獨特的事件 ID (字串)。

eventType

發生的事件類型。可以是 `provisioned` 或 `registered`。

wirelessDeviceId

無線裝置的識別符。

timestamp

事件發生時的 Unix 時間戳記。

操作

觸發事件的操作。有效值為 `create`、`register` 和 `deregister`。

sidewalk

要接收事件通知的 Sidewalk Amazon ID 或 `SidewalkManufacturingSn`。

接近事件

接近事件會在 AWS IoT 從 Sidewalk 裝置接收信標時發佈事件通知。當 Sidewalk 裝置接近 Amazon Sidewalk 時，Amazon Sidewalk 會定期篩選從裝置傳送的信標，並由 AWS IoT Wireless 接收。AWS IoT Wireless 會在收到信標時通知您這些事件。

接近事件的運作方式

接近事件會在 AWS IoT 接收到信標時發出通知，Sidewalk 裝置可以隨時發出信標。當裝置位於 Amazon Sidewalk 附近時，Sidewalk 會收到信標並會按固定的時間間隔將信標轉送到 AWS IoT Wireless。Amazon Sidewalk 已將此時間間隔設定為 10 分鐘。在 AWS IoT Wireless 接收來自 Sidewalk 的信標時，您會收到事件的通知。

當發現信標或信標遺失時，接近事件會通知您。您可以設定接收接近事件通知的間隔。

啟用接近事件的通知

在 Sidewalk 接近預留主題的訂閱者可以接收訊息之前，您必須先從 AWS Management Console 或透過使用 API 或 CLI 啟用事件通知。您可以為 AWS 帳戶 中的所有 Sidewalk 資源或選定資源啟用這些事件。如需如何啟用這些事件的詳細資訊，請參閱 為無線資源啟用事件。

接近事件的 MQTT 主題格式

若要通知自己接近事件，您可以訂閱以美元 ($) 符號開頭的 MQTT 預留主題。如需詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 主題。

Sidewalk 接近事件的預留 MQTT 主題使用以下格式：

- 對於資源層級主題：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 對於識別符主題：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必須是 `proximity`。

{eventType}

{eventType} 可以是 `beacon_discovered` 或 `beacon_lost`。

{resourceType}

{resourceType} 可以是 `sidewalk_accounts` 或 `wireless_devices`。

{resourceID}

{resourceID} 在 `sidewalk_accounts` 的 {resourceType} 為 `amazon_id`，在 `wireless_devices` 的 {resourceType} 為 `wireless_device_id`。

您也可以使用 + 萬用字元同時訂閱多個主題。此 + 萬用字元會比對包含該字元的層級中的任何字串。例如，如果您想要收到所有可能事件類型 (`beacon_discovered` 和 `beacon_lost`) 的通知，以及註冊到特定 Amazon ID 的所有裝置的通知，您可以使用下列主題篩選條件：

```
$aws/iotwireless/events/proximity/+/sidewalk/sidewalk_accounts/amazon_id/+
```

> **Note**
>
> 您不能使用萬用字元 # 訂閱預留主題。如需有關主題篩選器的詳細資訊，請參閱《AWS IoT 開發人員指南》中的 MQTT 主題篩選器。

接近事件的訊息承載

啟用接近事件通知之後，事件訊息會透過具 JSON 承載的 MQTT 發佈。這些事件包含下方的範例承載：

```
{
    "eventId": "string",
    "eventType": "beacon_discovered|beacon_lost",
    "WirelessDeviceId": "string",
    "timestamp": "1234567890123",

    // Event-specific fields
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

承載包含以下屬性：

eventId

唯一的事件 ID，這是一個字串。

eventType

發生的事件類型。可以是 `beacon_discovered` 或 `beacon_lost`。

WirelessDeviceId

無線裝置的識別符。

timestamp

事件發生時的 Unix 時間戳記。

sidewalk

要接收事件通知的 Sidewalk Amazon ID 或 `SidewalkManufacturingSn`。

AWS IoT Wireless API 作業

您可在加入 LoRaWAN 或 Sidewalk 終端裝置時，或建立用於大量佈建 Sidewalk 終端裝置的匯入任務時，執行下列其他 API 操作。

下列各節包含有關這些 API 操作的其他資訊。

主題

裝置設定檔的 AWS IoT Wireless API 操作

您可為 LoRaWAN 和 Sidewalk 裝置設定檔執行下列 API 操作：

- `CreateDeviceProfile` API 或 `create-device-profile` CLI
- `GetDeviceProfile` API 或 `get-device-profile` CLI
- `ListDeviceProfiles` API 或 `list-device-profiles` CLI
- `DeleteDeviceProfile` API 或 `delete-device-profile` CLI

下列章節為您展示如何列出和刪除設定檔。如需建立和擷取裝置設定檔的相關資訊，請參閱：

- 新增裝置設定檔
- 步驟 1：建立裝置設定檔

列出您 AWS 帳戶 中的裝置設定檔

您可使用 `ListDeviceProfiles` API 操作，列出您新增至 AWS IoT Wireless 之 AWS 帳戶 中的裝置設定檔。您可以使用此資訊來識別要與此設定檔建立關聯的裝置。

如要篩選清單，僅顯示 LoRaWAN 或 Sidewalk 裝置設定檔，請在執行 API 時設定 Type。下列顯示命令範例 CLI 命令：

```
aws iotwireless list-device-profiles --wireless-device-type "Sidewalk"
```

執行此命令會傳回您新增的裝置設定檔清單，包括其設定檔識別碼和 Amazon Resource Name (ARN)。如要擷取有關特定設定檔的其他詳細資訊，請使用 GetDeviceProfile API。

```
{
    "DeviceProfileList": [
        {
            "Name": "SidewalkDeviceProfile1",
            "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d"
        },
        {
            "Name": "SidewalkDeviceProfile2",
            "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/
a1b2c3d4-5678-90ab-cdef-12ab345c67de"
        }
    ]
}
```

從您的 AWS 帳戶 刪除裝置設定檔

您可使用 DeleteDeviceProfile API 操作刪除裝置設定檔。下列顯示範例 CLI 命令：

> **Warning**
>
> 刪除動作無法還原。該裝置設定檔將從您的AWS 帳戶 永久移除。

```
aws iotwireless delete-device-profile --name "SidewalkProfile"
```

此命令不會產生任何輸出。您可使用 GetDeviceProfile API 或 ListDeviceProfiles API 操作，來驗證設定檔是否已從您的帳戶中移除。

LoRaWAN 和 Sidewalk 裝置的 AWS IoT Wireless API 操作

您可為 LoRaWAN 和 Sidewalk 裝置執行下列 API 操作：

- CreateWirelessDevice API 或 create-wireless-device CLI
- GetWirelessDevice API 或 get-wireless-device CLI
- ListWirelessDevices API 或 list-wireless-devices CLI
- DeleteWirelessDevice API 或 delete-wireless-device CLI
- UpdateWirelessDevice API 或 update-wireless-device CLI
- AssociateWirelessDeviceWithThing API 或 associate-wireless-device-with-thing CLI
- DisassociateWirelessDeviceFromThing API 或 disassociate-wireless-device-from-thing CLI

下列章節為您展示如何列出和刪除裝置。如需建立無線裝置和擷取裝置資訊的相關資訊，請參閱：

- 將您的無線裝置新增至 AWS IoT Core for LoRaWAN
- 步驟 2：新增您的 Sidewalk 裝置

將您 AWS 帳戶 中的無線裝置與 IoT 物件建立關聯

若要將您的 LoRaWAN 和 Sidewalk 裝置與 AWS IoT 物件建立關聯，請使用 AssociateWirelessDeviceWithThing API 操作。

AWS IoT 中的物件可讓您更輕鬆地搜尋並管理您的裝置。將物件與您的裝置相關連可讓該裝置存取其他 AWS IoT Core 功能。如需使用此 API 的詳細資訊，請參閱 AssociateWirelessDeviceWithThing。

下列顯示執行此命令的範例。執行這個命令不會產生任何輸出。

```
aws iotwireless associate-wireless-device-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MySidewalkThing"
```

如要取消您的無線裝置與 AWS IoT 物件的關聯，請使用 DisassociateWirelessDeviceFromThing API 操作，如下列範例所示。

```
aws iotwireless disassociate-wireless-device-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

列出您 AWS 帳戶 中的無線裝置

如要列出您新增至 AWS IoT Wireless 之 AWS 帳戶 中的無線裝置，請使用 ListWirelessDevices API 操作。如要篩選清單，以便僅傳回 LoRaWAN 或 Sidewalk 裝置，請設定 WirelessDeviceType。

下列顯示執行此命令的範例。

```
aws iotwireless list-wireless-devices --wireless-device-type Sidewalk
```

執行此命令會傳回您新增的裝置清單，包括其設定檔識別碼和 Amazon Resource Name (ARN)。如要擷取有關特定裝置的其他詳細資訊，請使用 GetWirelessDevice API 操作。

```
{
    "WirelessDeviceList": [
        {
            "Name": "mySidewalkDevice",
            "DestinationName": "SidewalkDestination",
            "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
            "Type": "Sidewalk",
            "Sidewalk": {
                "SidewalkId": "1234567890123456"
            },
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f"
        }
    ]
}
```

從您的 AWS 帳戶 刪除無線裝置

如要刪除無線裝置，請將您要刪除之裝置的 WirelessDeviceID 傳遞至 DeleteWirelessDevice API 操作。

下列顯示範例命令：

```
aws iotwireless delete-wireless-device --id "23456789-abcd-0123-bcde-fabc012345678"
```

此命令不會產生任何輸出。您可使用 GetWirelessDevice API 或 ListWirelessDevices API 操作，來驗證裝置是否已從您的帳戶中移除。

無線裝置的目的地的 AWS IoT Wireless API 操作

您可為 LoRaWAN 和 Sidewalk 裝置的目的地執行下列 API 操作：

- CreateDestination API 或 create-destination CLI
- GetDestination API 或 get-destination CLI
- UpdateDestination API 或 update-destination CLI
- ListDestinations API 或 list-destinations CLI
- DeleteDestination API 或 delete-destination CLI

下列各節展示如何取得、列出、更新及和刪除目的地。如需建立目的地的相關資訊，請參閱 為您的 Sidewalk 終端裝置新增目的地。

取得目的地的相關資訊

您可使用 GetDestination API 操作，以取得您新增至 AWS IoT Wireless 帳戶中的目的地相關資訊。提供目的地名稱作為 API 的輸入。API 接著將會傳回與所指定識別符相符之目的地的資訊。

下列顯示範例 CLI 命令：

```
aws iotwireless get-destination --name SidewalkDestination
```

執行此命令會傳回您目的地的參數。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "SidewalkDestination",
    "Expression": "IoTWirelessRule",
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

更新您目的地的屬性

使用 UpdateDestination API 操作，來更新您為 AWS IoT Wireless 新增至帳戶的目的地屬性。下列顯示了更新描述屬性的範例 CLI 命令：

```
aws iotwireless update-destination --name SidewalkDestination \
    --description "Destination for messages processed using IoTWirelessRule"
```

列出 AWS 帳戶 中的目的地

使用 ListDestinations API 操作，在 AWS 帳戶 中列出您新增至 AWS IoT Wireless 的目的地。如要篩選清單，以便僅傳回 LoRaWAN 和 Sidewalk 終端裝置的目的地，請使用 WirelessDeviceType 參數。

下列顯示範例 CLI 命令：

```
aws iotwireless list-destinations --wireless-device-type "Sidewalk"
```

執行此命令會傳回您新增的目的地清單，包括其 Amazon Resource Name (ARN)。如要擷取有關特定目的地的其他詳細資訊，請使用 GetDestination API。

```
{
    "DestinationList": [
        {
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/IoTWirelessDestination",
            "Name": "IoTWirelessDestination",
            "Expression": "IoTWirelessRule",
            "Description": "Destination for messages processed using IoTWirelessRule",
            "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
        },
        {
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/IoTWirelessDestination2",
            "Name": "IoTWirelessDestination2",
            "Expression": "IoTWirelessRule2",
            "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
        }
    ]
}
```

從您的 AWS 帳戶 刪除目的地

如要刪除目的地，請將要刪除的目的地名稱作為輸入傳遞給 DeleteDestination API 操作。下列顯示範例 CLI 命令：

> **Warning**
>
> 刪除動作無法還原。該目的地將從您的AWS 帳戶 永久移除。

```
aws iotwireless delete-destination --name "SidewalkDestination"
```

此命令不會產生任何輸出。您可使用 GetDestination API 或 ListDestinations API 操作來驗證目的地是否已從您的帳戶中移除。

用於大量佈建的 AWS IoT Core for Amazon Sidewalk API 操作

您可執行下列 API 操作，以大量佈建 Sidewalk 終端裝置：

- StartWirelessDeviceImportTask API 或 start-wireless-device-import-task CLI
- StartSingleWirelessDeviceImportTask API 或 start-single-wireless-device-import-task CLI
- ListWirelessDeviceImportTasks API 或 list-wireless-device-import-tasks CLI
- ListDevicesForWirelessDeviceImportTask API 或 list-devices-for-wireless-device-import-task CLI
- GetWirelessDeviceImportTask API 或 get-wireless-device-import-task CLI
- UpdateWirelessDeviceImportTask API 或 update-wireless-device-import-task CLI
- DeleteWirelessDeviceImportTask API 或 delete-wireless-device-import-task CLI

下列各節展示如何取得、列出、更新及和刪除匯入任務。如需建立匯入任務的相關資訊，請參閱 用於大量佈建的 AWS IoT Core for Amazon Sidewalk API 操作。

取得匯入任務的資訊

您可以使用 ListDevicesForWirelessDeviceImportTask API 操作來擷取特定匯入任務的相關資訊，及該任務中裝置的加入狀態。作為 API 操作的輸入，請指定您從 StartWirelessDeviceImportTask 或 StartSingleWirelessDeviceImportTask API 操作取得的匯入任務 ID。API 接著將會傳回與所指定識別符相符之匯入任務的資訊。

下列顯示範例 CLI 命令：

```
aws iotwireless list-devices-for-wireless-device-import-task --id e2a5995e-743b-41f2-
a1e4-3ca6a5c5249f
```

執行此命令會傳回您的匯入任務資訊和裝置加入狀態。

```
{
   "DestinationName": "SidewalkDestination",
   "ImportedWirelessDeviceList": [
      {
         "Sidewalk": {
            "OnboardingStatus": "ONBOARDED",
            "LastUpdateTime": "2023-02021T06:11:09.151Z",
            "SidewalkManufacturingSn":
 "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A4A"
         },
         "Sidewalk": {
             "OnboardingStatus": "PENDING",
             "LastUpdateTime": "2023-02021T06:22:12.061Z",
             "SidewalkManufacturingSn":
 "12345ABCDE6789FABDESBDEF123456789012345FEABC0123679AFEBC01234EF"
         },
      }
   ]
}
```

取得匯入任務裝置摘要

如要取得新增至特定匯入任務之裝置加入狀態的摘要資訊計數，請使用 GetWirelessDeviceImportTask API 操作。下列顯示範例 CLI 命令。

```
aws iotwireless get-wireless-device-import-task --Id "e2a5995e-743b-41f2-
a1e4-3ca6a5c5249f"
```

下列程式碼顯示來自命令的範例回應。

```
{
   "NumberOfFailedImportedDevices": 2,
   "NumberOfOnboardedImportedDevices": 4,
   "NumberOfPendingImportedDevices": 1
}
```

新增裝置以匯入任務

使用 UpdateWirelessDeviceImportTask API 操作，將裝置新增至您新增的現有匯入任務。您可使用此 API 操作，來新增先前未包含您使用 StartWirelessDeviceImportTask API 操作建立之任務的裝置序號 (SMSN)。

如要將裝置附加到匯入任務，做為 API 請求的一部分，請於 Amazon S3 儲存貯體中指定一個新的 CSV 檔案，其中包含要新增的裝置序號。僅當目前位於匯入任務中的裝置尚未啟動加入程序時，才會接受此請求。若上線程序已經開始，則 UpdateWirelessDeviceImportTask API 請求將會失敗。

若您仍想將裝置附加至匯入任務，則可再次執行 UpdateWirelessDeviceImportTask API 操作。在執行此 API 操作之前，第一個 UpdateWirelessDeviceImportTask API 請求必須已完成處理 S3 儲存貯體中的 CSV 檔案。

> **Note**
>
> 當您執行 ListImportedWirelessDeviceTasks API 請求時，目前不會傳回使用 UpdateWirelessDeviceImportTask API 操作所指定新 CSV 檔案的 S3 URL。相反地，API 操作會傳回最初使用 StartWirelessDeviceImportTask API 請求傳送之請求的 S3 URL。

下列顯示範例 CLI 命令。

```
aws iotwireless update-wireless-device-import task \
    --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f" \
    --sidewalk '{"FileForCreateDevices": "s3://import_task_bucket/import_file3"}'
```

列出您 AWS 帳戶 中的匯入任務

使用 ListWirelessDeviceImportTasks API 或 list-imported-wireless-device-tasks CLI 命令，列出您 AWS 帳戶 中的匯入任務。下列顯示範例 CLI 命令。

```
aws iotwireless list-wireless-device-import-tasks
```

執行此命令會傳回您所建立的匯入任務清單。此清單包括其 Amazon S3 CSV 檔案和指定的 IAM 角色、匯入任務 ID，及裝置加入狀態的摘要資訊。

```
{
   "ImportWirelessDeviceTaskList": [
      {
         "FileForCreateDevices": "s3://import_task_bucket/import_file1",
         "ImportTaskId": "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f",
         "NumberOfFailedImportedDevices": 1,
         "NumberOfOnboardedImportedDevices": 3,
         "NumberOfPendingImportedDevices": 2,
         "Role": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI",
         "TimeStamp": "1012202218:23:55"
      },
      {
         "FileForCreateDevices": "s3://import_task_bucket/import_file2",
         "ImportTaskId": "a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a",
         "NumberOfFailedImportedDevices": 2,
         "NumberOfOnboardedImportedDevices": 4,
         "NumberOfPendingImportedDevices": 1,
         "Role": "arn:aws:iam::123456789012:role/service-role/CDEFaBC1",
         "TimeStamp": "1201202210:12:20"
      }
   ]
}
```

從您 AWS 帳戶 中刪除匯入任務

如要刪除匯入任務，請將匯入任務 ID 傳遞至 DeleteWirelessDeviceImportTask API 操作或 delete-wireless-device-import-task CLI 命令。

> **Warning**
>
> 刪除動作無法還原。該匯入任務將從您的 AWS 帳戶 永久移除。

當您執行 DeleteWirelessDeviceImportTask API 請求時，背景程序會開始刪除匯入任務。當請求正在進行時，匯入任務中的裝置序號 (SMSN) 正在刪除的過程中。只有在完成刪除後，您才可使用 ListImportedWirelessDeviceTasks 或 GetImportedWirelessDeviceTasks API 操作查看此資訊。

若匯入任務仍包含正在等待加入的裝置，則只僅於匯入任務中的所有裝置皆已加入或無法加入後，才會處理 DeleteWirelessDeviceImportTask API 請求。匯入任務會在 90 天後到期，一旦任務過期，則可從您的帳戶中將其刪除。不過，使用匯入任務成功加入的裝置將不會遭到刪除。

> **Note**
>
> 若您嘗試建立另一個匯入任務，其中包含使用 DeleteWirelessDeviceImportTask API 請求等待刪除之裝置的序號，則 StartWirelessDeviceImportTask API 操作將會傳回錯誤。

下列顯示範例 CLI 命令：

```
aws iotwireless delete-import-task --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

此命令不會產生任何輸出。於任務刪除後，如要確認匯入任務已從您的帳戶中移除，您可使用 GetWirelessDeviceImportTask API 操作或 ListWirelessDeviceImportTasks API 操作。

使用 AWS CloudFormation 建立 AWS IoT Wireless 資源

AWS IoT Wireless 與 AWS CloudFormation 進行了整合，該服務可協助您建立模型和設定 AWS 資源，藉此減少您建立和管理資源和基礎設施所花的時間。您建立一個範本來描述您需要的所有 AWS 資源，並 AWS CloudFormation 負責為您供應和配置這些資源。

使用 AWS CloudFormation 時，您可以重複使用範本，以便一致地重複設定 AWS IoT Wireless 資源。只需描述一次您的資源，即可在多個 AWS 帳戶 與區域內重複佈建相同的資源。

AWS IoT Wireless 和 AWS CloudFormation 範本

若要佈建和設定 AWS IoT Wireless 與相關服務的資源，您必須了解 AWS CloudFormation 範本。範本是以 JSON 或 YAML 格式化的文本檔案。而您亦可以透過這些範本的說明，了解欲在 AWS CloudFormation 堆疊中佈建的資源。如果您不熟悉 JSON 或 YAML，您可以使用 AWS CloudFormation 設計器協助您開始使用 AWS CloudFormation 範本。如需更多詳細資訊，請參閱 AWS CloudFormation 使用者指南 中的 什麼是 AWS CloudFormation 設計器？。

AWS IoT Wireless 支援在 AWS CloudFormation 中建立您的無線資源。如需詳細資訊 (包括用於 AWS IoT Wireless 資源的 JSON 和 YAML 範本範例)，請參閱《AWS CloudFormation 使用者指南》中的 AWS IoT Wireless 資源類型參考。

進一步了解 AWS CloudFormation

如需進一步了解 AWS CloudFormation，請參閱下列資源：

- AWS CloudFormation
- 《AWS CloudFormation 使用者指南》https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html
- AWS CloudFormation 命令列介面使用者指南

AWS IoT Wireless 的配額

對於每項 AWS 服務 服務，您的 AWS 帳戶 有預設配額，先前稱為限額。除非另有說明，否則每個配額都是區域特定規定。您可以請求提高某些配額，而其他配額無法提高。

若要檢視 AWS IoT Wireless 的配額，請開啟 Service Quotas 主控台。在導覽窗格中，選擇 AWS 服務 s，然後選取 AWS IoT Wireless。

若要請求增加配額，請參閱 Service Quotas 使用者指南中的請求提高配額。如果 Service Quotas 中尚未提供配額，請使用增加服務配額表單。

AWS IoT Wireless 擁有下列項目的配額：

- AWS IoT Core for LoRaWAN 配額可用於在裝置之間傳輸的裝置資料
- AWS IoT Wireless API 操作適用於 LoRaWAN 和 Sidewalk 裝置。

如需詳細資訊，請參閱《AWS 一般參考》中的 AWS IoT Core for LoRaWAN 配額。

標記您的 AWS IoT Wireless 資源

為了協助您管理和組織裝置、閘道、目的地及設定檔，您可以選擇使用標籤將自己的中繼資料指派給每個資源。本節說明標籤並示範如何建立標籤。AWS IoT Wireless 沒有帳單群組，且與 AWS IoT Core 使用相同的帳單群組。如需詳細資訊，請參閱《AWS IoT Core 文件》中的帳單群組。

標籤基本概念

若您已有數個相同類型的 AWS IoT Wireless 資源，可以使用標籤透過不同方式來分類資源 (例如依用途、擁有者或環境)。這樣做可協助您根據指派給資源的標籤來快速識別資源。

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，您可以針對要更新裝置韌體的一個 LoRaWAN 裝置群組定義一組標籤。為了更輕鬆地管理您的資源，我們建議您建立一組一致的標籤鍵，以滿足您對每種資源類型的需求。

您可以根據新增或套用的標籤來搜尋與篩選資源。您也可以使用標籤來控制資源的存取，藉由使用 IAM 政策和帳單群組標籤來分類和追蹤成本。

建立和管理標籤

您可以使用 AWS Management Console 中的標籤編輯器、AWS IoT Wireless 或 AWS CLI 來建立和管理標籤

使用主控台

為了方便使用起見，AWS Management Console 中的標籤編輯器提供了集中、統一的方式，讓您建立和管理標籤。如需詳細資訊，請參閱使用 AWS Management Console 中的使用標籤編輯器。

使用 API 或 CLI

您也可以使用 API 或 CLI，並在建立無線裝置、閘道、設定檔和目的地時，使用下列命令中的 `Tags` 欄位將標籤與其產生關聯：

- AssociateAwsAccountWithPartnerAccount
- CreateDestination
- CreateDeviceProfile

- CreateFuotaTask
- CreateMulticastGroup
- CreateServiceProfile
- CreateWirelessGateway
- CreateWirelessGatewayTaskDefinition
- CreateWirelessDevice
- API_StartBulkAssociateWirelessDeviceWithMulticastGroup

更新資源的標籤或列出標籤

您可以使用下列命令新增、修改或刪除支援標記功能的現有資源標籤：

- TagResource
- ListTagsForResource
- UntagResource

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。

標籤的限制與上限

以下基本限制適用於標籤：

- 每個資源的標籤數上限：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。
- 標籤鍵與值皆區分大小寫。
- 請勿於標籤名稱或值中使用 `aws:` 字首。保留以供日後 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。允許使用的字元包括可用 UTF-8 表示的英文字母、空格和數字，以及以下特殊字元：+ - = . _ : / @。

搭配 IAM 政策使用標籤

若要指定使用者可建立、修改或使用的資源，您可以在用於 AWS IoT Wireless API 動作的 IAM 政策中，套用以標籤為基礎的資源層級許可。若要根據資源的標籤控制使用者存取 (許可)，可使用 `Condition` 元素 (也稱為 `Condition` 區塊)，以及 IAM 政策中的以下條件內容金鑰和值。

- 使用 `aws:ResourceTag/`*`tag-key`*`: `*`tag-value`* 以允許或拒絕資源上具有特定標籤的使用者動作。
- 使用 `aws:RequestTag/`*`tag-key`*`: `*`tag-value`* 以在提出 API 請求時，要求使用 (或不使用) 特定標籤，以建立或修改允許標籤的資源。
- 使用 `aws:TagKeys: [`*`tag-key`*`, ...]` 以在提出 API 請求時，要求使用 (或不使用) 特定標籤金鑰集，以建立或修改允許標籤的資源。

> **Note**
>
> IAM 政策中的條件內容金鑰和值，只會套用到資源識別符可標記為必要參數的那些 AWS IoT 動作。例如，根據條件內容金鑰和值，不允許或拒絕使用 DescribeEndpoint，因為在此請求中所參照的項目沒有可標記資源。

如需使用標籤的詳細資訊，請參閱《AWS Identity and Access Management 使用者指南》中的使用標籤控制。該指南的 IAM JSON 政策參考章節有詳細的語法、說明，還有元素、變數範例，以及在 IAM 中的 JSON 政策評估邏輯。

以下範例政策會套用兩個以標籤為基礎的限制。受到此政策限制的 IAM 使用者：

- 無法提供資源「env=prod」標籤 (在範例中，請參閱此行 `"aws:RequestTag/env" : "prod"`)。
- 無法修改或存取具有現有標籤 "env = prod" 的資源 (在範例中，請參閱此行 `"aws:ResourceTag/env" : "prod"`)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:CreateMulticastGroup",
      "Resource": "*",
```

```
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/env": "prod"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "iot:CreateMulticastGroup",
        "iot:UpdateMulticastGroup",
        "iot:GetMulticastGroup",
        "iot:ListMulticastGroups"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/env": "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:CreateMulticastGroup",
        "iot:UpdateMulticastGroup",
        "iot:GetMulticastGroup",
        "iot:ListMulticastGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

您也可以透過將其包含在清單中，為特定標籤金鑰指定多個標籤值，如下所示：

```
"StringEquals" : {
              "aws:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> 如果您允許或拒絕使用者根據標籤存取資源，請務必考慮明確拒絕使用者將這些標籤新增至相同資源或從中移除的能力。否則，使用者可能透過修改標籤來避開您的限制，並取得資源的存取。

AWS IoT Wireless 使用者指南的文件歷程記錄

下表說明 AWS IoT Wireless 的文件發行版本。

變更	描述	日期
初始版本	AWS IoT Wireless 使用者指南的初始發行版本	2020 年 12 月 31 日