aws

开发人员指南

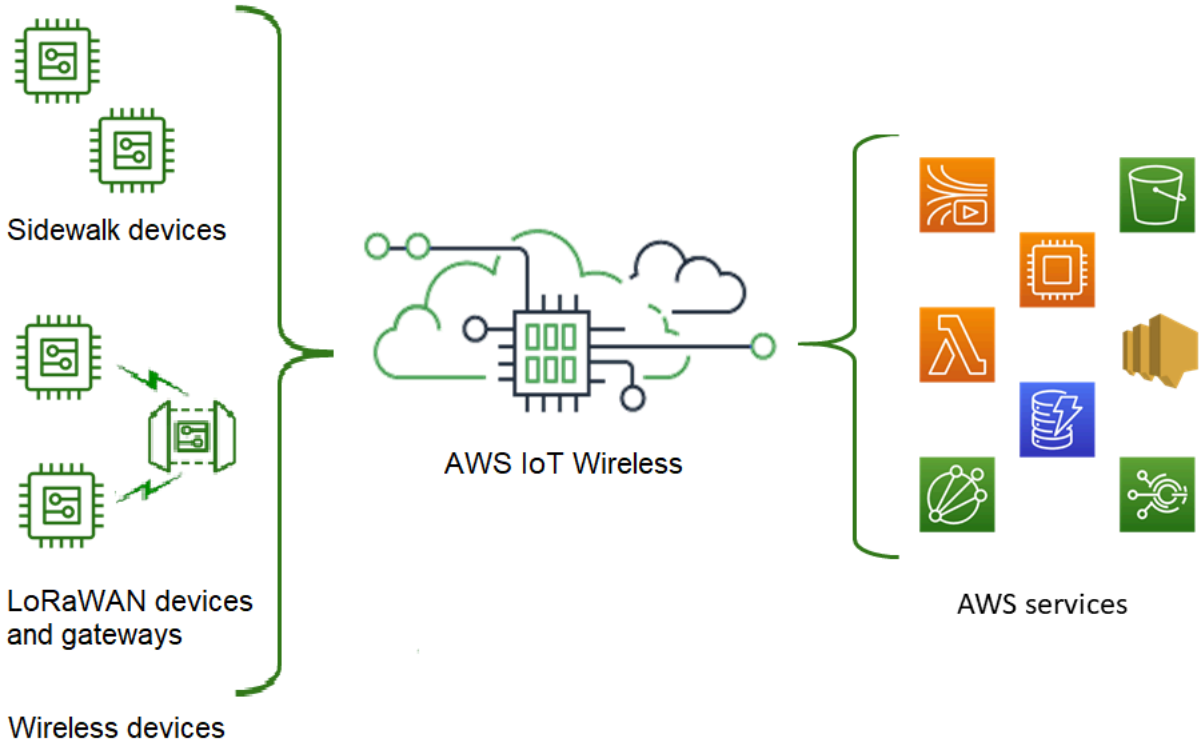
AWS IoT Wireless

AWS IoT Wireless: 开发人员指南

Table of Contents

什么是 AWS IoT Wireless？

AWS IoT Wireless 提供云服务，以将您的无线设备连接到其他设备和 AWS Cloud服务。通过将设备连接到 AWS IoT Wireless，您可以将设备集成到基于 AWS IoT 的解决方案。您可以使用 AWS IoT Wireless 将 LoRaWAN 和 Sidewalk 设备登记到 AWS IoT。这些无线设备使用低功耗广域网（LPWAN）通信协议来与 AWS IoT 进行通信。



AWS IoT Wireless 的功能

AWS IoT Wireless 提供以下功能：

登记 LoRaWAN 和 Sidewalk 设备

您可以将 LoRaWAN 和 Sidewalk 设备登记到 AWS IoT Wireless。

- 适用于 LoRaWAN 的 AWS IoT Core

  要将您的 LoRaWAN 设备和网关登记到 AWS IoT Wireless，请使用 适用于 LoRaWAN 的 AWS IoT Core。它是一个完全托管的 LoRaWAN 网络服务器（LNS），您无需设置和运行私有 LNS。适用于 LoRaWAN 的 AWS IoT Core 使用配置和更新服务器（CUPS）和无线固件更新（FUOTA）功能提供网关管理。有关更多信息，请参阅什么是 适用于 LoRaWAN 的 AWS IoT Core？。

- 适用于 Amazon Sidewalk 的 AWS IoT Core

  要将 Sidewalk 设备登记到 AWS IoT Wireless，可以使用适用于 Amazon Sidewalk 的 AWS IoT Core 提供的功能。Amazon Sidewalk 是一个共享网络，可连接 Amazon Echo、RingSecurity Cams 和户外灯等设备，并且可以支持您的社区中的其他 Sidewalk 设备。有关更多信息，请参阅什么是适用于 Amazon Sidewalk 的 AWS IoT Core？。

与 AWS IoT Core 集成

通过将 AWS IoT Wireless 与 AWS IoT Core 集成，您可以使用以下功能：

- 将设备与 AWS IoT 事物关联

  您可以将无线设备和网关与 AWS IoT 事物 进行关联，从而帮助您将设备的表示存储到云端。您可以使用 AWS IoT 中的事物更轻松地搜索和管理您的设备以及访问其他 AWS IoT Core 功能。有关更多信息，请参阅《AWS IoT Core Developer Guide》中的 Managing devices with AWS IoT。
- 使用 AWS IoT 规则路由消息

  您可以使用 AWS IoT 的规则功能与其他 AWS 服务和应用程序进行交互。从您的设备发送到云端的上行链路消息可以路由到这些服务和其他应用程序。有关更多信息，请参阅《AWS IoT Core Developer Guide》 中的 AWS IoT rules。

首次接触 AWS IoT Wireless 的用户

如果您是首次接触 AWS IoT Wireless 的用户，建议您先阅读以下各节：

- 什么是 适用于 LoRaWAN 的 AWS IoT Core？

  本节概述了 LoRaWAN 技术和 适用于 LoRaWAN 的 AWS IoT Core 的工作原理。它还提供相关资源来帮助您了解更多信息。
- 什么是适用于 Amazon Sidewalk 的 AWS IoT Core？

  本节概述了 Amazon Sidewalk 技术以及适用于 Amazon Sidewalk 的 AWS IoT Core 的工作原理。它还提供相关资源来帮助您了解更多信息。
- 开始使用适用于 Amazon Sidewalk 的 AWS IoT Core

  阅读本节，了解如何使用适用于 Amazon Sidewalk 的 AWS IoT Core 以及如何登记 Amazon Sidewalk 设备。

- 将网关和设备连接到 适用于 LoRaWAN 的 AWS IoT Core

  接下来，您可以详细了解如何使用控制台和 API 登记 LoRaWAN 设备。

相关服务

- Amazon CloudWatch

  将 LoRaWAN 或 Sidewalk 设备登记到 AWS IoT Wireless 后，您可以使用 Amazon CloudWatch 实时记录和监控您的无线设备和网关。要监控 LoRaWAN 设备和网关，您还可以使用网络分析器，以便显著缩短建立连接和开始接收跟踪消息所需的时间。
- AWS IoT Core

  您还可以使用 AWS IoT Core 集成连接到可从规则引擎访问的 AWS 服务。有关更多信息，请参阅AWS 服务s used by the rules engine。

访问 AWS IoT Wireless

您可以使用控制台、API 或 CLI 来登记 LoRaWAN 和 Sidewalk 设备。

- 使用 AWS IoT控制台

  要登记无线设备，请使用 AWS Management Console 中的 AWS IoT Wireless 页面。
- 使用 AWS IoT Wireless API

  您可以通过使用 AWS IoT Wireless API 登记 Sidewalk 和 LoRaWAN 设备。用于构建 AWS IoT Core 的 AWS IoT Wireless API 由 AWS SDK 提供支持。有关更多信息，请参阅 AWS SDK 和工具包。
- 使用 AWS CLI

  您可以使用 AWS CLI 运行用于登记和管理 LoRaWAN 及 Amazon Sidewalk 设备的命令。有关更多信息，请参阅 AWS IoT Wireless CLI 参考。

开始使用 AWS IoT Wireless

您可以通过注册 AWS 账户以及按照相关步骤创建 IAM 用户，来开始使用 AWS IoT Wireless。注册后，您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI，登记 Sidewalk 和 LoRaWAN 设备及网关。登记设备时，请考虑如何描述和标记您的资源，以帮助您更轻松地识别它们。

以下主题将向您展示如何开始使用 AWS IoT Wireless。

主题

设置 AWS IoT Wireless

在注册 AWS 时，将在 AWS 中为您的 AWS 账户 自动注册所有服务，包括 AWS IoT Wireless。您只需为使用的服务付费。

要设置 AWS IoT Wireless，请使用以下各节中的步骤：

主题

设置您的 AWS 账户

首次使用 适用于 LoRaWAN 的 AWS IoT Core 或适用于 Amazon Sidewalk 的 AWS IoT Core 之前，请完成以下任务以设置 AWS 账户。

主题

注册 AWS 账户

如果您没有AWS 账户，请完成以下步骤来创建一个。

要注册AWS 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

   注册AWS 账户时，系统将会创建AWS 账户根用户。根用户有权访问该账户中的所有AWS 服务和资源。作为安全最佳实践，请为管理用户分配管理访问权限，并且只使用根用户执行需要根用户访问权限的任务。

创建 IAM 用户

要创建管理员用户，请选择以下选项之一。

选择一种方法来管理您的管理员	目的	方式	您也可以
在 IAM Identity Center 中（建议）	使用短期凭证访问 AWS。 这符合安全最佳实操。有关最佳实践的信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。	有关说明，请参阅《AWS IAM Identity Center 用户指南》中的入门。	按照《AWS Command Line Interface 用户指南》中的配置 AWS CLI 以使用 AWS IAM Identity Center，配置编程式访问。
在 IAM 中（不推荐使用）	使用长期凭证访问 AWS。	按照《IAM 用户指南》中的创建您的首个 IAM 管理员用户和组的说明操作。	按照《IAM 用户指南》中的管理 IAM 用户的访问密钥，配置编程式访问。

作为 IAM 用户登录

创建 IAM 用户后，您可以使用 IAM 用户名和密码登录 AWS。

在以 IAM 用户身份登录之前，您可以在 IAM 控制台中验证 IAM 用户的登录链接。在 IAM 控制面板的 IAM 用户登录链接下，您可以看到您的 AWS 账户 的登录链接。您的登录链接的 URL 包含您的 AWS 账户 ID（不含破折号 (-)）。

如果您不希望登录链接的 URL 包含 AWS 账户 ID，可以创建账户别名。有关更多信息，请参阅《IAM 用户指南》中的创建、删除和列出 AWS 账户别名。

以 IAM 用户身份登录

1. 注销 AWS Management Console。
2. 输入登录链接，其中包括您的 AWS 账户 ID（不含破折号）或您的 AWS 账户 别名：

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

3. 输入您刚创建的 IAM 用户名和密码。

登录后，导航栏将显示*“your_user_name @ your_aws_account_id”*。

安装 Python 和 AWS CLI

在连接 LoRaWAN 或 Sidewalk 终端设备之前，必须设置 Python 安装和配置 AWS CLI。

> **Important**
>
> 要执行用于预置和注册 Sidewalk 终端设备的整个登记工作流，您还必须设置 Sidewalk 网关和 HDK。有关说明，请参阅《Amazon Sidewalk 文档》中的设置硬件开发工具包（HDK）和设置 Sidewalk 网关。

主题

- 安装 Python 和 Python3-pip
- 设置 AWS CLI

安装 Python 和 Python3-pip

要使用下一节中描述的 AWS CLI 和 boto3，必须使用 Python 3.6 或更高版本。如果要使用 AWS IoT 控制台登记终端设备，可以跳过本节并继续设置您的 AWS 账户。要检查您是否已经安装了 Python 和 Python3-Pip，请运行以下命令。如果运行这些命令返回了版本，则表示 Python 和 Python3-Pip 已正确安装。

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

如果此命令返回错误，可能是因为没有安装 Python，或者操作系统将 Python v3.x 可执行文件调用为 Python3。在这种情况下，请在运行命令时将 `python` 的所有实例替换为 `python3`。如果仍然出现错误，请下载并运行 Python 安装程序，或者根据您的操作系统安装 Python，如下所述。

Windows

在 Windows 计算机上，从 Python 网站下载 Python，然后运行安装程序来在您的计算机上安装 Python。

Linux

在 Ubuntu 计算机上，运行以下 `sudo` 命令来安装 Python。

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

macOS

在 Mac 计算机上，使用 Homebrew 安装 Python。Homebrew 还会安装 pip，后者指向已安装的 Python3 版本。

```
$ brew install python
```

设置 AWS CLI

以下步骤向您演示如何配置 AWS CLI 和 boto3（AWS SDK for Python）。在执行这些步骤之前，您必须先注册 AWS 账户并创建管理用户。有关说明，请参阅设置 AWS IoT Wireless。

1. 安装和配置 AWS CLI

   您可以使用 AWS CLI 以编程方式将 Sidewalk 终端设备登记到适用于 Amazon Sidewalk 的 AWS IoT Core。如果您想使用 AWS IoT 控制台登记设备，可以跳过本节。打开 AWS IoT Core 控制台，然后继续下一节，开始将设备连接到适用于 Amazon Sidewalk 的 AWS IoT Core。有关配置 AWS CLI 的说明，请参阅安装和配置 AWS CLI。

2. 安装 boto3（适用于 Python 的 AWS SDK）

   以下命令向您展示如何安装 boto3（适用于 Python 的 AWS SDK）和 AWS CLI。您还安装 botocore，这是运行 boto3 所必需的。有关详细说明，请参阅《Boto3 文档指南》https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#installation中的安装 Boto3。

   > **Note**
   >
   > `awscli` 版本 `1.26.6` 要求 PyYAML 版本为 3.10 或更高版本，但不得高于 5.5。

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

3. 配置您的凭证和默认区域

   在 `~/.aws/credentials` 和 `~/.aws/config` 文件中配置您的凭证和默认区域。boto3 库使用这些凭证来识别您的 AWS 账户并授权 API 调用。有关配置说明，请参阅：

   - 《Boto3 文档指南》中的配置
   - 《AWS CLI 文档指南》中的配置和凭证文件设置

描述您的 AWS IoT Wireless 资源

在开始登记 LoRaWAN 或 Sidewalk 设备之前，请考虑设备、网关和目标的命名约定。AWS IoT Wireless 提供了多种选项来帮助您标识您所创建的资源。尽管 AWS IoT Wireless 资源会在创建时提供一个唯一的 ID，但这个 ID 不是描述性的，也不能在创建资源后进行更改。为了更方便地选择、识别和管理资源，您可以指定名称、添加描述、将标签和标签值附加到大多数 AWS IoT Wireless 资源。

- 资源名称和描述

对于设备、网关和配置文件，资源名称是一个可选字段，您可以在创建资源后更改此字段。该名称将出现在资源中心页面上显示的列表中。

对于目标，您需要在您的AWS账户和AWS 区域中提供一个唯一的名称。创建目标资源之后无法更改目标名称。

虽然名称最多可包含 256 个字符，但资源中心中的显示空间有限。如果可能，请确保名称的前 20 到 30 个字符即可区分不同名称。

- 资源标签

  标签是能够附加到 AWS 资源上的键/值对（元数据）。您可以同时选择标签键及其相应的值。

  网关、目标和配置文件最多可以附加 50 个标签。设备不支持附加标签。

资源名称和描述

对名称的 AWS IoT Wireless 资源支持

资源	名称字段支持
目标	名称是资源的唯一 ID，无法更改。
无线设备	名称是资源的可选描述符，可以更改。
LoRaWAN 网关	名称是资源的可选描述符，可以更改。
配置文件	名称是资源的可选描述符，可以更改。

名称字段显示在资源中心的资源列表中；但是，由于空间有限，因此只能显示名称的前 15-30 个字符。为资源选择名称时，请考虑您希望它们如何识别资源以及如何在控制台中显示。

描述

目标、设备和网关资源还支持描述字段，该字段最多可接受 2048 个字符。描述字段仅显示在单个资源的详细信息页面中。虽然描述字段可以保存大量信息，但由于它只显示在资源的详细信息页面中，因此在多个资源的上下文中扫描并不方便。

资源标签

对 AWS 标签的 AWS IoT Wireless 资源支持

资源	AWS 标签支持
目标	可向资源中最多添加 50 个 AWS 标签。
无线设备	此资源不支持 AWS 标签。
LoRaWAN 网关	可向资源中最多添加 50 个 AWS 标签。
配置文件	可向资源中最多添加 50 个 AWS 标签。

标签是可作为元数据标识和组织 AWS 资源的词或短语。您可以将标签键视为信息类别，将标签值视为该类别中的特定值。例如，您的标记值可能为 color，然后对于一些资源的这个标签，您可以给予 blue 值，对于其他资源，则给予 red 值。有了这个，您可以使用AWS控制台中的 Tag Editor 来查找 color 标签值为 blue 的资源。

有关在 AWS IoT Wireless 中标记的更多信息，请参阅 标记您的 AWS IoT Wireless 资源。

有关标记和标记策略的更多信息，请参阅标签编辑器。

适用于 LoRaWAN 的 AWS IoT Core

适用于 LoRaWAN 的 AWS IoT Core 是一个完全托管的 LoRaWAN 网络服务器（LNS），它使用配置和更新服务器（CUPS）以及无线固件更新（FUOTA）功能提供网关管理。您可以用 适用于 LoRaWAN 的 AWS IoT Core 替换私有 LNS 并将远距离广域网（LoRaWAN）设备和网关连接到 AWS IoT Core。通过这样做，您将减少维护、运营成本、设置时间和开销成本。

> **Note**
>
> 适用于 LoRaWAN 的 AWS IoT Core 仅支持 IPv4 地址格式。它不支持 IPv6 或双堆栈配置（IPv4 和 IPv6）。有关更多信息，请参阅支持 IPv6 的 AWS 服务。

简介

LoRaWAN 设备是远程、低功耗、电池供电的设备，使用 LoRaWAN 协议在免许可的无线电频谱中运行。LoRaWAN 是基于 LoRa 构建的低功耗广域网 (LPWAN) 通信协议。LoRa 是实现设备之间低功耗、广域通信的物理层协议。

要将 LoRaWAN 设备连接到 AWS IoT，您必须使用 LoRaWAN 网关。网关充当桥梁，将您的设备连接到 适用于 LoRaWAN 的 AWS IoT Core 并交换消息。适用于 LoRaWAN 的 AWS IoT Core 使用 AWS IoT 规则引擎将消息从您的 LoRaWAN 设备路由到其他 AWS IoT 服务。

为了减少开发工作并将设备快速登记到 适用于 LoRaWAN 的 AWS IoT Core，建议您使用 LoRaWAN 认证的终端设备。有关更多信息，请参阅 适用于 LoRaWAN 的 AWS IoT Core 产品概述页。有关获得设备 LoRaWAN 认证的信息，请参阅认证 LoRaWAN 产品。

访问 适用于 LoRaWAN 的 AWS IoT Core

通过使用控制台或 AWS IoT Wireless API，您可以将 LoRaWAN 设备和网关快速登记到 适用于 LoRaWAN 的 AWS IoT Core。

使用控制台

要使用 AWS Management Console登记 LoRaWAN 设备和网关，请登录到 AWS Management Console然后在 AWS IoT 控制台中导航到 适用于 LoRaWAN 的 AWS IoT Core 页面。然后，您可以使

用简介部分将网关和设备添加到 适用于 LoRaWAN 的 AWS IoT Core。有关更多信息，请参阅使用控制台将您的设备和网关登记到 适用于 LoRaWAN 的 AWS IoT Core。

使用 API 或 CLI

您可以通过使用 AWS IoT Wireless API 登记 LoRaWAN 和 Sidewalk 设备。用于构建 适用于 LoRaWAN 的 AWS IoT Core 的 AWS IoT Wireless API 由 AWS SDK 提供支持。有关更多信息，请参阅 AWS SDK 和工具包。

您可以使用 AWS CLI 运行用于登记和管理 LoRaWAN 网关和设备的命令。有关更多信息，请参阅 AWS IoT Wireless CLI 参考。

适用于 LoRaWAN 的 AWS IoT Core 区域和端点

适用于 LoRaWAN 的 AWS IoT Core 可为特定于您所在 AWS 区域的控制面板和数据面板 API 端点提供支持。数据面板 API 端点特定于您的 AWS 账户和 AWS 区域。有关 适用于 LoRaWAN 的 AWS IoT Core Jobs 端点的更多信息，请参阅《AWS General Reference》中的 适用于 LoRaWAN 的 AWS IoT Core Endpoints。

为了在设备和 AWS IoT 之间进行更安全的通信，您可以在虚拟私有云（VPC）中通过 AWS PrivateLink 将设备连接到 适用于 LoRaWAN 的 AWS IoT Core，而不是通过公共网络进行连接。有关更多信息，请参阅适用于 LoRaWAN 的 AWS IoT Core 和接口 VPC 端点（AWS PrivateLink）。

适用于 LoRaWAN 的 AWS IoT Core 具有适用于在设备之间传输的设备数据的配额和 AWS IoT Wireless API 操作的最大 TPS。有关更多信息，请参阅《AWS General Reference》中的 适用于 LoRaWAN 的 AWS IoT Core quotas。

适用于 LoRaWAN 的 AWS IoT Core 定价

如果您是新客户，那么在注册 AWS 后，可以通过 AWS 免费套餐开始免费使用 适用于 LoRaWAN 的 AWS IoT Core。对于 适用于 LoRaWAN 的 AWS IoT Core，您只需按实际使用量付费。有关一般产品概述和定价的更多信息，请参阅 AWS IoT Core定价。

什么是 适用于 LoRaWAN 的 AWS IoT Core？

通过将您的 LoRaWAN 设备和网关连接到 AWS，适用于 LoRaWAN 的 AWS IoT Core 将替换私有 LoRaWAN 网络服务器（LNS）。使用 AWS IoT 规则引擎，您可以路由从 LoRaWAN 设备接收的消息，这些消息可以在那里格式化并发送到其他 AWS IoT 服务。为了保护与 AWS IoT 的设备通信，适用于 LoRaWAN 的 AWS IoT Core 使用 X.509 证书。

适用于 LoRaWAN 的 AWS IoT Core 可管理 AWS IoT Core 与 LoRaWAN 网关和设备进行通信所需的服务和设备策略。适用于 LoRaWAN 的 AWS IoT Core 还管理描述将设备数据发送到其他服务的 AWS IoT 规则的目标。

适用于 LoRaWAN 的 AWS IoT Core 的功能

通过使用 适用于 LoRaWAN 的 AWS IoT Core，您可以：

- 登记 LoRaWAN 设备和网关并将其连接到 AWS IoT，无需设置和管理私有 LNS。
- 连接符合 1.0.x 或 1.1 标准或 LoRa Alliance 标准化的 LoRaWAN 规范的 LoRaWAN 设备。这些设备可以在 A 类、B 类或 C 类模式下运行。
- 使用支持 LoRa 基本站点版本 2.0.4 或更高版本的 LoRaWAN网关。支持 适用于 LoRaWAN 的 AWS IoT Core 运行 LoRa 基本站点兼容版本的所有网关。
- 使用公开的 LoRaWAN 网络将您的 LoRaWAN 设备连接到云端，以缩短部署时间，并且无需管理私有 LoRaWAN 网络，从而节省时间和成本。
- 使用 适用于 LoRaWAN 的 AWS IoT Core 的自适应数据速率监控信号强度、带宽和扩展因子，并根据需要优化数据速率。您还可以使用网络分析器实时监控 LoRaWAN 资源。
- 使用 CUPS 服务更新 LoRaWAN 网关的固件，并使用无线固件更新 (FUOTA) 更新 LoRaWAN 设备的固件。

以下主题将提供有关 LoRaWAN 技术和 适用于 LoRaWAN 的 AWS IoT Core 的更多信息。

主题

什么是 LoRaWAN？

LoRa Alliance 将 LoRaWAN 描述为“一种低功耗、广域 (LPWA) 网络协议，旨在将电池供电的“事物”无线连接到区域、国家或全球网络中的互联网，并满足物联网 (IoT) 的关键要求，如双向通信、端到端安全、移动性和本地化服务。"。

LoRa 和 LoRaWAN

LoRaWAN 协议是在 LoRA 上运行的低功耗广域网 (LPWAN) 通信协议。

LoRaWAN 已被公认为是适用于低功耗广域网的国际标准。要了解更多信息，请参阅 LoRAWAN formally recognized as ITU internationl standard。LoRaWAN 规范是开放的，因此任何人都可以设置和操作 LoRa 网络。

LoRa 是一种无线音频技术，在无许可证的射频频谱中运行。LoRa 是一种物理层协议，使用扩频调制并支持远程通信，但以窄带宽为代价。它使用具有中央频率的窄带波形发送数据，从而使其能够抵御干扰。

LoRaWAN 技术的特点

- 远距离通信可达 10 英里的视线
- 电池续航时间长达 10 年。为了延长电池续航时间，您可以在 A 类或 B 类模式下操作设备，这需要增加下行链路延迟。
- 设备和维护成本低。
- 无许可证的无线电频谱，但适用于特定区域的规定。
- 低功耗，但有限的负载大小为 51 字节到 241 字节，具体取决于数据速率。数据速率可以是 0.3 Kbit/s — 27 Kbit/s 的数据速率，最大有效负载大小为 222。

LoRaWAN 协议版本

LoRa Alliance 使用 LoRaWAN 规范文件指定了 LoRaWAN 协议。考虑到特定地区的法规，LoRa Alliance 还发布了区域参数文件。有关更多信息，请参阅 LoRaWAN regional parameters and specifications。

LoRaWAN 的初始版本为 1.0。发布的其他版本包括 1.0.1、1.0.2、1.0.3、1.0.4 和 1.1。1.0.1-1.0.4 版本通常称为 1.0.x。

了解有关 LoRaWAN 的更多信息

以下链接包含有关 LoRaWAN 技术和 LoRa Basics Station 的有用信息，这是在 LoRaWAN 网关上运行的软件，用于将终端设备连接到 适用于 LoRaWAN 的 AWS IoT Core。

- LoRaWAN 被国际电联认定为国际标准

  LoRaWAN 已被国际电联正式记录为适用于低功耗广域网的国际标准。该标准名为 ITU-T Y.4480 建议书“广域无线网络的低功耗协议”。
- LoRaWAN 的事物基础

LoRaWAN 的事物基础知识包含了一个介绍视频，涵盖了 LoRaWAN 的基础知识，以及一系列章节内容，可帮助您了解 LoRa 和 LoRaWAN。

- 什么是 LoRaWAN

  LoRa Alliance 提供了 LoRa 和 LoRaWAN 的技术概览，包括不同区域的 LoRaWAN 规范摘要。

- LoRa Basics Station

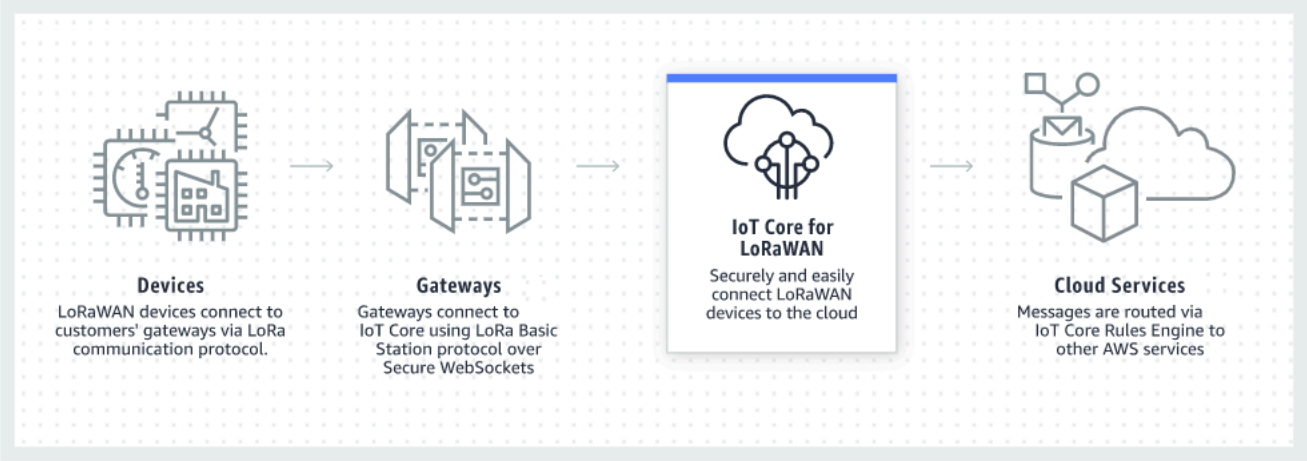
  Semtech Corporation 为网关和端点提供了有关 LoRa 基础知识的有用概念。LoRa Basics Station 是一种在您的 LoRaWAN 网关上运行的开源软件，通过 Semtech Corporation 的 GitHub 存储库进行维护和分发。您还可以了解描述如何交换 LoRaWAN 数据和执行配置更新的 LNS 和 CUPS 协议。

- LoRaWAN 区域参数和规范

  RP002-1.0.2 文档包括对 LoRaWAN 第 2 层规范的所有版本的支持。它包括有关 LoRaWAN 规范和区域参数以及不同 LoRaWAN 版本的信息。

适用于 LoRaWAN 的 AWS IoT Core 的工作原理

LoRaWAN 网络架构部署在 星形拓扑中，网关在终端设备和 LoRaWAN 网络服务器 (LNS) 之间中继信息。下面显示了 LoRaWAN 设备如何与 适用于 LoRaWAN 的 AWS IoT Core 进行交互。还显示 适用于 LoRaWAN 的 AWS IoT Core 如何充当 LNS 并与 AWS Cloud中的其他 AWS 服务进行通信。



LoRaWAN 设备通过 LoRaWAN 网关与 AWS IoT Core 进行通信。适用于 LoRaWAN 的 AWS IoT Core 可管理 AWS IoT Core 用于管理 LoRaWAN 网关和设备并与其进行通信所需的服务和设备策略。适用于 LoRaWAN 的 AWS IoT Core 还管理描述将设备数据发送到其他服务的 AWS IoT 规则的目标。

开始使用 适用于 LoRaWAN 的 AWS IoT Core

以下步骤概述了如何开始使用 适用于 LoRaWAN 的 AWS IoT Core。

1. 选择您需要的无线设备和 LoRaWAN 网关

   AWS 合作伙伴设备目录包含网关和开发人员工具包，这些网关和开发人员工具包可与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用。有关更多信息，请参阅使用来自AWS Partner Device Catalog 的合格网关。

2. 将您的无线设备和 LoRaWAN 网关添加到 适用于 LoRaWAN 的 AWS IoT Core

   将网关和设备连接到 适用于 LoRaWAN 的 AWS IoT Core 为您提供有关如何描述资源以及如何将无线设备和 LoRaWAN 网关添加到 适用于 LoRaWAN 的 AWS IoT Core 的信息。您还将了解如何配置其他 适用于 LoRaWAN 的 AWS IoT Core 资源，以便管理这些设备并将其数据发送到 AWS 服务。

3. 完善您的 适用于 LoRaWAN 的 AWS IoT Core 解决方案。

   开始使用我们的 适用于 LoRaWAN 的 AWS IoT Core 解决方案示例并使其为您所用。

适用于 LoRaWAN 的 AWS IoT Core 资源

以下资源将帮助您进一步了解 适用于 LoRaWAN 的 AWS IoT Core 以及如何开始使用它。

- 开始使用 适用于 LoRaWAN 的 AWS IoT Core

  下面的视频介绍了 适用于 LoRaWAN 的 AWS IoT Core 的工作原理，并指导您完成从 AWS Management Console 添加 LoRaWAN 网关的整个过程。

- 适用于 LoRaWAN 的 AWS IoT Core 研讨会

  该研讨会涵盖 LoRaWAN 技术基础知识及其 适用于 LoRaWAN 的 AWS IoT Core 实现。您还可以利用研讨会演练实验，以便演示如何将网关和设备连接到 适用于 LoRaWAN 的 AWS IoT Core，以构建 IoT 解决方案示例。

- 借助 AWS IoT 实施低功耗广域网（LPWAN）解决方案

  本白皮书为您提供决策框架，帮助您决定 LPWAN 是否是物联网用例的正确选择，概述 LPWAN 连接技术及其功能，并提供实施指南。

将网关和设备连接到 适用于 LoRaWAN 的 AWS IoT Core

适用于 LoRaWAN 的 AWS IoT Core 将帮助您连接和管理无线 LoRaWAN（低功耗远距离广域网）设备，并且无需开发和操作 LNS。远距离 WAN (LoRaWAN) 设备和网关可以使用 适用于 LoRaWAN 的 AWS IoT Core 连接到 AWS IoT Core。

设备、网关、配置文件和目标的命名约定

在您开始使用 适用于 LoRaWAN 的 AWS IoT Core 并创建资源之前，请考虑设备、网关和目标的命名约定。

适用于 LoRaWAN 的 AWS IoT Core 会为您为无线设备、网关和配置文件创建的资源分配唯一 ID；但是，您也可以为资源指定更具描述性的名称，以便更轻松地识别它们。在将设备、网关、配置文件和目标添加到 适用于 LoRaWAN 的 AWS IoT Core 时，请考虑如何命名以便更轻松地管理。

您可以将标签添加到创建的资源中。在添加 LoRaWAN 设备之前，请考虑如何使用标签来识别和管理 适用于 LoRaWAN 的 AWS IoT Core 资源。添加标签后，可以对其进行修改。

有关命名和标记资源的更多信息，请参阅 描述您的 AWS IoT Wireless 资源。

将设备数据映射到服务数据

LoRaWAN 无线设备的数据通常被编码以优化带宽。这些编码的消息将以不容易被其他AWS服务使用的格式抵达 适用于 LoRaWAN 的 AWS IoT Core。适用于 LoRaWAN 的 AWS IoT Core 将使用 AWS IoT 规则，可以借助 AWS Lambda 函数来处理设备消息并将其解码为其他AWS服务可以使用的格式。

转换设备数据并将其发送到其他AWS服务，您需要知道：

- 无线设备发送的数据格式和内容。
- 您想要向其发送数据的服务。
- 服务所需的格式。

使用该信息，您可以创建 AWS IoT 规则，该规则执行转换并将转换后的数据发送到将使用该数据的AWS服务。

使用控制台将您的设备和网关登记到 适用于 LoRaWAN 的 AWS IoT Core

您可以使用控制台界面或 API 添加 LoRaWAN 网关和设备。如果您第一次使用的是 适用于 LoRaWAN 的 AWS IoT Core，我们建议您使用控制台。控制台界面在一次性管理多个 适用于 LoRaWAN 的 AWS

IoT Core 资源时最为实用。当管理大量 适用于 LoRaWAN 的 AWS IoT Core 资源时，请考虑使用 AWS IoT Wireless API 创建更为自动化的解决方案。

配置 适用于 LoRaWAN 的 AWS IoT Core 资源时输入的大部分数据由设备供应商提供，并且特定于他们支持的 LoRaWAN 规范。以下主题将介绍如何描述您适用于 LoRaWAN 的 AWS IoT Core，并使用控制台或 API 添加您的网关和设备。

> **Note**
>
> 如果您使用公共网络将 LoRaWAN 设备连接到云端，则可以跳过网关登记步骤。有关更多信息，请参阅管理来自公共 LoRaWAN 设备网络（Everynet）的 LoRaWAN 流量。

主题

将您的网关登记到 适用于 LoRaWAN 的 AWS IoT Core

如果您是首次使用 适用于 LoRaWAN 的 AWS IoT Core，您可以使用控制台添加您的第一个 LoRaWan 网关和设备。

> **Note**
>
> 如果您使用公共网络将 LoRaWAN 设备连接到云端，则可以跳过网关登记步骤。有关更多信息，请参阅管理来自公共 LoRaWAN 设备网络（Everynet）的 LoRaWAN 流量。

登记网关之前

将您的网关登记到 适用于 LoRaWAN 的 AWS IoT Core 之前，我们建议您：

- 使用能够与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用的网关。这些网关将连接到 AWS IoT Core，无需进行任何其他配置设置，并且 2.0.4 或更高版本的 LoRa Basics Station 软件将在上面运行。有关更多信息，请参阅使用 AWS IoT Wireless 管理网关。
- 请考虑您创建的资源的命名约定，以便更轻松地管理这些资源。有关更多信息，请参阅描述您的 AWS IoT Wireless 资源。

- 预先准备好每个网关独有的配置参数，使数据输入控制台更加顺畅。AWS IoT 需要用来与网关进行通信和管理的无线网关配置参数包括网关的 EUI 及其 LoRa 频带。

如要登记您的网关以便与 适用于 LoRaWAN 的 AWS IoT Core 通信：

- 考虑频段选择并添加必要的 IAM 角色
- 将网关添加到 适用于 LoRaWAN 的 AWS IoT Core
- 连接您的 LoRaWAN 网关并验证其连接状态

考虑频段选择并添加必要的 IAM 角色

在将网关添加到 适用于 LoRaWAN 的 AWS IoT Core 之前，我们建议您考虑网关运行的频段，并添加必要的 IAM 角色，以便将网关连接到 适用于 LoRaWAN 的 AWS IoT Core。

> **Note**
>
> 如果您使用控制台添加网关，请在控制台中单击 Create role（创建角色）创建必要的 IAM 角色，以便跳过这些步骤。仅当您使用 CLI 创建网关时，才需要执行这些步骤。

为您的网关和设备连接选择 LoRa 频带

适用于 LoRaWAN 的 AWS IoT Core 支持 EU863-870、US902-928、AU915 和 AS923-1 频段，您可以使用这些频段连接实际存在于支持这些频段的频率范围和特性的国家/地区的网关和设备。EU863-870 频段和 US902-928 频段通常分别用于欧洲和北美。AS923-1 频段常用于澳大利亚、新西兰、日本和新加坡等国家。AU915 用于澳大利亚和阿根廷等国家。有关在您所在地区或国家/地区使用哪些频段的更多信息，请参阅 LoRaWAN® 区域参数。

LoRa Alliance 发布 LoRaWAN 规范和区域参数文档，可从 LoRa Alliance 网站下载。LoRa Alliance 的区域参数可帮助公司决定在其所在地区或国家使用哪个频段。适用于 LoRaWAN 的 AWS IoT Core 的频段执行操作遵循了区域参数规范文档中的建议。这些区域参数被分组为一组无线电参数，以及适应工业、科学和医疗 (ISM) 波段的频率分配。我们建议您与合规团队合作，以确保您符合任何适用的法规要求。

添加 IAM 角色以允许 Configuration and Update Server (CUPS) 管理网关凭证

此流程介绍如何添加允许 Configuration and Update Server (CUPS) 管理网关凭证的 IAM 角色。请确保在 LoRaWAN 网关尝试连接到 适用于 LoRaWAN 的 AWS IoT Core 前执行此流程；不过您只需进行一次该操作。

添加 IAM 角色以允许 Configuration and Update Server (CUPS) 管理网关凭证

1. 开启 IAM 控制台的角色中心，然后选择 Create role（创建角色）。
2. 如果您认为您可能已经添加了 IoTWirelessGatewayCertManagerRole 角色，请在搜索栏中输入 **`IoTWirelessGatewayCertManagerRole`**。

   如果您在搜索结果中看到了 IoTWirelessGatewayCertManagerRole 角色，则您拥有了必要的 IAM 角色。您现在可以离开流程。

   如果搜索结果为空，则您还没有必要的 IAM 角色。继续执行流程以添加该角色。
3. 在 Select type of trusted entity（选择受信任实体的类型）下，选择 Another AWS 账户（另一个亚马逊云科技账户）。
4. 在 Account ID（账户 ID），请输入您的AWS 账户 ID，然后选择 Next: Permission（下一步：权限）。
5. 在搜索框中，输入 **`AWSIoTWirelessGatewayCertManager`**。
6. 在搜索结果中，选择名为 AWSIoTWirelessGatewayCertManager 的策略。
7. 依次选择 Next: Tags（下一步：标签）和 Next: Review（下一步：查看）。
8. 对于 Role name（角色名称），请输入 **`IoTWirelessGatewayCertManagerRole`**，然后选择 Create role（创建角色）。
9. 要编辑新角色，请在确认消息中，选择 IoTWirelessGatewayCertManagerRole。
10. 在 Summary（摘要）页面上，选择 Trust relationships（信任关系），然后选择 Edit trust relationship（编辑信任关系）。
11. 在 Policy Document（策略文档）中，更改 `Principal` 属性以使其类似于此示例。

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

在您更改 `Principal` 属性后，完整的策略文档应该如此示例所示。

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

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

12. 要保存更改并退出，请选择 Update Trust Policy（更新信任策略）。

您现在已经创建了 IoTWirelessGatewayCertManagerRole。今后便无需再次执行此操作了。

如果您在添加网关时执行了此流程，则可以关闭此窗口和 IAM 控制台，然后返回 AWS IoT 控制台以完成网关的添加。

将网关添加到 适用于 LoRaWAN 的 AWS IoT Core

您可以私用控制台或 CLI 将网关添加到 适用于 LoRaWAN 的 AWS IoT Core。

添加您的网关之前，我们建议您考虑将您的网关登记到 适用于 LoRaWAN 的 AWS IoT Core的登记网关之前部分中提到的各项因素。

如果您是第一次添加网关，我们建议您使用控制台。如果要改为使用 CLI 添加网关，您必须已创建必要的 IAM 角色，以便网关可以连接到 适用于 LoRaWAN 的 AWS IoT Core。有关如何创建此角色的信息，请参阅 添加 IAM 角色以允许 Configuration and Update Server (CUPS) 管理网关凭证。

使用控制台添加网关

导航到 AWS IoT 控制台的 适用于 LoRaWAN 的 AWS IoT Core 简介页面并选择 Get started（开始使用），然后选择 Add gateway（添加网关）。如果您已添加网关，请选择 View gateway（查看网关）以查看您添加的网关。如果要添加更多网关，请选择 Add gateway（添加网关）。

1. 提供网关详细信息和频带信息

   使用 Gateway details（网关详细信息）部分提供有关设备配置数据（如网关的 EUI 和频带配置）的信息。

   - 网关的 EUI

     单个网关设备的 EUI（扩展唯一标识符）。EUI 是一个 16 位的字母数字代码，例如 `c0ee40ffff29df10`，它可以唯一标识 LoRaWAN 网络中的网关。此信息特定于您的网关型号，您可以在网关设备或其用户说明手册中找到。

> **Note**
>
> 网关的 EUI 与您可能看到印在网关设备上的 Wi-Fi MAC 地址不同。EUI 遵循 EUI-64 标准，该标准可唯一标识您的网关，因此不能在其他 AWS 账户和区域中重复使用。

- 频段 (RFRegion)

  网关的频带。您可以从 US915、EU868、AU915 或者 AS923-1 中进行选择，具体取决于您的网关支持的标准以及网关从哪个国家或地区进行实际连接。有关频段的更多信息，请参阅 为您的网关和设备连接选择 LoRa 频带。

2. 指定您的无线网关配置数据（可选）

   这些字段是可选字段，您可以使用它们提供有关网关及其配置的其他信息。

   - 网关的名称、描述和标签

     这些可选字段中的信息来自于您如何组织和描述无线系统中的元素。您可以将 Name（名称）添加到网关，使用 Description（说明）字段提供有关网关的信息，并使用 Tag（标签）添加有关网关的元数据键值对。有关命名和描述资源的更多信息，请参阅 描述您的 AWS IoT Wireless 资源。

   - 使用子带和滤波器的 LoRaWAN 配置

     此外，您还可以指定 LoRaWAN 配置数据，例如要使用的子带和可控制流量的滤波器。在本教程中，您可以跳过这些字段。有关更多信息，请参阅配置网关的子带和筛选功能。

3. 将 AWS IoT 事物与网关关联

   指定是否创建 AWS IoT 事物，并将其与网关联。AWS IoT 中的事物可以让您更轻松地搜索和管理您的设备。将某个事物与您的网关关联将允许网关访问其他 AWS IoT Core 特征。

4. 创建并下载网关证书

   对您的网关进行身份验证，以便它能够安全地与 AWS IoT 通信，您的 LoRaWAN 网关必须向 适用于 LoRaWAN 的 AWS IoT Core 提供私有秘钥和证书。创建网关证书，从而使 AWS IoT 能够使用 X.509 标准验证网关的身份。

   单击 Create certificate（创建证书）按钮并下载证书文件。稍后您将使用它们来配置网关。

5. 复制 CUPS 和 LNS 端点并下载证书

在建立与 适用于 LoRaWAN 的 AWS IoT Core 之间的连接时，LoRaWAN 网关必须连接到 CUPS 或 LNS 端点。建议您使用 CUPS 端点，因为它还提供配置管理。要验证 适用于 LoRaWAN 的 AWS IoT Core 的端点的身份，您的网关将为每个 CUPS 和 LNS 端点使用信任证书，

单击 Copy（复制）按钮复制 CUPS 和 LNS 端点。稍后您将需要此信息来配置您的网关。然后单击 Download server trust certificates（下载服务器信任证书）按钮下载 CUPS 和 LNS 端点的信任证书。

6. 为网关权限创建 IAM 角色

   您需要添加 IAM 角色，以允许 Configuration and Update Server (CUPS) 管理网关凭证。

   > **Note**
   >
   > 在此步骤中，您将创建 IoTWirelessGatewayCertManager 角色。如果已创建该角色，则可以跳过此步骤。您必须在 LoRaWAN 网关尝试连接到 适用于 LoRaWAN 的 AWS IoT Core 之前执行此操作；不过，该操作只需执行一次。

   要为您的账户创建 IoTWirelessGatewayCertManager IAM 角色，请单击 Create role（创建角色）按钮。如果该角色已存在，请从下拉列表中选择该角色。

   单击 Submit（提交）以完成网关创建。

使用 API 添加网关

如果您是首次使用 API 或 CLI 添加网关，必须将 IoTWirelessGatewayCertManager IAM 角色，以便网关可以与 适用于 LoRaWAN 的 AWS IoT Core 连接。有关如何创建角色的信息，请参阅以下部分 添加 IAM 角色以允许 Configuration and Update Server (CUPS) 管理网关凭证。

以下列表描述了执行与添加、更新或删除 LoRaWAN 网关相关任务的 API 操作。

适用于 LoRaWAN 的 AWS IoT Core 网关的 AWS IoT Wireless API 操作

- CreateWirelessGateway
- GetWirelessGateway
- ListWirelessGateways
- UpdateWirelessGateway
- DeleteWirelessGateway

有关可用于创建和管理 适用于 LoRaWAN 的 AWS IoT Core 资源的操作和数据类型的完整列表，请参阅 AWS IoT Wireless API 参考。

如何使用 AWS CLI 添加网关

您可以使用 AWS CLI 创建无线网关，方法是利用 create-wireless-gateway 命令。以下示例将展示如何创建无线 LoRaWAN 设备网关。您也可以提供 `input.json` 文件，该文件将包含其他详细信息，例如网关证书和预调配凭证。

> **Note**
>
> 您也可以使用 API 执行此流程，即使用 AWS API 中与此处显示的 CLI 命令对应的方法。

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

有关您可以使用的 CLI 的信息，请参阅 AWS CLI 参考

连接您的 LoRaWAN 网关并验证其连接状态

在检查网关连接状态之前，您必须已添加网关并将其连接到 适用于 LoRaWAN 的 AWS IoT Core。有关如何添加网关的信息，请参阅 将网关添加到 适用于 LoRaWAN 的 AWS IoT Core。

将网关连接到 适用于 LoRaWAN 的 AWS IoT Core

添加网关后，连接到网关的配置接口以输入配置信息和信任证书。

将网关的信息添加到 适用于 LoRaWAN 的 AWS IoT Core 之后，请向网关设备添加一些 适用于 LoRaWAN 的 AWS IoT Core 信息。网关供应商提供的文档中描述的过程应告知您如何将证书文件上载到网关以及配置网关设备以与 适用于 LoRaWAN 的 AWS IoT Core 通信。

可以与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用的网关

有关如何配置 LoRaWAN 网关的说明，请参阅 适用于 LoRaWAN 的 AWS IoT Core 工作室的配置网关设备部分。在这里，您可以找到相关说明，指导您连接能够与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用的网关。

支持 CUPS 协议的网关

以下说明介绍如何连接支持 CUPS 协议的网关。

1. 上载您在添加网关时获得的以下文件。
   - 网关设备证书和私有密钥文件。
   - CUPS 端点的信任证书文件，`cups.trust`。
2. 指定您之前获得的 CUPS 端点 URL。端点的格式为 *`prefix`*`.cups.lorawan.`*`region`*`.amazonaws.com:443`。

有关如何获取此信息的详细信息，请参阅 将网关添加到 适用于 LoRaWAN 的 AWS IoT Core。

支持 LNS 协议的网关

以下说明介绍如何连接支持 LNS 协议的网关。

1. 上载您在添加网关时获得的以下文件。
   - 网关设备证书和私有密钥文件。
   - LNS 端点的信任证书文件，`lns.trust`。
2. 指定您之前获得的 LNS 端点 URL。端点的格式为 *`prefix`*`.lns.lorawan.`*`region`*`.amazonaws.com:443`。

有关如何获取此信息的详细信息，请参阅 将网关添加到 适用于 LoRaWAN 的 AWS IoT Core。

在您将您的网关连接到 适用于 LoRaWAN 的 AWS IoT Core 后，您可以使用控制台或 API 检查连接状态并获取有关何时接收上一个上行链路的信息。

使用控制台检查网关连接状态

要使用控制台检查连接状态，请导航到 AWS IoT 控制台中的 Gateway（网关）页面，然后选择已添加的网关。在 Gatetway（网关）详细信息页面的 LoRaWAN specific details（LoRaWAN 具体详情）部分，您将看到连接状态以及上次接收上行链路的日期和时间。

使用 API 检查网关连接状态

要使用 API 检查连接状态，请使用 `GetWirelessGatewayStatistics` API。此 API 没有请求正文，只包含一个响应正文，用于显示网关是否已连接以及上一个上行链路于何时收到。

```
HTTP/1.1 200
```

```
Content-type: application/json

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

将您的设备登记到 适用于 LoRaWAN 的 AWS IoT Core

在您将网关登记到 适用于 LoRaWAN 的 AWS IoT Core 并验证其连接状态后，您就可以登记无线设备了。有关如何登记网关的信息，请参阅 将您的网关登记到 适用于 LoRaWAN 的 AWS IoT Core。

LoRaWAN 设备使用 LoRaWAN 协议与云托管的应用程序交换数据。适用于 LoRaWAN 的 AWS IoT Core 支持符合 LoRa Alliance 标准化 1.0.x 或 1.1 LoRAWAN 规范的设备。

LoRaWAN 设备通常包含一个或多个传感器和角色。这些设备通过 LoRaWAN 网关将上行链路遥测数据发送到 适用于 LoRaWAN 的 AWS IoT Core。云托管应用程序可以通过 LoRaWAN 网关向 LoRaWAN 设备发送下行链路命令来控制传感器。

登记无线设备之前

在您将无线设备登记到 适用于 LoRaWAN 的 AWS IoT Core 前，您需要提前准备好以下信息：

- LoRaWAN 规范和无线设备配置

  提前准备好每台设备独有的配置参数，使数据输入控制台更加顺畅。您需要输入的具体参数取决于设备使用的 LoRaWAN 规范。有关其规格和配置参数的完整列表，请参阅每个设备的文档。

- 设备名称和描述（可选）

  这些可选字段中的信息来自于您如何组织和描述无线系统中的元素。有关命名和描述资源的更多信息，请参阅 描述您的 AWS IoT Wireless 资源。

- 设备和服务配置文件

  准备好一些无线设备配置参数，这些参数由许多设备共享，并且可以存储在 适用于 LoRaWAN 的 AWS IoT Core 中作为设备和服务配置文件。配置参数可在设备的文档中或设备本身找到。在添加设备之前，您需要确定与设备配置参数匹配的设备配置文件，或创建一个配置文件（如有必要）。有关更多信息，请参阅将配置文件添加到 适用于 LoRaWAN 的 AWS IoT Core。

- 适用于 LoRaWAN 的 AWS IoT Core 目标

必须将每个设备分配到一个目标，该目标将处理其消息以发送到 AWS IoT 和其他服务。负责处理和发送设备消息的 AWS IoT 规则特定于设备的消息格式。要处理来自设备的消息并将其发送到正确的服务，请确定您将创建与设备消息一起使用的目标，并将其分配给设备。

将您的无线设备登记到 适用于 LoRaWAN 的 AWS IoT Core

- 将您的无线设备添加到 适用于 LoRaWAN 的 AWS IoT Core
- 将配置文件添加到 适用于 LoRaWAN 的 AWS IoT Core
- 将目标添加到 适用于 LoRaWAN 的 AWS IoT Core
- 创建规则以处理 LoRaWAN 设备消息
- 连接您的 LoRaWAN 设备并验证其连接状态

将您的无线设备添加到 适用于 LoRaWAN 的 AWS IoT Core

如果您是首次添加无线设备，我们建议您使用控制台。在 AWS IoT 控制台中导航到 适用于 LoRaWAN 的 AWS IoT Core 简介页面，选择 Get started（开始使用），然后选择 Add device（添加设备）。如果您已添加设备，请选择 View device（查看设备）以查看您添加的网关。如果要添加更多设备，请选择 Add device（添加设备）。

此外，您还可以从 AWS IoT 控制台的 Devices（设备）页面添加无线设备。

使用控制台将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core

根据您的激活方式和 LoRaWAN 版本选择 Wireless device specification（无线设备规范）。选择后，您的数据将使用AWS拥有并为您管理的密钥加密。

OTAA 和 ABP 激活模式

在 LoRaWAN 设备能够发送上行链路数据之前，您必须完成一个名为激活或者联接流程的操作。要激活您的设备，您可以使用 OTAA（空中激活）或 ABP（通过个性化激活）。

ABP 不需要联接流程，并使用静态密钥。当您使用 OTAA 时，LoRaWAN 设备会发送联接请求，网络服务器将允许该请求。我们建议您使用 OTAA 激活您的设备，因为每次激活都会生成新的会话密钥，这样更加安全。

LoRaWAN 版本

当您使用 OTAA 时，您的 LoRaWAN 设备和云托管应用程序将共享根密钥。这些根密钥取决于您使用的是版本 v1.0.x 还是 v1.1。v1.0.x 只有一个根密钥，即 AppKey（应用程序密钥）；而 v1.1 有两个

根密钥，AppKey（应用程序密钥）和 NwkKey（网络密钥）。会话密钥是根据每次激活的根密钥派生的。NwkKey 和 AppKey 都是无线供应商提供的 32 位十六进制值。

无线设备 EUI

在您选择无线设备规范时，您会看到控制台上显示无线设备的 EUI（扩展唯一标识符）参数。您可以从设备或无线供应商的文档中找到此信息。

- DevEUI：设备标签或文档中找到的 16 位六进制值，为您的设备所专有。
- AppEUI：在设备文档中找到的联接服务器的 16 位六进制值，为此服务器所专有。在 LoRaWAN v1.1 版本中，AppEUI 被称为 JoinEUI。

有关唯一标识符、会话密钥和根密钥的详细信息，请参阅 LoRa Alliance 文档。

使用 API 将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core

如果要使用 API 添加无线设备，则必须先创建设备配置文件和服务配置文件，然后再创建无线设备。创建无线设备时，您将用到设备配置文件和服务配置文件 ID。有关如何使用 API 创建这些配置文件的信息，请参阅 使用 API 添加设备配置文件。

以下列表描述了执行与添加、更新或删除服务配置文件相关任务的 API 操作。

服务配置文件的 AWS IoT Wireless API 操作

- CreateWirelessDevice
- GetWirelessDevice
- ListWirelessDevices
- UpdateWirelessDevice
- DeleteWirelessDevice

有关可用于创建和管理 适用于 LoRaWAN 的 AWS IoT Core 资源的操作和数据类型的完整列表，请参阅 AWS IoT Wireless API 参考。

如何使用 AWS CLI 创建无线设备

您可以使用 AWS CLI 创建无线设备，方法是使用 create-wireless-device 命令。以下示例通过使用输入 input.json 文件来输入参数以创建无线设备。

> **Note**
>
> 您也可以使用 API 执行此流程，即使用 AWS API 中与此处显示的 CLI 命令对应的方法。

input.json 的内容

```
{
    "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
}
```

您可以将此文件作为 `create-wireless-device` 命令的输入。

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

有关您可以使用的 CLI 的信息，请参阅 AWS CLI 参考

将配置文件添加到 适用于 LoRaWAN 的 AWS IoT Core

可以定义设备和服务配置文件来描述常见设备配置。这些配置文件描述了设备共享的配置参数，以便更轻松地添加这些设备。 适用于 LoRaWAN 的 AWS IoT Core 支持设备配置文件和服务配置文件。

要输入到这些配置文件的配置参数和值由设备制造商提供。

添加设备配置文件

设备配置文件定义网络服务器用于设置 LoRaWAN 无线电访问服务的设备功能和引导参数。它包括选择参数，例如 LoRa 频段、LoRa 区域参数版本和设备的 MAC 版本。要了解不同频段的信息，请参阅为您的网关和设备连接选择 LoRa 频带。

使用控制台添加设备配置文件

如果您按 使用控制台将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core 中所述使用控制台添加无线设备，则在添加无线设备规范后，您可以添加设备配置文件。此外，您还可以在 LoRaWAN 选项卡上的 AWS IoT 控制台的 Profiles（配置文件）的页面上添加无线设备。

您可以从默认设备配置文件中进行选择或创建新的设备配置文件。建议您使用默认设备配置文件。如果您的应用程序要求您创建设备配置文件，请提供设备配置文件名称，选择 Frequency ban (RfRegion)（频段（射频区域）），并将其他设置保持为默认值，除非在设备文档中另有指定。

使用 API 添加设备配置文件

如果要使用 API 添加无线设备，则必须在创建无线设备之前创建设备配置文件。

以下列表描述了执行与添加、更新或删除服务配置文件相关任务的 API 操作。

服务配置文件的 AWS IoT Wireless API 操作

- CreateDeviceProfile
- GetDeviceProfile
- ListDeviceProfiles
- UpdateDeviceProfile
- DeleteDeviceProfile

有关可用于创建和管理 适用于 LoRaWAN 的 AWS IoT Core 资源的操作和数据类型的完整列表，请参阅 AWS IoT Wireless API 参考。

如何使用 AWS CLI 创建设备配置文件

您可以使用 AWS CLI 创建设备配置文件，方法是使用 create-device-profile 命令。以下示例创建了一个配置文件。

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

运行此命令会自动创建一个带有 ID 的设备配置文件，您可以在创建无线设备时使用该 ID。现在，您可以使用以下 API 创建服务配置文件，然后使用设备和服务配置文件创建无线设备。

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

有关您可以使用的 CLI 的信息，请参阅 AWS CLI 参考

添加服务配置文件

服务配置文件描述了设备与应用程序服务器通信所需的通信参数。

使用控制台添加服务配置文件

如果您正在按照 使用控制台将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core 中所述使用控制台添加无线设备，则在添加设备配置文件后，您可以添加服务配置文件。此外，您还可以在 LoRaWAN 选项卡上的 AWS IoT 控制台的 Profiles（配置文件）的页面上添加无线设备。

建议您将设置保留启用 AddGWMetaData，从而您将接收到每个负载的其他网关元数据，例如用于数据传输的 RSSI 和 SNR。

使用 API 添加服务配置文件

如果要使用 API 添加无线设备，则必须先创建服务配置文件，然后再创建无线设备。

以下列表描述了执行与添加、更新或删除服务配置文件相关任务的 API 操作。

服务配置文件的 AWS IoT Wireless API 操作

- CreateServiceProfile
- GetServiceProfile
- ListServiceProfiles
- UpdateServiceProfile
- DeleteServiceProfile

有关可用于创建和管理 适用于 LoRaWAN 的 AWS IoT Core 资源的操作和数据类型的完整列表，请参阅 AWS IoT Wireless API 参考。

如何使用 AWS CLI 创建服务配置文件

您可以使用 AWS CLI 创建服务，方法是使用 create-service-profile 命令。以下示例创建了一个服务配置文件。

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

运行此命令会自动创建一个带有可在创建无线设备时使用的 ID 的服务配置文件。现在，您可以使用设备和服务配置文件创建无线设备。

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

将目标添加到 适用于 LoRaWAN 的 AWS IoT Core

适用于 LoRaWAN 的 AWS IoT Core 目标描述了 AWS IoT 规则，该规则用于处理设备数据以供AWS服务采用。

因为大多数 LoRaWAN 设备不会将数据以AWS能够使用的格式发送到适用于 LoRaWAN 的 AWS IoT Core，因为 AWS IoT 规则必须先行处理。AWS IoT 规则包含解释设备数据的 SQL 语句和主题规则操作，这些操作会将 SQL 语句的结果发送到将要使用它的服务。

如果您是首次添加目标，我们建议您使用控制台。

使用控制台添加一个目标

如果您正在按照 使用控制台将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core 中所述使用控制台添加无线设备，则当您将无线设备规范和配置文件按前述操作添加到 适用于 LoRaWAN 的 AWS IoT Core 之后，您可以继续添加目标。

您还可以从 AWS IoT 控制台的 Destinations（目标）页面添加一个 适用于 LoRaWAN 的 AWS IoT Core 目标。

要处理设备的数据，请在创建适用于 LoRaWAN 的 AWS IoT Core 目标时指定以下字段，然后选择 Add destination（添加目标）。

- 目标详细信息

  输入 Destination name（目标名称）以及您的目标描述（可选）。

- Rule name（规则名称）

  被配置为评估设备发送的消息并处理设备数据的 AWS IoT 规则。规则名称将映射到您的目标。目标要求规则来处理收到的消息。您可以通过调用 AWS IoT 规则或通过发布到 AWS IoT 消息代理。

  - 如果选择 Enter a rule name（输入规则名称），请输入名称，然后选择 Copy（复制）以复制创建 AWS IoT 规则时要输入的规则名称。您可以选择 Create rule（创建规则）以立即创建规则，或导航到 AWS IoT 控制台的 Rule（规则）中心并使用该名称创建规则。

    您也可以输入规则并使用高级设置以指定主题名称。主题名称是在规则调用期间提供的，可通过使用 `topic` 规则中的表达式访问。有关AWS IoT规则的更多信息，请参阅https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html。
  - 如果选择 Publish to AWS IoT message broker（发布到消息代理），输入主题名称。然后，您可以复制 MQTT 主题名称，多个订阅者可以订阅此主题以接收发布到该主题的消息。有关更多信息，请参阅https://docs.aws.amazon.com/iot/latest/developerguide/topics.html。

  有关目标的 AWS IoT 规则的更多信息，请参阅 创建规则以处理 LoRaWAN 设备消息。
- 角色名称

  IAM 角色，该角色授予设备数据权限以访问在 Rule name（规则名称）中命名的规则。在控制台中，您可以创建新的服务角色或选择现有的服务角色。如果要创建新的服务角色，可以输入角色名称（例如，**`IoTWirelessDestinationRole`**），或将其留白 适用于 LoRaWAN 的 AWS IoT Core 以生成新的角色名称。然后 适用于 LoRaWAN 的 AWS IoT Core 将代表您自动创建具有适当权限的 IAM 角色。

  有关 IAM 角色的更多信息，请参阅使用 IAM 角色。

使用 API 添加目标

如果要改为使用 CLI 添加目标，必须先为目标创建规则和 IAM 角色。有关角色中目标所需的详细信息，请参阅 为您的目标创建 IAM 角色。

以下列表包括执行与添加、更新或删除目标相关任务的 API 操作。

AWS IoT Wireless目标的 API 操作

- CreateDestination
- GetDestination
- ListDestinations

- UpdateDestination
- DeleteDestination

有关可用于创建和管理 适用于 LoRaWAN 的 AWS IoT Core 资源的操作和数据类型的完整列表，请参阅 AWS IoT Wireless API 参考。

如何使用 AWS CLI 添加一个目标

您可以使用 AWS CLI 添加目标，方法是使用 create-destination 命令。下面的示例演示如何通过输入规则名称创建目标，并将 RuleName 用作 expression-type 参数的值。如果要指定用于发布或订阅消息代理的主题名称，请将 expression-type 参数值更改为MqttTopic。

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

运行此命令可创建带有指定目标名称、规则名称和角色名称的目标。有关目标的规则和角色名称的信息，请参阅 创建规则以处理 LoRaWAN 设备消息 和 为您的目标创建 IAM 角色。

有关您可以使用的 CLI 的信息，请参阅 AWS CLI 参考。

为您的目标创建 IAM 角色

适用于 LoRaWAN 的 AWS IoT Core 目标需要 IAM 角色为 适用于 LoRaWAN 的 AWS IoT Core 提供将数据发送到 AWS IoT 规则所需的权限。如果尚未定义此类角色，必须对其进行定义，以便出现在角色列表中。

如本主题之前所述，当您使用控制台添加目标时，适用于 LoRaWAN 的 AWS IoT Core 自动为您创建 IAM 角色。使用 API 或 CLI 添加目标时，必须为目标创建 IAM 角色。

为您的 适用于 LoRaWAN 的 AWS IoT Core 目标角色创建 IAM 策略

1. 打开 IAM 控制台中的 Policies（策略）中心。
2. 选择 Create policy（创建策略），然后选择 JSON 选项卡。
3. 在编辑器中，删除编辑器中的所有内容并粘贴此策略文档。

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeEndpoint",
                "iot:Publish"
            ],
            "Resource": "*"
        }
    ]
}
```

4. 选择 Review policy（查看策略），并在 Name（名称）中输入此策略的名称。在下一流程中，您将需要使用此名称。

   如有需要，您还可以在 Description（说明）中描述此策略。

5. 选择创建策略。

要为 适用于 LoRaWAN 的 AWS IoT Core 目标创建 IAM 角色

1. 开启 IAM 控制台的角色中心，然后选择 Create role（创建角色）。
2. 在 Select type of trusted entity（选择受信任实体的类型）下，选择 Another AWS 账户（另一个亚马逊云科技账户）。
3. 在 Account ID（账户 ID），请输入您的AWS 账户 ID，然后选择 Next: Permission（下一步：权限）。
4. 在搜索框中，输入您在上一流程中创建的 IAM 策略名称。
5. 在搜索结果中，检查您在上一个流程中创建的 IAM 策略。
6. 依次选择 Next: Tags（下一步：标签）和 Next: Review（下一步：查看）。
7. 在 Role name（角色名称），输入此角色的名称，然后选择 Create role（创建角色）。
8. 在确认消息中，选择您创建的角色的名称以编辑新角色。
9. 在 Summary（摘要）页面上，选择 Trust relationships（信任关系），然后选择 Edit trust relationship（编辑信任关系）。
10. 在 Policy Document（策略文档）中，更改 `Principal` 属性以使其类似于此示例。

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

在您更改 `Principal` 属性后，完整的策略文档应该如此示例所示。

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

11. 要保存更改并退出，请选择 Update Trust Policy（更新信任策略）。

定义了此角色后，您可以在配置 适用于 LoRaWAN 的 AWS IoT Core 目标时在角色列表中找到它。

创建规则以处理 LoRaWAN 设备消息

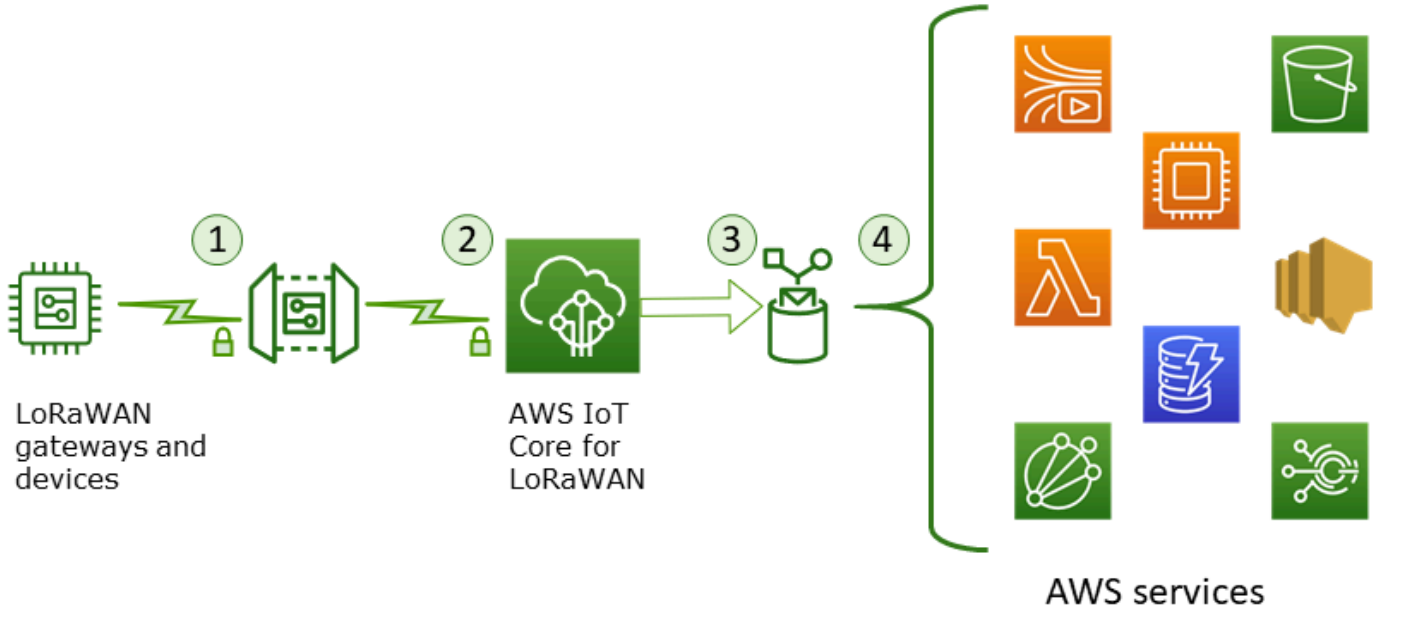
AWS IoT 规则会将设备消息发送到其他服务。AWS IoT 规则还可以处理从 LoRaWAN 设备接收的二进制消息，以便将消息转换为其他格式，从而更加便于其他服务使用。

适用于 LoRaWAN 的 AWS IoT Core 目标将无线设备与规则相关联，该规则将处理要发送到其他服务的设备消息数据。该规则会在 适用于 LoRaWAN 的 AWS IoT Core 收到数据时立即开始处理设备数据。适用于 LoRaWAN 的 AWS IoT Core 目标可由消息具有相同数据格式并将数据发送到同一服务的所有设备共享。

AWS IoT 规则如何处理设备消息

AWS IoT 规则处理设备消息数据的方式取决于将接收数据的服务、设备消息数据的格式以及该服务所需的数据格式。通常，该规则会调用 AWS Lambda 函数将设备的消息数据转换为服务所需的格式，然后将结果发送到服务。

以下示意图显示了在消息数据在从无线设备移动到AWS服务的过程中是如何确保安全以及如何处理的。



1. LoRaWAN 无线设备在传输二进制消息之前会使用 AES128 CTR 模式对其进行加密。
2. 适用于 LoRaWAN 的 AWS IoT Core 会对二进制消息进行解密，并将解密的二进制消息负载编码为 base64 字符串。
3. 生成的 base64 编码消息将作为二进制消息负载（未格式化为 JSON 文档的消息负载）发送到分配给设备的目标中所描述的 AWS IoT 规则。
4. AWS IoT 规则将消息数据定向到规则配置中描述的服务。

从无线设备接收的加密二进制负载不会被 适用于 LoRaWAN 的 AWS IoT Core 修改或解释。解密的二进制消息负载仅编码为 base64 字符串。要使服务访问二进制消息负载中的数据元素，必须通过规则调用的函数从负载中解析数据元素。base64 编码的消息负载是一个 ASCII 字符串，因此可以将其存储以便稍后进行解析。

为 LoRaWAN 设备创建规则

适用于 LoRaWAN 的 AWS IoT Core 使用 AWS IoT 规则来安全地将设备消息直接发送到其他 AWS 服务，而无需使用消息代理。通过从摄取路径中删除消息代理，可以降低成本并优化数据流。

适用于 LoRaWAN 的 AWS IoT Core 规则如要将设备消息发送到其他 AWS 服务，则需要 适用于 LoRaWAN 的 AWS IoT Core 目标，并要将 AWS IoT 规则分配给该目标。AWS IoT 规则必须包含 SQL 查询语句和至少一个规则操作。

通常，AWS IoT 规则查询语句由以下内容组成：

- 一个 SQL SELECT 子句，用于从消息负载中选择数据并设置其格式
- 标识要使用的消息的主题筛选条件（规则查询语句中的 FROM 对象）
- 可选条件语句（SQL WHERE 子句），用于指定执行操作的条件

以下是规则查询语句示例：

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

构建 AWS IoT 规则来处理来自 LoRaWAN 设备的负载时，则不必将 FROM 子句指定为规则查询对象的一部分。规则查询语句必须具有 SQL SELECT 子句，并且可以包含或不包含 WHERE 子句。如果查询语句使用 FROM 子句，该子句会被忽略。

以下是一个可以处理来自 LoRaWAN 设备的负载的规则查询语句示例：

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

在此示例中，`PayloadData` 是由您的 LoRaWAN 设备发送的 base64 编码二进制负载。

下面是一个示例规则查询语句，它可以对传入的有效负载执行二进制解码，并将其转换为不同的格式，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
```

有关使用 SELECT AND WHERE 子句的更多信息，请参阅 https://docs.aws.amazon.com/iot/latest/developerguide/iot-sql-reference.html。

有关 AWS IoT 规则以及如何创建和使用该规则的信息，请参阅 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html 和 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules-tutorial.html。

有关创建和使用 适用于 LoRaWAN 的 AWS IoT Core 目标的信息，请参阅 将目标添加到 适用于 LoRaWAN 的 AWS IoT Core。

有关在规则中使用二进制消息负载的信息，请参阅 https://docs.aws.amazon.com/iot/latest/developerguide/binary-payloads.html。

有关用于在传输过程中保护消息负载的数据安全和加密技术的详细信息，请参阅 AWS IoT Wireless 中的数据保护。

有关显示 IoT 规则二进制解码和实现示例的参考体系架构，请参阅 GitHub 上的 适用于 LoRaWAN 的 AWS IoT Core 解决方案示例。

连接您的 LoRaWAN 设备并验证其连接状态

在检查设备连接状态之前，您必须已经添加设备并将其连接到 适用于 LoRaWAN 的 AWS IoT Core。有关如何添加设备的信息，请参阅 将您的无线设备添加到 适用于 LoRaWAN 的 AWS IoT Core。

添加设备后，请参阅设备的用户手册，了解如何从 LoRaWAN 设备初始发送上行链路消息。

使用控制台检查设备连接状态

要使用控制台检查连接状态，请导航到 AWS IoT 控制台的 Devices（设备）的页面，然后选择已添加的设备。在无线设备详细信息页面的 Details（详细信息）部分，您将看到上次接收上行链路的日期和时间。

使用 API 检查设备连接状态

要使用 API 检查连接状态，请使用 `GetWirelessDeviceStatistics` API。此 API 没有请求体，只包含显示最后一个上行链路何时收到的响应体。

```
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"
}
```

后续步骤

现在，您已连接设备并验证了连接状态，您可以使用 AWS IoT 控制台 Test（测试）页面的 MQTT 测试客户端观察从设备接收的上行链路元数据格式。有关更多信息，请参阅查看从 LoRaWAN 设备发送的上行链路消息的格式。

使用 适用于 LoRaWAN 的 AWS IoT Core 配置无线资源的位置

在使用此特征之前，请注意，用于解析 LoRaWAN 设备位置信息的所选第三方提供商依赖于国际 GNSS 服务 (IGS)、通过 NASA 的 EarthData 或其他第三方提供或维护的数据源和数据集。这些数据源和数据集是第三方内容（定义见客户协议），并按原样提供。有关更多信息，请参阅 AWS 服务条款。

可以使用 适用于 LoRaWAN 的 AWS IoT Core 指定静态位置数据，或使用第三方解析器激活定位功能以实时识别设备的位置。您可以添加或更新 LoRaWAN 设备和/或网关的位置信息。

在将设备或网关添加到 适用于 LoRaWAN 的 AWS IoT Core 或编辑设备或网关的配置详细信息时，可以指定位置信息。位置信息被指定为 GeoJSON 有效负载。GeoJSON 格式是一种用于对地理数据结构进行编码的格式。此有效负载中包含设备位置的纬度和经度坐标，这些坐标基于世界大地坐标系统 (WGS84)。

求解器计算资源的位置后，如果您具有 Amazon Location Service，可以激活 Amazon Location 地图，其中将显示您的资源的位置。使用位置数据，您可以：

- 激活定位以识别和获取 LoRaWAN 设备的位置。
- 跟踪和监控网关和设备的位置。
- 定义用于处理对位置数据的任何更新并将其路由到其他 AWS 服务 的 AWS IoT 规则。有关规则操作的列表，请参阅《AWS IoT Developer Guide》中的AWS IoT rule actions。
- 使用位置数据和 Amazon SNS 创建提醒，并在发生任何异常活动时向设备接收通知。

LoRaWAN 设备定位的工作原理

可以使用第三方 Wi-Fi 和 GNSS 求解器激活定位以识别设备的位置。可以使用此信息跟踪和监控设备。以下步骤向您展示如何激活定位和查看 LoRaWAN 设备的位置信息。

> **Note**
>
> 第三方求解器只能与具有 LoRa Edge 芯片的 LoRaWAN 设备一起使用。不能与 LoRaWAN 网关一起使用。对于网关，您仍然可以指定静态位置信息，并在 Amazon Location 地图上识别该位置。

1. 添加设备

   在激活定位功能之前，先将设备添加到 适用于 LoRaWAN 的 AWS IoT Core。LoRaWAN 设备必须具有 LoRa Edge 芯片，LoRa Edge 是一个超低功耗平台，集成了针对地理定位应用的远程 LoRa 收发器、多星座 GNSS 扫描仪和无源 Wi-Fi MAC 扫描仪。

2. 激活定位

   要获取设备的实时位置，请激活定位。当您的 LoRaWAN 设备发送上行链路消息时，会使用地理定位帧端口将消息中包含的 Wi-Fi 和 GNSS 扫描数据发送到 适用于 LoRaWAN 的 AWS IoT Core。

3. 检索位置信息

   从基于来自收发器的扫描结果计算的求解器中检索估计的设备位置。如果位置信息是使用 Wi-Fi 和 GNSS 扫描结果计算出来的，适用于 LoRaWAN 的 AWS IoT Core 会选择准确性较高的估计位置。

4. 查看位置信息

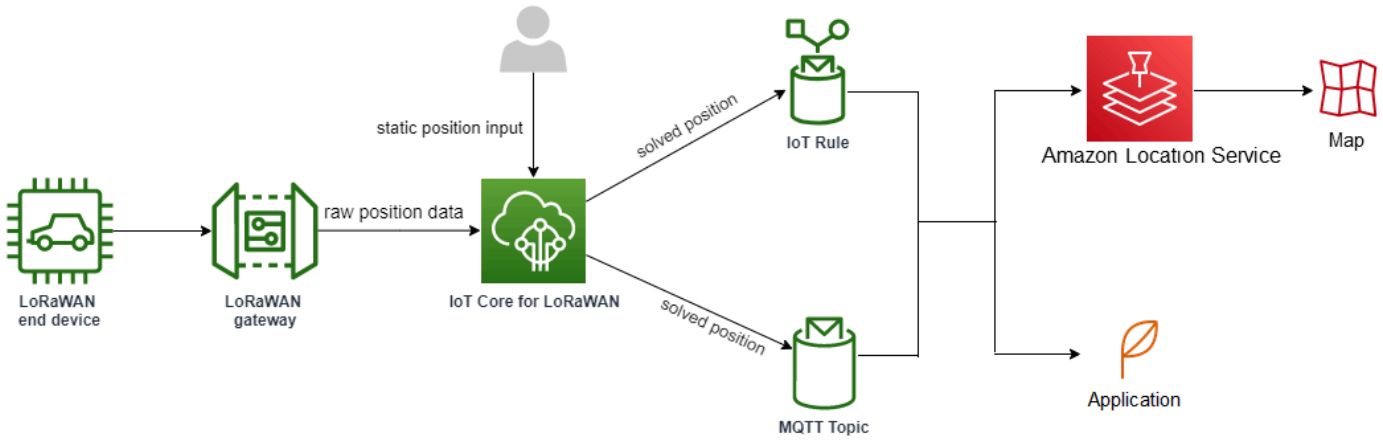
   求解器在计算位置信息后，还将提供准确性信息，该信息指示求解器计算的位置与您输入的静态位置信息之间的差异。您还可以在 Amazon Location 地图上查看设备位置。

   > **Note**
   >
   > 由于求解器不能用于 LoRaWAN 网关，因此准确性信息将报告为 `0.0`。

有关用于定位求解器的上行链路消息格式和频率端口的更多信息，请参阅 将上行链路消息从 适用于 LoRaWAN 的 AWS IoT Core 发送到规则引擎。

定位工作流概述

下图显示了 适用于 LoRaWAN 的 AWS IoT Core 如何存储和更新设备和网关的位置信息。



1. 指定资源的静态位置

   使用纬度和经度坐标指定设备或网关的静态位置信息，作为 GeoJSON 有效负载。您还可以指定可选的高度坐标。这些坐标基于 WGS84 坐标系。有关更多信息，请参阅 World Geodetic System (WGS84) [世界大地测量系统（WGS84）]。
2. 激活设备定位

   如果您使用的是具有 LoRa Edge 芯片的 LoRaWAN 设备，则可以选择激活定位来实时跟踪您的设备位置。当您的设备发送上行链路消息时，会使用地理定位帧端口将 GNSS 和 Wi-Fi 扫描数据发送到 适用于 LoRaWAN 的 AWS IoT Core。然后，求解器使用此信息来解析设备位置。
3. 添加目标以路由位置数据

   您可以添加一个目标来描述用于处理设备数据的 IoT 规则，并将更新的位置信息路由到 适用于 LoRaWAN 的 AWS IoT Core。您还可以在 Amazon Location 地图上查看资源的上次已知位置。

配置资源位置

您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI 配置资源的位置。

如果您的设备具有 LoRa Edge 芯片，您可以激活定位来计算实时位置信息。对于您的网关，您仍然可以输入静态位置坐标并使用 Amazon Location 在 Amazon Location 地图上跟踪网关位置。

主题

配置 LoRaWAN 网关的位置

将网关添加到 适用于 LoRaWAN 的 AWS IoT Core 时，您可以指定静态位置数据。如果您激活了 Amazon Location Service 地图，位置数据将显示在 Amazon Location 地图上。

> **Note**
>
> 第三方求解器不能与 LoRaWAN 网关一起使用。对于网关，您仍然可以指定静态位置坐标。如果不使用求解器来计算位置（例如，在网关的情况下），准确度信息将报告为 `0.0`。

您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI 配置网关位置。

使用控制台配置网关的位置

要使用 AWS Management Console 配置网关资源的位置，请先登录控制台，然后转到 AWS IoT 控制台的 Gateways（网关）中心页面。

添加位置信息

为网关添加位置配置

1. 在 Gateways（网关）中心页面中，选择 Add gateway（添加网关）。
2. 输入网关的 EUI、频段（RFRegion）以及任何其他网关详细信息和 LoRaWAN 配置信息。有关更多信息，请参阅使用控制台添加网关。
3. 转到 Position information - Optional（位置信息 - 可选），使用纬度和经度坐标以及可选的海拔坐标输入网关的位置信息。位置信息基于 WGS84 坐标系。

查看网关的位置

配置网关位置后，适用于 LoRaWAN 的 AWS IoT Core 会创建一个称为 `iotwireless.map` 的 Amazon Location 地图。您可以在网关详细信息页面中的 Position（位置）选项卡上查看此地图。根据您指定的位置坐标，网关的位置将在地图上显示为标记。您可以放大或缩小以在地图上清晰地查看网关的位置。在 Position（位置）选项卡上，您还将看到准确性信息，以及确定网关位置的时间戳。

> **Note**
>
> 如果没有安装 Amazon Location Service 地图，您将看到一条消息，指示您必须使用 Amazon Location Service 才能访问地图并查看网关位置。使用 Amazon Location Service 地图可能会对您的 AWS 账户 额外计费。有关更多信息，请参阅AWS IoT Core 定价。

地图 `iotwireless.map` 用作可使用 `Get` API 操作（`GetMapTile`）访问的地图数据的来源。有关与地图一起使用的 `Get` API 的信息，请参阅 Amazon Location Service API 参考。

要获取有关此地图的其他详细信息，请转到 Amazon Location Service 控制台，选择 maps（地图），然后选择 iotwireless.map。有关更多信息，请参阅《Amazon Location Service 开发人员指南》中的地图。

更新网关的位置配置

要更改网关的位置配置，请在网关详细信息页面中选择 Edit（编辑），然后更新位置信息和目标。

> **Note**
>
> 有关历史位置数据的信息不可用。当您更新网关的位置坐标时，它将覆盖以前报告的位置数据。更新位置后，在网关详细信息的 Position（位置）选项卡中，您将看到新的位置信息。时间戳的更改表示它对应于网关的上次已知位置。

使用 API 配置网关的位置

您可以指定位置信息，并使用 AWS IoT Wireless API 或 AWS CLI 配置网关位置。

> **Important**
>
> 不再支持 API 操作 UpdatePosition、GetPosition、PutPositionConfiguration、GetPositionConfiguration 和 ListPositionConfigurations。更新和检索位置信息的调用应改用 GetResourcePosition 和 UpdateResourcePosition API 操作。

添加位置信息

要添加给定无线网关的静态位置信息，请使用 UpdateResourcePosition API 操作或 update-resource-position CLI 命令指定坐标。将 `WirelessGateway` 指定为 `ResourceType`，将要更新的无线网关 ID 指定为 `ResourceIdentifier`，并将位置信息指定为 GeoJSON 有效负载。

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

下面显示的是 *gatewayposition.json* 文件的内容。

gatewayposition.json 的内容

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

运行此命令不会生成任何输出。要查看您指定的位置信息，请使用 `GetResourcePosition` API 操作。

获取位置信息

要获取给定无线网关的位置信息，请使用 GetResourcePosition API 操作或 get-resource-position CLI 命令。将 `WirelessGateway` 指定为 `resourceType`，并提供无线网关的 ID 作为 `resourceIdentifier`。

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

运行此命令会显示无线网关的位置信息，作为 GeoJSON 有效负载。您将看到有关位置坐标、位置信息类型和其他属性（例如与网关的上次已知位置相对应的时间戳）的信息。

```
{
    {
    "type": "Point",
```

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

配置 LoRaWAN 设备的位置

将设备添加到 适用于 LoRaWAN 的 AWS IoT Core 时，您可以指定静态位置信息，也可以激活定位并指定目标。目标描述用于处理设备位置信息并将更新的位置路由到 Amazon Location Service 的 IoT 规则。配置设备位置后，位置数据将显示在 Amazon Location 地图上，其中包含准确性信息和您指定的目标。

您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI 配置设备的位置。

上行链路消息的帧端口和格式

如果激活定位，则必须指定用于将 Wi-Fi 和 GNSS 扫描数据从设备传输到 适用于 LoRaWAN 的 AWS IoT Core 的地理定位帧端口。系统会使用此帧端口将位置信息传输给 适用于 LoRaWAN 的 AWS IoT Core。

LoRaWAN 规范提供了数据传输字段（FRMPayload）和端口字段（FPort）来区分不同类型的消息。要传输位置信息，您可以为帧端口指定一个介于 1 到 223 之间的任意值。FPort 0 为 MAC 消息预留，FPort 224 为 MAC 合规性测试预留，端口 225-255 为未来标准化应用扩展预留。

将上行链路消息从 适用于 LoRaWAN 的 AWS IoT Core 发送到规则引擎

当您添加目标时，它会创建一个 AWS IoT 规则，以使用规则引擎将数据路由到 Amazon Location Service。然后，更新的位置信息将显示在 Amazon Location 地图上。如果您尚未激活，则在您更新设备的静态位置坐标时，目标将路由位置数据。

以下代码显示了从 适用于 LoRaWAN 的 AWS IoT Core 发送的上行链路消息的格式，其中包含位置信息、准确度、解析器配置和无线元数据。下面突出显示的字段是可选的。如果没有垂直准确性信息，则此值为 `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 ### LoRaWAN # AWS IoT Core
    "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"

        }
    }
```

```
}
```

使用控制台配置设备的位置

要使用 AWS Management Console配置和管理设备位置，请先登录控制台，然后转到 AWS IoT 控制台的 Devices（ 设备 ）中心页面。

添加位置信息

添加设备的位置信息：

1. 在 Devices（设备）中心页面中，选择 Add wireless device（添加无线设备）。
2. 输入无线设备规格、设备和服务配置文件以及用于定义将数据路由到其他 AWS 服务 的 IoT 规则的目标。有关更多信息，请参阅将您的设备登记到 适用于 LoRaWAN 的 AWS IoT Core。
3. 输入位置信息，可以选择激活地理位置，并指定要用于路由消息的位置数据目标。
   - 位置信息

     使用纬度和经度坐标以及可选的高度坐标指定设备的位置数据。位置信息基于 WGS84 坐标系。
   - 地理位置

     如果您希望 适用于 LoRaWAN 的 AWS IoT Core 使用地理位置来计算设备位置，请激活定位功能。它使用第三方 GNSS 和 Wi-Fi 求解器实时识别设备位置。

     要输入地理位置信息，请选择激活定位，然后输入用于向 适用于 LoRaWAN 的 AWS IoT Core 传输 GNSS 和 Wi-Fi 扫描数据的地理位置帧端口。您将看到填充的默认 FPort 供您参考。但是，您可以选择 1 到 223 之间的其他任意值。
   - 位置数据目标

     选择一个目标以描述用于处理设备位置数据并将此数据转发到 适用于 LoRaWAN 的 AWS IoT Core 的 AWS IoT 规则。此目标仅用于路由位置数据。它必须不同于您用于将设备数据路由到其他 AWS 服务 的目标。

查看设备的位置配置

配置设备的位置后，适用于 LoRaWAN 的 AWS IoT Core 会创建一个称为 `iotwireless.map` 的 Amazon Location 地图。您可以在设备详细信息页面中的 Position（位置）选项卡上查看此地图。根据您指定的位置坐标或第三方求解器计算的位置，设备的位置将在地图上显示为标记。您可以放大或缩小以在地图上清晰地查看设备的位置。在设备详细信息页面中的 Position（位置）选项卡上，您还将看到准确度信息、确定设备位置的时间戳以及您指定的位置数据目标。

> **Note**
>
> 如果尚未激活 Amazon Location Service 地图，您将看到一条消息，指示您必须使用 Amazon Location Service 才能访问地图并查看位置。使用 Amazon Location Service 地图可能会对您的 AWS 账户 额外计费。有关更多信息，请参阅AWS IoT Core 定价。

地图 `iotwireless.map` 用作可使用 `Get` API 操作（`GetMapTile`）访问的地图数据的来源。有关与地图一起使用的 `Get` API 的信息，请参阅 Amazon Location Service API 参考。

要获取有关此地图的其他详细信息，请转到 Amazon Location Service 控制台，选择 maps（地图），然后选择 iotwireless.map。有关更多信息，请参阅《Amazon Location Service 开发人员指南》中的地图。

更新设备的位置配置

要更改设备的位置配置，请在设备详细信息页面中选择 Edit（编辑），然后更新位置信息、任何地理位置设置和目标。

> **Note**
>
> 有关历史位置数据的信息不可用。当您更新设备的位置坐标时，它将覆盖以前报告的位置数据。更新位置后，在设备详细信息的 Position（位置）选项卡中，您将看到新的位置信息。时间戳的更改表示它对应于设备的上次已知位置。

使用 API 配置设备位置

您可以使用 AWS IoT Wireless API 或 AWS CLI 指定位置信息、配置设备位置，以及激活可选的地理位置。

> **Important**
>
> 不再支持 API 操作 UpdatePosition、GetPosition、PutPositionConfiguration、GetPositionConfiguration 和 ListPositionConfigurations。更新和检索位置信息的调用应改用 GetResourcePosition 和 UpdateResourcePosition API 操作。

添加位置信息和配置

要添加给定无线设备的位置信息，请使用 UpdateResourcePosition API 操作或 update-resource-position CLI 命令指定坐标。将 WirelessDevice 指定为 ResourceType，将要更新的无线设备 ID 指定为 ResourceIdentifier 和位置信息。

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

下面显示的是 *deviceposition.json* 文件的内容。要指定用于发送地理位置数据的 FPort 值，请将定位对象与 CreateWirelessDevice 和 UpdateWirelessDevice API 操作结合使用。

deviceposition.json 的内容

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

运行此命令不会生成任何输出。要查看您指定的位置信息，请使用 GetResourcePosition API 操作。

获取位置信息和配置

要获取给定无线设备的位置信息，请使用 GetResourcePosition API 或 get-resource-position CLI 命令。将 WirelessDevice 指定为 resourceType，并提供无线设备的 ID 作为 resourceIdentifier。

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

运行此命令会将无线设备的位置信息显示为 GeoJSON 有效负载。您将看到有关位置坐标、位置类型和属性的信息，这些信息可能包括准确性信息和对应于设备最后一个已知位置的时间戳。

```
{
     "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 管理网关

以下是将网关与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用时的一些重要注意事项。有关如何向 适用于 LoRaWAN 的 AWS IoT Core 添加网关的信息，请参阅 将您的网关登记到 适用于 LoRaWAN 的 AWS IoT Core。

LoRa Basics Station 软件要求

如要连接到 适用于 LoRaWAN 的 AWS IoT Core，您的 LoRaWAN 网关上必须运行名为 LoRa Basics Station 的软件。LoRa 基础站是一个开源软件，由 Semtech Corporation 维护，并通过 GitHub 存储库进行分发。适用于 LoRaWAN 的 AWS IoT Core 支持 LoRa Basics Station 2.0.4 及更高版本。最新版本为 2.0.6。

使用来自AWS Partner Device Catalog 的合格网关

AWS 合作伙伴设备目录包含网关和开发人员工具包，这些网关和开发人员工具包可与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用。我们建议您使用这些合格的网关，因为您不必修改嵌入软件才能将网关连接到 AWS IoT Core。这些网关已经具有一个版本的 BasicStation 软件，兼容 适用于 LoRaWAN 的 AWS IoT Core。

> **Note**
>
> 如果您的网关未在 Partner Catalog 中列为可与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用的合格网关，则如果网关正在运行 2.0.4 及更高版本的 LoRa Basic Station 软件，那么您仍然能够使用它。确保使用 TLS Server and Client Authentication（TLS 服务器和客户端身份验证）来验证您的 LoRaWAN 网关。

使用 CUPS 和 LNS 协议

LoRa Basics Station 软件包含两个子协议，用于将网关连接到网络服务器，即 LoRaWAN Network Server (LNS) 和 Configuration and Update Server (CUPS) 协议。

LNS 协议在 LoRa Basics Station 兼容网关和网络服务器之间建立了数据连接。LoRa 上行链路和下行链路消息将借助安全 WebSockets 通过此数据连接进行交换。

CUPS 协议支持凭证管理以及网关远程配置和固件更新。适用于 LoRaWAN 的 AWS IoT Core 提供 LNS 和 CUPS 端点，分别用于 LoRaWAN 数据摄取和远程网关管理。

有关更多信息，请参阅 LNS 协议和 CUPS 协议。

主题

配置 LoRaWAN 网关的信标和筛选功能

在使用 LoRaWAN 设备时，您可以为 LoRaWAN 网关配置某些可选参数。这些参数包括：

- 信标

  您可以为充当 B 类 LoRaWAN 设备的网桥的 LoRaWAN 网关配置信标参数。这些设备在预定时隙接收下行链路消息，因此，您必须为网关配置信标参数，才能传输这些时间同步信标。

- 过滤

  您可以为 LoRaWAN 网关配置 `NetID` 和 `JoinEUI` 参数，以筛选设备数据流量。筛选流量有助于节省带宽用量并减少网关与 LNS 之间的流量。

- 子频段

  您可以为网关配置子频段，以指定要使用的特定子频段。对于无法在不同子频带之间跳转的无线设备，您可以使用此功能与仅使用该特定子频带中的频率通道的设备通信。

以下主题包含有关这些参数及其配置方法的更多信息。信标参数在 AWS Management Console中不可用，只能使用 AWS IoT Wireless API 或 AWS CLI 指定。

主题

将网关配置为向 B 类设备发送信标

如果您将 B 类无线设备登记到 适用于 LoRaWAN 的 AWS IoT Core，这些设备会在预定时隙内接收下行链路消息。这些设备根据网关传输的时间同步信标打开这些时隙。为了让网关传输这些时间同步信标，可以使用 适用于 LoRaWAN 的 AWS IoT Core 为网关配置某些信标相关参数。

要配置这些信标参数，您的网关必须运行 LoRa Basics Station 2.0.6 版。请参阅 使用来自AWS Partner Device Catalog 的合格网关。

如何配置信标参数

> **Note**
>
> 只有在网关与 B 类无线设备通信时，才需要为网关配置信标参数。

使用 `CreateWirelessGateway` API 操作将网关添加到 适用于 LoRaWAN 的 AWS IoT Core 时，可以配置信标参数。调用 API 操作时，通过为网关使用 `Beaconing` 对象指定以下参数。配置参数后，网关将以 128 秒的间隔向您的设备发送信标。

- `DataRate`：传输信标的网关的数据速率。
- `Frequencies`：网关传输信标的频率列表。

下面的示例显示如何为网关配置这些参数。`input.json` 文件将包含其他详细信息，如网关证书和预调配凭证。有关使用 `CreateWirelessGateway` API 操作将网关添加到 适用于 LoRaWAN 的 AWS IoT Core 的更多信息，请参阅使用 API 添加网关。

> **Note**
>
> 当您使用 AWS IoT 控制台将网关添加到 适用于 LoRaWAN 的 AWS IoT Core 时，信标参数不可用。

```
aws iotwireless create-wireless-gateway \
```

```
    --name "myLoRaWANGateway" \
    --cli-input-json file://input.json
```

下面显示的是 input.json 文件的内容。

input.json 的内容

```
{
    "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]
    }
}
```

下面的代码显示运行此命令的输出示例。

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

获取有关信标参数的信息

您可以使用 GetWirelessGateway API 操作获取有关网关信标参数的信息。

> **Note**
>
> 如果已经登记网关，则不能使用 UpdateWirelessGateway API 操作配置信标参数。要配置这些参数，您必须删除网关，然后在使用 CreateWirelessGateway API 操作添加网关时指定这些参数。

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

运行此命令会返回有关您的网关和信标参数的信息。

配置网关的子带和筛选功能

LoRaWAN 网关运行 LoRa Basics Station 软件，使网关能够连接到 适用于 LoRaWAN 的 AWS IoT Core。为连接到 适用于 LoRaWAN 的 AWS IoT Core，您的 LoRa 网关首先查询 CUPS 服务器的 LNS 端点，然后建立与该端点的 WebSockets 数据连接。建立连接后，上行链路和下行链路帧便可以通过该连接进行交换。

筛选网关接收的 LoRa 数据帧

LoRaWAN 网关建立与端点的连接后， 适用于 LoRaWAN 的 AWS IoT Core 将响应一条 `router_config` 消息，为 LoRa 网关的配置指定一系列参数，其中包括筛选参数 `NetID` 和 `JoinEui`。有关 `router_config` 的更多信息以及如何与 LoRaWAN Network Server (LNS) 建立连接，请参阅 LNS 协议。

```
{
"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
}
```

这些网关通常通过 Wi-Fi、Ethernet 或蜂窝移动网络等高带宽网络将 LoRaWAN 设备数据传输到 LNS。网关通常会接收所有消息，然后将收到的流量传送到 适用于 LoRaWAN 的 AWS IoT Core。但是，您可以配置网关以筛选某些设备数据流量，这有助于节省带宽用量并减少网关与 LNS 之间的流量流。

要配置 LoRa 网关以筛选数据帧，可以在 `router_config` 消息中使用参数 `NetID` 和 `JoinEui`。`NetID` 是接受的 NetID 值列表。任何带有除所列内容以外数据帧的 LoRa 数据帧都将被删除。`JoinEui` 是编码 JoinEUI 值范围的整数值对列表。联接请求帧将被网关删除，除非消息中的字段 `JoinEui` 处于 [BegEui，EndEui] 范围内。

频率通道和子频段

对于 US915 和 AU915 射频区域，无线设备可选择 64 个 125 KHz 和 8 个 500kHz 上行链路通道，以便使用 LoRa 网关访问 LoRaWAN 网络。上行链路频率通道分为 8 个子频段，每个子频段有 8 个 125KHz 通道和一个 500KqHz 通道。对于 AU915 区域中的每个常规网关，将支持一个或多个子频段。

某些无线设备无法在子频段之间跳远，且在连接到 适用于 LoRaWAN 的 AWS IoT Core 时仅在一个子频段中使用频率通道。对于要传输且来自这些设备的上行链路数据包，请配置 LoRa 网关以使用该特定子频段。对于其他射频区域（如 EU868）的网关，不需要此配置。

使用控制台配置您的网关以使用筛选条件和子频段

您可以配置网关以使用特定的子频段，还可以启用筛选 LoRa 数据帧的功能。要使用控制台指定这些参数：

1. 导航到 AWS IoT 控制台的 适用于 LoRaWAN 的 AWS IoT Core Gateways（网关）页面并选择 Add gateway（添加网关）。
2. 指定网关详细信息，例如 Gateway's Eui（网关的 Eui）、Frequency band (RFRegion)（频段（射频区域））和可选内容，如 Name（名称）和 Description（说明），然后选择是否将 AWS IoT 事物与您的网关关联。有关如何添加网关的更多信息，请参阅 使用控制台添加网关。
3. 在 LoRaWAN configuration（LoRaWAN 配置）部分中，您可以指定子频段和筛选信息。
   - `SubBands`：要添加子频段，请选择 Add SubBand（添加子频段）并指定一个整数值列表，这些值将说明网关支持哪些子频段。`SubBands` 参数只能在 `RfRegion` US915 和 AU915 上进行配置，并且在其中支持的区域之一中拥有处于范围 `[1,8]` 内的值。
   - `NetIdFilters`：要筛选上行链路帧，请选择 Add NetId（添加 NetId）并指定网关使用的字符串值列表。来自无线设备的传入上行链路帧的 NetID 必须与至少一个列出的值匹配，否则该帧将被丢弃。

- `JoinEuiFilters`：选择 Add JoinEui 范围（添加 JoinEui 范围），并指定网关用于筛选 LoRa 帧的字符串值对列表。作为来自无线设备的联接请求的一部分指定的 JoinEui 值必须至少在一个 JoinEuiRange 值的范围内，每个值都以 [BegeUI, EndeUI] 对的形式列示，否则该帧将被删除。

4. 然后，您可以按照 使用控制台添加网关 中所述的指示继续配置网关。

添加网关后，在 AWS IoT 控制台的 适用于 LoRaWAN 的 AWS IoT Core Gateways（网关）页面，如果选择已添加的网关，则可以在网关详细信息页面的 LoRaWAN specific details（LoRaWAN 具体详情）部分看到 SubBands 和筛选条件 `NetIdFilters` 及 `JoinEuiFilters`。

使用 API 配置您的网关以使用筛选条件和子频段。

您可以使用您用来创建网关的 CreateWirelessGateway API，来配置要使用的子频段并启用筛选功能。借助 `CreateWirelessGateway` API，您可以指定子频段和筛选条件，将其作为您使用 `LoRaWAN` 字段所提供的网关配置信息的一部分。下面显示包含此信息的请求令牌。

```
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 来更新筛选条件，但不更新子频段。如果 `JoinEuiFilters` 和 `NetIdfilters` 值为空，这意味着字段没有更新。如果值不为空且包含空列表，则应用更新。若要获取您指定的字段的值，请使用 GetWirelessGateway API。

使用 适用于 LoRaWAN 的 AWS IoT Core 的 CUPS 服务更新网关固件

您的网关上运行的 LoRa Basics Station 软件使用 Configuration and Update Server (CUPS) 协议提供凭证管理和固件更新接口。CUPS 协议借助 ECDSA 签名提供安全的固件更新交付。

您必须经常更新网关的固件。您可以使用 适用于 LoRaWAN 的 AWS IoT Core 的 CUPS 服务来向能够签署更新的网关提供固件更新。要更新网关固件，您可以使用 SDK 或 CLI，但不要使用控制台。

该升级过程可能最多需要 45 分钟才能完成。如果您是首次设置网关来连接到 适用于 LoRaWAN 的 AWS IoT Core，则可能需要更长的时间。网关制造商通常会提供自己的固件更新文件和签名，因此您可以使用它们并前进至 将固件文件上载到 S3 存储桶并添加 IAM 角色。

如果您没有固件更新文件，请参阅 生成固件更新文件和签名 作为示例，您可以使用该示例来修改您的应用程序。

要执行网关固件更新：

- 生成固件更新文件和签名
- 将固件文件上载到 S3 存储桶并添加 IAM 角色
- 使用任务定义安排并运行固件更新

生成固件更新文件和签名

本流程的步骤是可选的，取决于您使用的网关。网关制造商会以更新文件或脚本的形式提供自己的固件更新，Basics Station 会在后台运行此脚本。在这种情况下，您很可能会在所使用网关的发布说明中找到固件更新文件。然后，您可以改为使用该更新文件或脚本，然后前进至 将固件文件上载到 S3 存储桶并添加 IAM 角色。

如果您没有此脚本，下面显示了用于生成固件更新文件的命令。您还可以签署更新，以确保代码未被更改或损坏，并且设备仅运行由受信任的作者发布的代码。

在此流程中，您将：

- 生成固件更新文件
- 为固件更新生成签名
- 查看后续步骤

生成固件更新文件

在网关上运行的 LoRa Basics Station 软件能够接收 CUPS 响应中的固件更新。如果您没有制造商提供的脚本，请参阅以下固件更新脚本，该脚本是为基于 Raspberry Pi 的 RAKWireless Gateway 编写的。我们有一个基本脚本，而新工作站二进制文件、版本文件和 `station.conf` 将被附加其上。

> **Note**
>
> 该脚本特定于 RAKWireless Gateway ，因此您必须根据您使用的网关对其进行调整，以使其适用于您的应用程序。

基本脚本

下面显示了基于 Raspberry Pi 的 RAKWireless Gateway 的基本脚本示例。您可以将以下命令保存在文件 `base.sh` 中，然后在 Raspberry Pi 的 Web 浏览器终端中运行脚本。

```
*#!/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
```

添加负载脚本

在基本脚本中，我们会附加 Basics Station 二进制文件、标识要更新版本的 version.txt 和名为 `addpayload.sh` 脚本中的 `station.conf`。然后，运行此脚本。

```
*#!/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
```

运行这些脚本后，您可以在终端中运行以下命令来生成固件更新文件，fwstation。

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

为固件更新生成签名

LoRa Basics Station 软件提供带有 ECDSA 签名的已签名固件更新。要支持已签名的更新，您需要：

- 必须由 ECDSA 私有秘钥生成且少于 128 字节的签名。
- 用于签名的私有密钥且必须存储在网关中，其文件名格式应为 sig-%d.key。建议使用 sig-0.key 作为文件名。
- 私有密钥上的 32 位 CRC。

签名和 CRC 将传递给 适用于 LoRaWAN 的 AWS IoT Core API。要生成以前的文件，可以使用以下脚本，gen.sh。该脚本是根据 GitHub 存储库中的 basicstaion 示例编写的。

```
*#!/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
```

脚本生成的私有秘钥应保存到网关中。密钥文件采用二进制格式。

```
./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
```

查看后续步骤

现在您已经生成了固件和签名，请转到下一个主题，将固件文件 `fwstation` 上载到 Amazon S3 存储桶。存储桶是将固件更新文件作为对象存储的容器。您可以添加 IAM 角色，该角色将授予 CUPS 服务器读取 S3 存储桶中的固件更新文件的权限。

将固件文件上载到 S3 存储桶并添加 IAM 角色

您可以使用 Amazon S3 创建存储桶，这是一个可以存储固件更新文件的容器。您可以将文件上载到 S3 存储桶，并添加允许 CUPS 服务器从存储桶读取更新文件的 IAM 角色。有关 Amazon S3 的更多信息，请参阅 Amazon S3 入门。

要上载的固件更新文件取决于您使用的网关。如果您遵循类似于 生成固件更新文件和签名 中所述的流程，则您将上载通过运行脚本所生成的 `fwstation` 文件。

完成本流程需要大约 20 分钟。

要上载您的固件文件：

- 创建 Amazon S3 存储桶并上载更新文件
- 创建具有读取 S3 存储桶权限的 IAM 角色
- 查看后续步骤

创建 Amazon S3 存储桶并上载更新文件

您将使用 AWS Management Console 创建 Amazon S3 存储桶，然后将固件更新文件上载到存储桶中。

创建 S3 存储桶

要创建 S3 存储桶，请打开 Amazon S3 控制台。登录（如果尚未登录），然后执行以下步骤：

1. 选择创建存储桶。
2. 输入唯一且有意义的名称作为 Bucket name（存储桶名称），例如 `iotwirelessfwupdate`。有关存储桶的建议命名约定，请参阅 https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html。

3. 确保您选择了AWS 区域来创建 LoRaWAN 网关和设备，并且选中 Block all public access（阻止所有公有访问）设置，以便您的存储桶使用默认权限。
4. 为 Bucket versioning（存储桶版本控制）选择 Enable（启用），这将帮助您将固件更新文件的多个版本保留在同一存储桶中。
5. 确认将 Server-side encryption（服务器端加密）设置为 Disable（禁用），然后选择 Create bucket（创建存储桶）。

上载固件更新文件

现在，您可以在显示在 AWS Management Console 的存储桶列表中看到您的存储桶了。选择您的存储桶并完成以下步骤以上载您的文件。

1. 选择您的存储桶，然后选择 Upload（上载）。
2. 选择 Add file（添加文件），然后上载固件更新文件。如果您按照 生成固件更新文件和签名 中所述的流程执行，您应上载 `fwstation` 文件，否则请上载网关制造商提供的文件。
3. 确保所有设置都保留为默认值。请确保 Predefined ACL（预定义 ACL）设置为 private（私有），然后选择 Upload（上载）以上载您的文件。
4. 复制您所上载文件的 S3 URI。选择您的存储桶，您将看到上载的文件显示在 Objects（对象）列表中。选择您的文件，然后选择 Copy S3 URI（复制 S3 URI）。如果您将存储桶命名为类似于前述示例 (`fwstation`)，则 URI 将类似于：`s3://iotwirelessfwupdate/fwstation`。在创建 IAM 角色时，您将使用 S3 URI。

创建具有读取 S3 存储桶权限的 IAM 角色

现在，您将创建 IAM 角色和策略，该角色和策略将授予 CUPS 从 S3 存储桶读取固件更新文件的权限。

为您的角色创建 IAM 策略

要为 适用于 LoRaWAN 的 AWS IoT Core 目标角色创建 IAM 策略，请打开 IAM 控制台的策略中心，然后完成以下步骤：

1. 选择 Create policy（创建策略），然后选择 JSON 选项卡。
2. 从编辑器中删除所有内容并粘贴此策略文档。策略提供访问 `iotwireless` 存储桶的权限，固件更新文件 `fwstation` 存储在对象内。

```
{
```

```
    "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. 选择 Review policy（查看策略），在 Name（名称）中，输入此策略的名称（例如，`IoTWirelessFwUpdatePolicy`）。在下一流程中，您将需要使用此名称。
4. 选择创建策略。

使用附加的策略创建 IAM 角色。

现在，您将创建一个 IAM 角色并附加以前为访问 S3 存储桶而创建的策略。打开 IAM 控制台的角色中心并完成以下步骤：

1. 选择创建角色。
2. 在 Select type of trusted entity（选择受信任实体的类型）下，选择 Another AWS 账户（另一个亚马逊云科技账户）。
3. 在 Account ID（账户 ID），请输入您的AWS 账户 ID，然后选择 Next: Permission（下一步：权限）。
4. 在搜索框中，输入您在上一流程中创建的 IAM 策略名称。在搜索结果中检查您之前创建的 IAM 策略（例如 `IoTWirelessFwUpdatePolicy`），然后选择该策略。
5. 依次选择 Next: Tags（下一步：标签）和 Next: Review（下一步：查看）。
6. 对于 Role name（角色名称），输入此角色的名称（例如 `IoTWirelessFwUpdateRole`），然后选择 Create role（创建角色）。

编辑 IAM 角色的信任关系

在运行上一步后显示的确认信息中，选择您创建的角色的名称进行编辑。您将编辑角色以添加以下信任关系。

1. 在刚刚创建的角色的 Summary（摘要）页面上，选择 Trust relationship（信任关系）选项卡，然后选择 Edit trust relationship（编辑信任关系）。
2. 在 Policy Document（策略文档）中，更改 `Principal` 属性以使其类似于此示例。

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

在您更改 `Principal` 属性后，完整的策略文档应该如此示例所示。

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

3. 要保存更改并退出，请选择 Update Trust Policy（更新信任策略）。
4. 获取角色的 ARN。选择在“摘要”部分您的 IAM 角色，您将看到 Role ARN（角色 ARN），例如 `arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`。复制此 Role ARN（角色 ARN）。

查看后续步骤

现在，您已创建 S3 存储桶和允许 CUPS 服务器读取 S3 存储桶的 IAM 角色，请转到下一个主题以安排并运行固件更新。保留您之前复制的 S3 URI 和 Role ARN（角色 ARN），以便您可以输入这些信息以创建将运行的任务定义，以执行固件更新。

使用任务定义安排并运行固件更新

您可以使用任务定义包含有关固件更新的详细信息并定义更新。适用于 LoRaWAN 的 AWS IoT Core 将根据与网关关联的以下三个字段中的信息提供固件更新。

- Station

  Basics Station 软件的版本和构建时间。要识别此信息，您还可以使用网关运行的 Basics Station 软件生成该信息（例如，`2.0.5(rpi/std) 2021-03-09 03:45:09`）。
- PackageVersion

  固件版本，由网关中的文件 `version.txt` 指定。虽然网关中可能不存在此信息，但我们建议您使用此方法定义固件版本（例如，`1.0.0`）。
- 模型

  网关正在使用的平台或模型（例如，Linux）。

此流程需要 20 分钟时间完成。

要完成此流程：

- 获取网关上运行的当前版本
- 创建无限网关任务定义
- 运行固件更新任务并跟踪进度

获取网关上运行的当前版本

要确定网关是否有资格进行固件更新，CUPS 服务器将检查所有三个字段：`Station`、`PackageVersion` 和 `Model`，网关在 CUPS 请求期间提供这些信息时，服务器将进行匹配。使用任务定义时，这些字段将作为 `CurrentVersion` 字段的一部分予以存储。

您可以使用 适用于 LoRaWAN 的 AWS IoT Core API 或 AWS CLI 为您的网关获取 `CurrentVersion`。以下命令演示如何使用 CLI 获取此信息。

1. 如果您已经预调配了网关，则可以使用 get-wireless-gateway 命令获取有关网关的信息。

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

下面显示了此命令的示例输出。

```
{
    "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. 借助 `get-wireless-gateway` 命令报告的无线网关 ID，您可以使用 get-wireless-gateway-firmware-information 命令来获取 `CurrentVersion`。

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

下面显示了该命令的示例输出，其中包含 `CurrentVersion` 显示的所有三个字段的信息。

```
{
    "LoRaWAN": {
        "CurrentVersion": {
            "PackageVersion": "1.0.0",
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
        }
    }
}
```

创建无限网关任务定义

在创建任务定义时，我们建议您使用 AutoCreateTasks 参数指定任务自动创建。`AutoCreateTasks` 适用于与前面提到的所有三个参数匹配的任何网关。如果禁用此参数，则必须手动将参数分配给网关。

您可以使用 适用于 LoRaWAN 的 AWS IoT Core API 或 AWS CLI 创建无限网关任务定义。以下命令演示如何使用 CLI 创建任务定义。

1. 创建一个文件，input.json，它将包含要传递给 CreateWirelessGatewayTaskDefinition API 的信息。在 input.json 文件中，提供之前获取的以下信息：

   - UpdateDataSource

     提供指向对象的链接，该对象中包含您上载到 S3 存储桶的固件更新文件（例如，s3://iotwirelessfwupdate/fwstation）。

   - UpdateDataRole

     提供指向您创建的 IAM 角色的角色 ARN 的链接，该角色提供读取 S3 存储桶的权限（例如，arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole）。

   - SigKeyCRC 和 UpdateSignature

     此信息可能由您的网关制造商提供，但如果您遵循 生成固件更新文件和签名 中所述的流程，则您将在生成签名时找到此信息。

   - CurrentVersion

     提供您之前通过运行 get-wireless-gateway-firmware-information 命令获得的 CurrentVersion 输出。

   ```
   cat input.json
   ```

   下面显示的是 input.json 文件的内容。

   ```
   {
       "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. 将 input.json 文件传递到 create-wireless-gateway-task-definition 命令以创建任务定义。

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

以下显示了命令的输出。

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

运行固件更新任务并跟踪进度

网关已准备好接收固件更新，一旦打开电源，就会连接到 CUPS 服务器。当 CUPS 服务器在网关版本中找到匹配项时，它会计划固件更新。

任务是正在进行的任务定义。当您通过将 AutoCreateTasks 设置为 True 指定自动任务创建时，一旦找到匹配的网关，固件更新任务就会立即启动。

您可以使用 GetWirelessGatewayTask API 跟踪任务的进度。当您首次运行 get-wireless-gateway-task 命令时，它会将任务状态显示为 IN_PROGRESS。

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

以下显示了命令的输出。

```
{
    "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"
}
```

下次运行命令时，如果固件更新生效，它将显示更新后的字段，Package、Version 和 Model，而任务状态将变为 COMPLETED。

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

以下显示了命令的输出。

```
{
    "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"
}
```

在此示例中，我们向您展示了使用基于 Raspberry Pi 的 RAKWireless 网关的固件更新。固件更新脚本停止 BasicStation 的运行，以存储更新后的 Package、Version 和 Model 字段，因此必须重新启动 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
```

如果固件更新失败，您将从 CUPS server 看到 FIRST_RETRY 的状态，而网关会发送相同的请求。如果 SECOND_RETRY 后 CUPS 服务器无法连接到网关，它将显示 FAILED 的状态。

在上一任务处于 COMPLETED 或者 FAILED 之后，使用 delete-wireless-gateway-task 命令删除旧任务，然后再启动一个新任务。

```
aws iotwireless delete-wireless-gateway-task \
```

```
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

选择网关以接收 LoRaWAN 下行链路数据流量

当您从 适用于 LoRaWAN 的 AWS IoT Core 向设备发送下行链路消息时，您可以选择要用于下行链路数据流量的网关。您可以指定单个网关或从网关列表中进行选择，以接收下行链路流量。

如何指定网关列表

当使用 SendDataToWirelessDevice API 操作从 适用于 LoRaWAN 的 AWS IoT Core 向设备发送下行链路消息时，您可以指定要使用的单个网关或网关列表。调用 API 操作时，通过为网关使用 ParticipatingGateways 对象指定以下参数。

> **Note**
>
> 您要使用的网关列表在 AWS IoT 控制台中不可用。仅当使用 SendDataToWirelessDevice API 操作或 CLI 时，才可以指定要使用的这一网关列表。

- DownlinkMode：指示是以顺序模式还是并发模式发送下行链路消息。对于 A 类设备，请指定 UsingUplinkGateway，以仅使用先前上行链路消息传输中选定的网关。
- GatewayList：您要用于发送下行链路数据流量的网关列表。下行链路负载将以指定的频率发送到指定的网关。这是使用由 GatewayId 和 DownlinkFrequency 对组成的 GatewayListItem 对象的列表来表示的。
- TransmissionInterval：适用于 LoRaWAN 的 AWS IoT Core 在将负载传输到下一个网关之前将等待的时间。

> **Note**
>
> 您可以将此网关列表指定为仅在向 B 类或 C 类无线设备发送下行链路消息时使用。如果您使用 A 类设备，则在向设备发送下行链路消息时，将使用您在发送上行链路消息时选择的网关。

下面的示例显示如何为网关指定这些参数。input.json 文件将包含其他详细信息。有关使用 SendDataToWirelessDevice API 操作发送下行链路消息的更多信息，请参阅使用 API 执行下行链路队列操作。

> **Note**
>
> 使用 AWS IoT 控制台从 适用于 LoRaWAN 的 AWS IoT Core 发送下行链路消息时，用于指定参与网关列表的参数不可用。

```
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
```

下面显示的是 input.json 文件的内容。

input.json 的内容

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

运行此命令的输出会生成下行链路消息的 MessageId。在某些情况下，即使收到 MessageId，数据包也可能丢失。有关如何解决错误的更多信息，请参阅 排查下行链路消息队列错误。

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

获取有关参与网关的列表的信息

您可以通过列出下行链路队列中的消息，获取有关参与接收下行链路消息的网关列表的信息。要列出消息，请使用 `ListQueuedMessages` API。

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

运行此命令会返回有关队列中的消息及其参数的信息。

使用 适用于 LoRaWAN 的 AWS IoT Core 管理设备

以下是将您的设备与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用时需要注意的重要事项。有关如何将设备添加到 适用于 LoRaWAN 的 AWS IoT Core 的信息，请参阅 将您的设备登记到 适用于 LoRaWAN 的 AWS IoT Core。

设备注意事项

选择要用于与 适用于 LoRaWAN 的 AWS IoT Core 通信的设备时，请考虑以下事项：

- 可用传感器
- 电池容量
- 能源消耗
- 费用
- 天线类型和传输范围

使用具有可与 适用于 LoRaWAN 的 AWS IoT Core 搭配使用的网关的设备

您使用的设备可以与能够同 适用于 LoRaWAN 的 AWS IoT Core 搭配使用的无限网关配对。这些网关和开发人员工具包可在 AWS 合作伙伴设备目录中找到。我们还建议您考虑将这些设备靠近您的网关。有关更多信息，请参阅使用来自AWS Partner Device Catalog 的合格网关。

LoRaWAN 版本

适用于 LoRaWAN 的 AWS IoT Core 支持所有符合 LoRa Alliance 标准化的 1.0.x 或 1.1 LoRAWAN 规范的设备。

激活模式

在 LoRaWAN 设备能够发送上行链路数据之前，您必须完成一个名为激活或者联接流程的操作。要激活您的设备，您可以使用 OTAA（空中激活）或 ABP（通过个性化激活）。我们建议您使用 OTAA 激活您的设备，因为每次激活都会生成新的会话密钥，这样更加安全。

您的无线设备规范基于 LoRaWAN 版本和激活模式，该模式决定了为每次激活生成的根密钥和会话密钥。有关更多信息，请参阅使用控制台将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core。

设备类

LoRaWAN 设备可以随时发送上行链路消息。侦听下行链路消息会消耗电池容量并缩短电池持续时间。LoRaWAN 协议指定了三类 LoRaWAN 设备。

- A 类设备大部分时间都处于睡眠状态，并且仅在短时间内侦听下行链路消息。这些设备主要是电池供电的传感器，电池寿命长达 10 年。
- B 类设备可以在计划的下行链路插槽中接收消息。这些设备主要是电池供电的执行器。
- C 类设备永远不会睡眠，并会持续侦听传入的消息，因此接收消息时没有太多延迟。这些器件主要是电源驱动器。

有关这些无线设备注意事项的更多信息，请参阅 了解有关 LoRaWAN 的更多信息 中提及的资源。

主题

- 使用 适用于 LoRaWAN 的 AWS IoT Core 执行自适应数据速率 (ADR)
- 管理 LoRaWAN 设备与 AWS IoT 之间的通信
- 管理来自公共 LoRaWAN 设备网络（Everynet）的 LoRaWAN 流量

使用 适用于 LoRaWAN 的 AWS IoT Core 执行自适应数据速率 (ADR)

适用于 LoRaWAN 的 AWS IoT Core 使用自适应数据速率来降低设备传输功耗，同时确保由网关接收来自终端设备的消息。自适应数据速率指示终端设备优化数据速率、传输功率和重传次数，同时尝试降

低网关收到的数据包的错误率。例如，如果您的终端设备靠近网关，则自适应数据速率会降低传输功率并提高数据传输速率。

主题

自适应数据速率（ADR）的工作原理

要启用 ADR，您的设备必须在帧头中设置 ADR 位。设置 ADR 位后，适用于 LoRaWAN 的 AWS IoT Core 会发送 `LinkADRReq` MAC 命令，您的设备将使用包含 ADR 命令的 ACK 状态的 `LinkADRAns` 命令进行响应。设备确认 ADR 命令后，它将按照来自 适用于 LoRaWAN 的 AWS IoT Core 的 ADR 说明调整传输参数值以获得最佳数据速率。

适用于 LoRaWAN 的 AWS IoT Core ADR 算法使用上行链路元数据历史记录中的 SINR 信息来确定设备使用的最佳传输功率和数据速率。该算法使用在帧头中设置 ADR 位后开始的 20 个最近的上行链路消息。为了确定重传次数，它使用数据包错误率（PER），即丢失的数据包总数的百分比。使用此算法时，您只能控制数据速率的范围，即数据速率的最小和最大限制。

配置数据速率限制（CLI）

默认情况下，当您在 LoRaWAN 设备的帧头中设置 ADR 位时，适用于 LoRaWAN 的 AWS IoT Core 将执行 ADR。使用 AWS IoT Wireless API 操作 `CreateServiceProfile` 或 AWS CLI 命令 `create-service-profile` 为 LoRaWAN 设备创建服务配置文件时，您可以控制数据速率的最小和最大限制。

> **Note**
>
> 从 AWS Management Console 创建服务配置文件时，无法指定最大和最小数据速率限制。只能使用 AWS IoT Wireless API 或 AWS CLI 指定该值。

要指定数据速率的最小和最大限制，请在 `CreateServiceProfile` API 操作中使用 `DrMin` 和 `DrMax` 参数。默认的最小和最大数据速率限制为 0 和 15。例如，以下 CLI 命令将最低数据速率限制设置为 3，将最大限制设置为 12。

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

运行此命令会生成服务配置文件的 ID 和 Amazon 资源名称（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 获取指定参数的值。

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

运行此命令会生成服务配置文件参数的值。

```
{
    "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
    }
}
```

如果您创建了多个配置文件，则可以使用 API 操作 ListServiceProfiles 或 AWS CLI 命令 list-service-profiles 列出 AWS 账户中的服务配置文件，然后使用 GetServiceProfile API 或 get-service-profile CLI 命令检索您为其自定义数据速率限制的服务配置文件。

管理 LoRaWAN 设备与 AWS IoT 之间的通信

将 LoRaWAN 设备连接到 适用于 LoRaWAN 的 AWS IoT Core 之后，您的设备可以开始向云发送消息。上行链路消息是从您设备发送并由 适用于 LoRaWAN 的 AWS IoT Core 接收的消息。LoRaWAN 设备可以随时发送上行链路消息，随后会转发至其他 AWS 服务 和云托管应用程序。从 适用于 LoRaWAN 的 AWS IoT Core 以及其他 AWS 服务 和应用程序发送到设备的消息称为下行链路消息。

下面演示了如何查看和管理在设备与云之间发送的上行链路和下行链路消息。您可以维护下行链路消息队列，然后按照这些消息的队列添加顺序将消息发送到设备。

主题

查看从 LoRaWAN 设备发送的上行链路消息的格式

将 LoRaWAN 设备连接到 适用于 LoRaWAN 的 AWS IoT Core 后，您可以观察将从无线设备接收的上行链路消息的格式。

在您观察上行链路消息之前

您必须已登记无线设备并将设备连接到 AWS IoT，以便它能够传输和接收数据。有关将您的设备登记到 适用于 LoRaWAN 的 AWS IoT Core 的更多消息，请参阅 将您的设备登记到 适用于 LoRaWAN 的 AWS IoT Core。

上行链路消息包含什么内容？

LoRaWAN 设备使用 LoRaWAN 网关连接到 适用于 LoRaWAN 的 AWS IoT Core。您从设备接收的上行链路消息将包含以下信息。

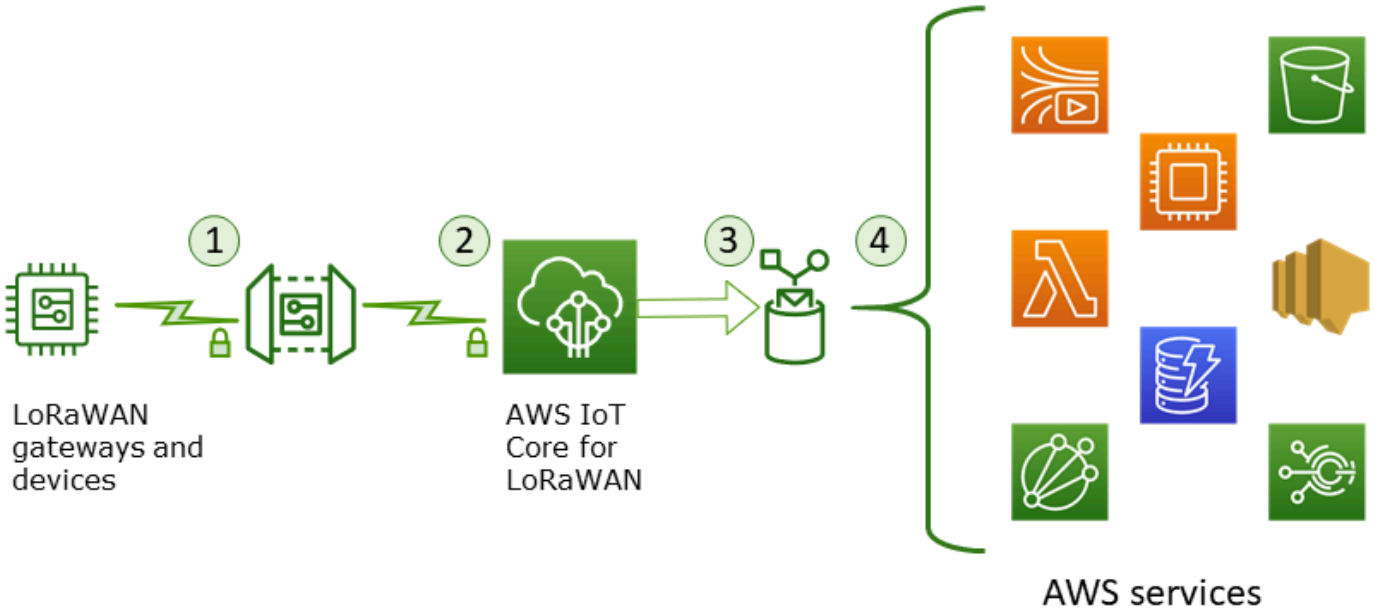
- 对应从无线设备发送的加密负载消息的负载数据。
- 无线元数据，包括：
  - 设备信息，例如 DevEui、数据速率和设备运行所在的频率通道。
  - 连接到设备的网关的可选附加参数和网关信息。网关参数包括网关的 EUI、SNR 和 RSSi。

  通过使用无线元数据，您可以获取有关无线设备的有用信息以及在设备和 AWS IoT 之间传输的数据。例如，您可以使用 `AckedMessageId` 参数来检查设备是否已收到上次确认的下行链路消息。或者，如果您选择包含网关信息，则可以确定是否要切换到更靠近设备的更强网关通道。

如何观察上行链路消息？

登记设备后，您可以使用 AWS IoT 控制台中 Test（测试）页面的 MQTT 测试客户端订阅创建目标时指定的主题。连接设备并开始发送负载数据后，您将开始看到消息。

此图表标识了连接到 适用于 LoRaWAN 的 AWS IoT Core 的 LoRaWAN 系统中的密钥元素，其中显示了主数据面板以及数据在系统中流动的方式。



当无线设备开始发送上行链路数据时，适用于 LoRaWAN 的 AWS IoT Core 将无线元数据信息与负载打包，然后将其发送到 AWS 应用程序。

上行链路消息示例

以下示例显示了从您的设备接收的上行链路消息的格式。

```
{
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",
    "PayloadData": "Cc48AAAAAAAAAAA=",
    "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"

        }
    }
}
```

从上行链路元数据中排除网关元数据

如果要从上行链路元数据中排除网关元数据信息，请在您创建服务配置文件时禁用 AddGwMetadata 参数。有关禁用此参数的信息，请参阅 添加服务配置文件。

在这种情况下，您将不会在上行链路元数据中看到 Gateways 部分，如以下示例所示。

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

管理要发送到 LoRaWAN 设备的下行链路消息队列

云托管应用程序和其他 AWS 服务 可以向无线设备发送下行链路消息。下行链路消息是从 适用于 LoRaWAN 的 AWS IoT Core 发送到无线设备的消息。您可以为已登记到 适用于 LoRaWAN 的 AWS IoT Core 的每台设备调度和发送下行链路消息。

如果要向多台设备发送下行链路消息，可以使用多播组。多播组中的设备共享同一个多播地址，将该地址分发到整组接收设备。有关更多信息，请参阅创建多播组向多台设备发送下行链路有效负载。

下行链路消息队列的工作原理

LoRaWAN 设备的设备类别决定了队列中的消息如何发送到设备。A 类设备将上行链路消息发送到 适用于 LoRaWAN 的 AWS IoT Core，表明设备可以接收下行链路消息。B 类设备可在常规的下行链路槽接收消息。C 类设备可以随时接收下行链路消息。有关设备类别的更多信息，请参阅 设备类。

下面演示了如何形成消息队列以及如何将消息发送到 A 类设备。

1. 适用于 LoRaWAN 的 AWS IoT Core 借助您通过 AWS IoT 控制台或 AWS IoT Wireless API 指定的帧端口、有效负载数据和确认模式参数，对添加到队列中的下行链路消息进行缓冲。
2. LoRaWAN 设备会发送上行链路消息，表明其处于线上状态且可以开始接收下行链路消息。
3. 如果已将多条下行链路消息添加到队列中，适用于 LoRaWAN 的 AWS IoT Core 会将队列中的第一条下行链路消息发送到已设置确认 (ACK) 标志的设备。
4. 设备可立即向 适用于 LoRaWAN 的 AWS IoT Core 发送上行链路消息，也可休眠直至出现下一条上行链路消息且消息中包含 ACK 标志。
5. 适用于 LoRaWAN 的 AWS IoT Core 收到带有 ACK 标志的上行链路消息后，会清除队列中的下行链路消息，表示设备已成功收到下行链路消息。如果三次检查后，上行链路消息中仍缺少 ACK 标志，则会丢弃该消息。

使用控制台执行下行链路队列操作

您可以使用 AWS Management Console 管理下行链路消息队列，以及根据需要清除单条消息或整个队列。如果为 A 类设备，收到从设备发送的表明设备处于线上状态的上行链路消息后，将队列消息发送到该设备。发送消息后，系统会自动从队列中清除该消息。

管理下行链路消息队列

创建下行链路消息队列

1. 转到 AWS IoT 控制台的设备中心，然后选择要管理下行链路消息队列的设备。
2. 在设备详细信息页面的 Downlink messages（下行链路消息）部分，选择 Queue downlink messages（管理下行链路消息队列）。
3. 要配置下行链路消息，请指定以下参数：
   - FPort：选择设备与 适用于 LoRaWAN 的 AWS IoT Core 通信的帧端口。
   - Payload（有效负载）：指定要发送到设备的有效负载消息。最大有效负载大小为 242 字节。如果已启用自适应数据速率 (ADR)，适用于 LoRaWAN 的 AWS IoT Core 会用其选择有效负载大小的租价数据速率。您可以根据需要进一步优化数据速率。
   - Acknowledge mode（确认模式）：确认设备是否收到下行链路消息。如果消息需要此模式，数据流会显示一条带有 ACK 标志的上行链路消息，并且该消息将从队列中清除。
4. 要将下行链路消息添加到队列中，请选择 Submit（提交）。

下行链路消息现已添加到队列中。如果没有显示消息或收到错误，您可以如 排查下行链路消息队列错误 中所述排查该错误。

> **Note**
>
> 将下行链路消息添加到队列后，便无法继续编辑 FPort 参数、Payload（有效负载）和 Acknowledge mode（确认模式）。要为这些参数发送具有不同值的下行链路消息，您可以删除此消息，再对具有更新参数值的新下行链路消息进行队列管理。

队列会列出已添加的下行链路消息。若要查看设备与 适用于 LoRaWAN 的 AWS IoT Core 之间交换的上行链路和下行链路消息的有效负载，您可以使用网络分析器。有关更多信息，请参阅使用网络分析器实时监控无线资源机群。

列出下行链路消息队列

已将创建的下行链路消息添加到队列中。后续每条下行链路消息均会在此消息之后添加到队列中。您可以在设备详细信息页面的 Downlink messages（下行链路消息）部分查看下行链路消息列表。收到上行链路消息后，消息就会发送到设备。设备收到下行链路消息后，该消息会从队列中删除。然后，下一条消息会在队列中向前移动，等待发送到设备。

删除单条下行链路消息或清除整个队列

发送到设备后，每条下行链路消息都会自动从队列中清除。您还可以删除单条消息或清除整个下行链路队列。这些操作无法撤消。

- 如果发现队列中有不想发送的消息，请选中相应消息并选择 Delete（删除）。
- 如果不想将任何消息从队列发送到设备，请选择 Clear downlink queue（清除下行链路队列）来清除整个队列。

使用 API 执行下行链路队列操作

您可以使用 AWS IoT Wireless API 管理下行链路消息队列，以及根据需要清除单条消息或整个队列。

管理下行链路消息队列

若要创建下行链路消息队列，请使用 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}
```

运行此命令的输出会生成下行链路消息的 MessageId。在某些情况下，即使收到 MessageId，数据包也可能丢失。有关如何解决错误的更多信息，请参阅 排查下行链路消息队列错误。

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

列出队列中的下行链路消息

若要列出队列中的所有下行链路消息，请使用 ListQueuedMessages API 操作或 list-queued-messages CLI 命令。

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

默认情况下，运行此命令时最多会显示 10 条下行链路消息。

删除单条下行链路消息或清除整个队列

若要从队列中删除单条消息或清除整个队列，请使用 DeleteQueuedMessages API 操作或 delete-queued-messages CLI 命令。

- 若要删除单条消息，请针对无线设备（由 wirelessDeviceId 指定）提供待删除消息的 messageID。
- 若要清除整个下行链路消息队列，请针对无线设备（由 wirelessDeviceId 指定）将 messageID 指定为 *。

排查下行链路消息队列错误

如果没有看到期望的结果，可检查以下事项：

- AWS IoT 控制台中未显示下行链路消息

  如果如 使用控制台执行下行链路队列操作 中所述添加下行链路消息后，但队列中并未显示下行链路消息，可能是因为设备尚未完成名为 activation（激活）或 join（联接）过程。当设备登记到 适用于 LoRaWAN 的 AWS IoT Core，此过程才算完成。有关更多信息，请参阅使用控制台将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core。

  将设备登记到 适用于 LoRaWAN 的 AWS IoT Core 后，您可以使用网络分析器或 Amazon CloudWatch 监控设备来检查联接和重新联接是否成功。有关更多信息，请参阅监控工具。
- 使用 API 时缺少下行链路消息包

  使用 SendDataToWirelessDevice API 操作时，API 会返回唯一的 MessageId。不过，其无法确认 LoRaWAN 设备是否收到下行链路消息。在设备尚未完成联接过程等情况下，下行链路数据包可能遭到丢弃。有关如何解决此错误的更多信息，请参阅上一节的内容。
- 发送下行链路消息时缺少 ARN 错误

  从队列向设备发送下行链路消息时，可能会出现缺少 Amazon Resource Name (ARN) 错误。出现此错误可能是因为尚未为接收下行链路消息的设备准确指定目标。若要解决此错误，请检查设备的目标详细信息。

管理来自公共 LoRaWAN 设备网络（Everynet）的 LoRaWAN 流量

您可以使用公开可用的 LoRaWAN 网络，在几分钟内将您的 LoRaWAN 设备连接到云端。适用于 LoRaWAN 的 AWS IoT Core 现在支持 Everynet 在美国和英国的网络覆盖。使用公共网络时，您每月需要为每台设备支付公共网络连接费。该定价适用于所有提供公共网络连接的 AWS 区域。有关此特征的定价的信息，请参阅 AWS IoT Core 定价页面。

> **Important**
>
> 公共网络由 Everynet 直接作为一项服务运营和提供。在使用此特征之前，请参阅适用的 AWS 服务条款。此外，如果您通过 适用于 LoRaWAN 的 AWS IoT Core 使用公共网络，则某些 LoRaWAN 设备信息（例如 `DevEUI` 和 `JoinEUI`）将在 适用于 LoRaWAN 的 AWS IoT Core 可用的区域之间复制。

适用于 LoRaWAN 的 AWS IoT Core 根据 LoRa Alliance 漫游规范支持公共 LoRaWAN 网络，如 LoRaWAN 后端接口 1.0 规范中所述。公共网络功能可用于连接家庭网络之外的终端设备。为了支持此功能，适用于 LoRaWAN 的 AWS IoT Core 与 Everynet 合作，提供更广泛的无线电覆盖范围。

使用公共 LoRaWAN 网络的优势

您的 LoRaWAN 设备可以使用公共网络连接到云端，这可以缩短部署时间，并减少维护私有 LoRaWAN 网络所需的时间和成本。

通过使用公共 LoRaWAN 网络，您将获得诸如覆盖范围扩展、在没有无线电网络的情况下运行核心以及覆盖密集化等优势。此特征可用于：

- 为设备移出家庭网络时提供覆盖范围，例如公共 LoRaWAN 网络支持架构部分所示的图中的设备 A。
- 将覆盖范围扩大到没有 LoRa 网关可供连接的设备，例如公共 LoRaWAN 网络支持架构部分所示的图中的设备 B。然后，设备可以使用由合作伙伴提供的网关连接到家庭网络。

您的 LoRaWAN 设备可以通过公共网络使用漫游功能连接到云端，这可以缩短部署时间，并减少维护私有 LoRaWAN 网络所需的时间和成本。

以下各节描述了公共网络支持架构、公共 LoRaWAN 网络支持的工作原理以及如何使用此特征。

主题

LoRaWAN 公共网络支持的工作原理

适用于 LoRaWAN 的 AWS IoT Core 根据 LoRa Alliance 规范支持被动漫游特征。使用被动漫游时，漫游过程对终端设备完全透明。在家庭网络之外漫游的终端设备可以连接到公共网络中的网关，并使用应用程序服务器交换上行链路和下行链路数据。在整个漫游过程中，设备保持与家庭网络的连接。

> **Note**
>
> 适用于 LoRaWAN 的 AWS IoT Core 仅支持被动漫游的无状态特征。不支持移交漫游。在移交漫游中，当您的设备移动到家庭网络之外时，它会切换到其他运营商。

主题

公共 LoRaWAN 网络概念

适用于 LoRaWAN 的 AWS IoT Core 支持的公共网络功能使用以下概念。

LoRaWAN 网络服务器（LNS）

LNS 是一种独立的私有服务器，可以在本地运行，也可以是基于云的服务。适用于 LoRaWAN 的 AWS IoT Core 是在云端提供服务的 LNS。

家庭网络服务器（hNS）

家庭网络是设备所属的网络。家庭网络服务器（hNS）是一个 LNS，其中 适用于 LoRaWAN 的 AWS IoT Core 用于存储设备的预置数据，例如 `DevEUI`、`AppEUI` 和会话密钥。

访问的网络服务器（vNS）

访问的网络是设备离开家庭网络时从中获得覆盖的网络。访问的网络服务器（vNS）是与 hNS 签订业务和技术协议的 LNS，可以为终端设备提供服务。AWS 合作伙伴 Everynet 充当访问的网络以提供覆盖范围。

提供服务的网络服务器（sNS）

提供服务的网络服务器（sNS）是处理设备的 MAC 命令的 LNS。一个 LoRa 会话只能有一个 sNS。

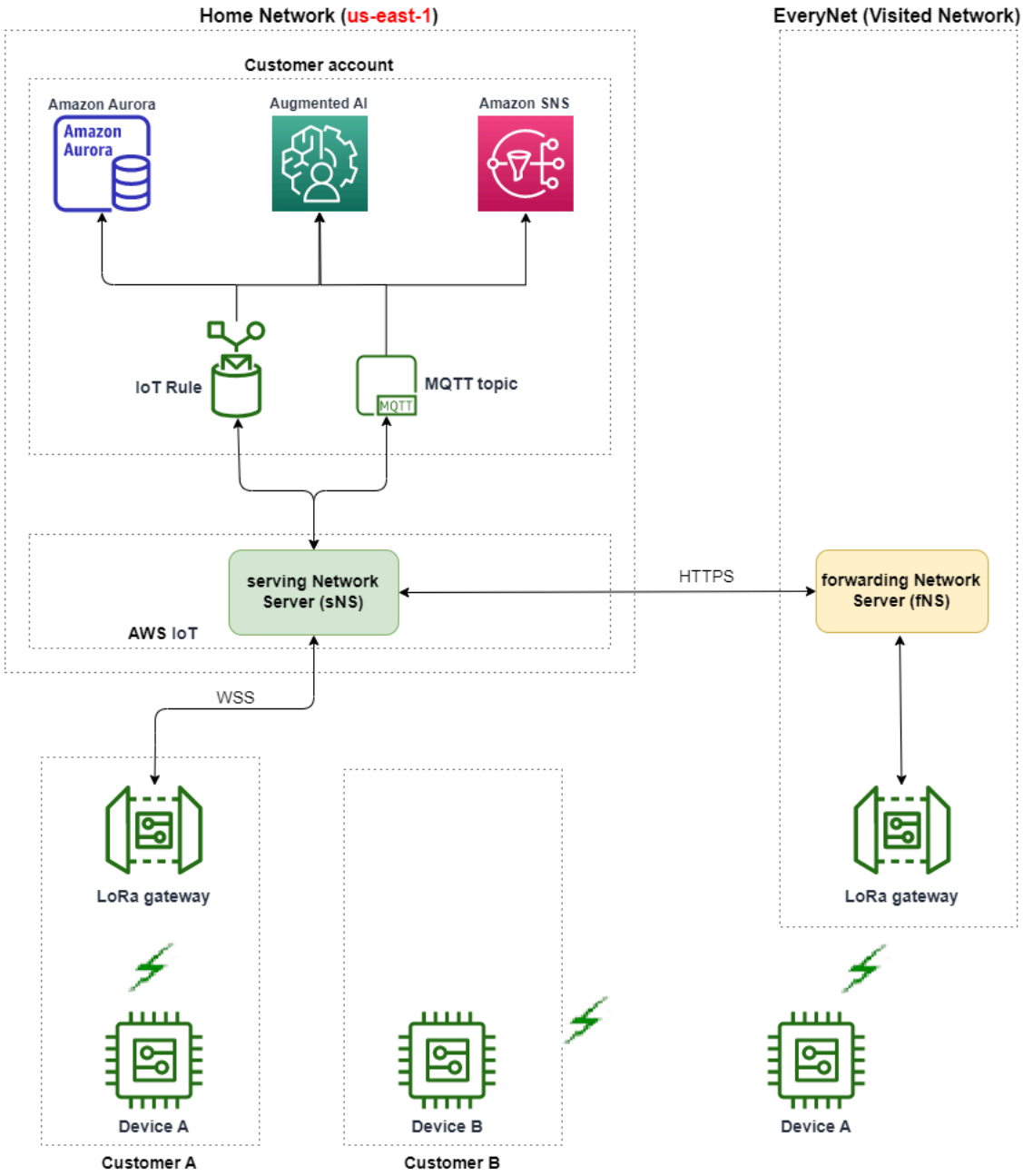
转发网络服务器（fNS）

转发网络服务器（fNS）是管理无线电网关的 LNS。一个 LoRa 会话中可以涉及零个或多个 fNS。该网络服务器管理将从设备收到的数据包转发到家庭网络的过程。

公共 LoRaWAN 网络支持架构

下面的架构图显示了 适用于 LoRaWAN 的 AWS IoT Core 如何与 Everynet 合作以提供公共网络连接。在这种情况下，设备 A 通过 LoRa 网关连接到由 适用于 LoRaWAN 的 AWS IoT Core 提供的 hNS（家庭网络服务器）。当设备 A 移出家庭网络时，它会进入访问的网络，并被 Everynet 提供的访问的网络服务器（vNS）覆盖。vNS 还将覆盖范围扩展到没有 LoRa 网关可供连接的设备 B。

您可以在 AWS IoT 控制台中查看公共网络覆盖信息，如下一节所述。



适用于 LoRaWAN 的 AWS IoT Core 根据 LoRa Alliance LoRaWAN 漫游中心技术建议使用漫游中心功能。漫游中心为 Everynet 提供一个端点，用于路由从终端设备接收到的流量。在这种情况下，Everynet 充当转发网络服务器（fNS），以转发从设备收到的流量。它使用由 LoRa Alliance 规范定义的 HTTP RESTful API。

> **Note**
>
> 如果您的设备从其家庭网络移出，并进入您的家庭网络和 Everynet 都可以提供覆盖的位置，它会使用先到先得的策略来确定是连接到 LoRa 网关，还是连接到 Everynet 的网关。

访问公共网络时，hNS 和提供服务的网络服务器（sNS）是分开的。然后，在 sNS 与 hNS 之间交换上行链路和下行链路数据包。

如何使用公共网络支持

要启用 Everynet 的公共网络支持，请在创建服务配置文件时启用某些漫游参数。在此测试版中，这些参数在您使用 AWS IoT Wireless API 或 AWS CLI 时可用。以下各节显示了必须启用的参数以及如何使用 AWS CLI 启用公共网络。

> **Note**
>
> 只有在创建新的服务配置文件时，才能启用公共网络支持。您无法使用这些参数更新现有配置文件以启用公共网络。

主题

漫游参数

在为设备创建服务配置文件时，请指定以下参数。在从 AWS IoT 控制台的配置文件中心添加服务配置文件时，请指定这些参数，也可以使用 AWS IoT Wireless API 操作 `CreateServiceProfile` 或 AWS CLI 命令 `create-service-profile`。

> **Note**
>
> 适用于 LoRaWAN 的 AWS IoT Core 不支持移交漫游。创建服务配置文件时，您无法启用指定是否使用移交漫游的 `HrAllowed` 参数。

- 允许漫游激活（RaAllowed）：此参数指定是否启用漫游激活。漫游激活使终端设备能够在 vNS 的覆盖范围内激活。使用漫游特征时，RaAllowed 必须设置为 true。
- 允许被动漫游（PrAllowed）：此参数指定是否启用被动漫游。使用漫游特征时，PrAllowed 必须设置为 true。

为设备启用公共网络支持

要在设备上启用公共 LoRaWAN 网络支持，请运行以下过程。

> **Note**
>
> 只能为 OTAA 设备启用公共网络功能。使用 ABP 作为激活方法的设备不支持此特征。

1. 使用漫游参数创建服务配置文件

   通过启用漫游参数创建服务配置文件。

   > **Note**
   >
   > 在为要与该服务配置文件关联的设备创建设备配置文件时，我们建议您为 RxDelay1 参数指定一个较大的值，至少大于 2 秒。

   - 使用 AWS IoT控制台

     转到 AWS IoT 控制台的配置文件中心，然后选择添加服务配置文件。创建配置文件时，选择启用公共网络。

   - 使用 AWS IoT Wireless API

     要在创建服务配置文件时启用漫游功能，请使用 CreateServiceProfile API 操作或 create-service-profile CLI 命令，如下例所示。

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

     运行此命令将返回服务配置文件的 ARN 和 ID 作为输出。

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

2. 检查服务配置文件中的漫游参数

要检查您指定的漫游参数，可以在控制台中或使用 `get-service-profile` CLI 命令查看服务配置文件，如下例所示。

- 使用 AWS IoT控制台

  转到 AWS IoT 控制台的配置文件中心，然后选择您创建的配置文件。在详细信息页面的配置文件配置选项卡中，您将看到 RaAllowed 和 PrAllowed 设置为 `true`。

- 使用 AWS IoT Wireless API

  要查看您启用的漫游参数，请使用 GetServiceProfile API 操作或 `get-service-profile` CLI 命令，如下例所示。

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

运行此命令会将服务配置文件详细信息作为输出返回，包括漫游参数 `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. 将服务配置文件附加到设备

将您使用漫游参数创建的服务配置文件附加到终端设备。您也可以创建设备配置文件并为无线设备添加目标。您将使用此目标路由从您的设备发送的上行链路消息。有关创建设备配置文件和目标的更多信息，请参阅添加设备配置文件和将目标添加到 适用于 LoRaWAN 的 AWS IoT Core。

- 登记新设备

  如果您尚未登记设备，则可以指定此服务配置文件，以便在将设备添加 适用于 LoRaWAN 的 AWS IoT Core 时使用。以下命令显示如何使用 `create-wireless-device` CLI 命令，通过您创建的服务配置文件的 ID 添加设备。有关使用控制台添加服务配置文件的信息，请参阅使用控制台将您的无线设备规范添加到 适用于 LoRaWAN 的 AWS IoT Core。

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

  下面显示的是 *createdevice.json* 文件的内容。

  createdevice.json 的内容

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

运行此命令的输出将生成无线设备的 ARN 和 ID 作为输出。

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

- 更新现有设备

  如果您已经登记了设备，则可以更新现有的无线设备以使用此服务配置文件。以下命令显示如何使用 `update-wireless-device` CLI 命令，通过您创建的服务配置文件的 ID 更新设备。

```
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"
```

  此命令不会生成任何输出。您可以使用 `GetWirelessDevice` API 或 `get-wireless-device` CLI 命令来获取更新的信息。

4. 使用 Everynet 将设备连接到云端

   由于已启用漫游，您的设备现在必须执行联接才能获得新的 `DevAddr`。当您使用 OTAA 时，LoRaWAN 设备会发送联接请求，网络服务器将允许该请求。然后，它可以使用 Everynet 提供的网络覆盖范围连接到 AWS Cloud。有关如何为设备执行激活过程或联接的说明，请参阅设备文档。

> **Note**
>
> - 只能针对使用 OTAA 作为激活方式的设备，启用漫游功能并连接到公共网络。不支持 ABP 设备。有关如何为设备执行激活过程或联接的说明，请参阅设备文档。请参阅 激活模式。

- 要禁用设备的漫游功能，您可以取消设备与该服务配置文件的关联，然后将其与漫游参数设置为 `false` 的另一个服务配置文件进行关联。切换到此服务配置文件后，您的设备必须再次加入，这样它们就不会继续在公共网络上运行。

5. 交换上行链路和下行链路消息

在将设备联接到 适用于 LoRaWAN 的 AWS IoT Core 后，即可开始在设备和云端之间交换消息。

- 查看上行链路消息

  当您从设备发送上行链路消息时，适用于 LoRaWAN 的 AWS IoT Core 使用您之前配置的目标将这些消息传送到您的 AWS 账户。这些消息将通过 Everynet 的网络从您的设备发送到云端。

  您可以使用 AWS IoT 规则名称查看消息，也可以使用 MQTT 客户端订阅在创建目标时指定的 MQTT 主题。有关规则名称和您指定的其他目标详细信息的更多信息，请参阅使用控制台添加一个目标。

  有关查看上行链路消息和格式的更多信息，请参阅查看从 LoRaWAN 设备发送的上行链路消息的格式。

- 发送下行链路消息

  您可以从控制台对下行链路消息排队并向设备发送这些消息，也可以使用 AWS IoT Wireless API 命令 `SendDataToWirelessDevice` 或 AWS CLI 命令 `send-data-to-wireless-device` 执行这一过程。有关对下行链路消息排队和发送这些消息的信息，请参阅管理要发送到 LoRaWAN 设备的下行链路消息队列。

  以下代码显示了如何使用 `send-data-to-wireless-device` CLI 命令发送下行链路消息的示例。您可以指定接收数据的无线设备的 ID、有效负载、是否使用确认模式以及无线元数据。

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

  运行此命令的输出会生成下行链路消息的 `MessageId`。

> **Note**
>
> 在某些情况下，即使收到 MessageId，数据包也可能丢失。有关对此类场景进行问题排查和解决问题的信息，请参阅排查下行链路消息队列错误。

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

- 查看覆盖范围信息

  启用公共网络后，可以在 AWS IoT 控制台中查看网络覆盖范围信息。转至 AWS IoT 控制台的覆盖范围中心，然后搜索位置以在地图上查看设备的覆盖范围信息。

  > **Note**
  >
  > 此特征使用 Amazon Location Service 在 Amazon Location 地图上显示您设备的覆盖范围信息。使用 Amazon Location 地图之前，请查看 Amazon Location Service 的条款和条件。请注意，AWS 可能会将您的 API 查询传输给您选择的第三方数据提供商，这些提供商可能不在您当前使用的 AWS 区域范围内。有关更多信息，请参阅 AWS 服务条款。

对 LoRaWAN 设备和多播组执行无线固件更新（FUOTA）

您可以通过无线方式执行固件更新，以更新单个 LoRaWAN 设备或一组设备的设备固件。要更新设备固件或向多台设备发送下行链路有效负载，请创建一个多播组。借助多播，源可以将数据发送到单个多播组，然后分发到整组接收设备。

适用于 LoRaWAN 的 AWS IoT Core 对 FUOTA 和多播组的支持基于 LoRa Alliance 的以下规范：

- LoRaWAN 远程组播设置规范，TS005-2.0.0
- loraWAN 碎片化数据块传输规范，TS004-2.0.0
- LoRaWAN 应用层时钟同步规范，TS003-2.0.0

> **Note**
>
> 适用于 LoRaWAN 的 AWS IoT Core 根据 LoRa Alliance 规范自动执行时钟同步。它使用的是函数 AppTimeReq 将服务器端时间回复使用 ClockSync 信令请求时间的设备。

以下主题将介绍如何创建多播组并执行 FUOTA。

主题

为组播和 FUOTA 配置准备设备

将无线设备添加到 适用于 LoRaWAN 的 AWS IoT Core 时，您可以使用控制台或 CLI 准备无线设备进行多播设置和 FUOTA 配置。如果您是第一次执行这一配置，我们建议您使用控制台。要管理多播组并从组中添加或删除多个设备，我们建议使用 CLI 管理大量资源。

GenAppKey 和 FPort

添加无线设备时，请先配置下列参数，然后再将您的设备添加到多播组或执行 FUOTA。在配置这些参数之前，请确保您的设备支持 FUOTA 和组播，并且无线设备规格为 OTAA v1.1 或 OTAAv1.0.x。

- GenAppKey：对于支持 LoRaWAN 1.0.x 版本的设备以及要使用多播组的设备，GenAppKey 是具体设备的根密钥，从中派生多播组的会话密钥。

  > **Note**
  >
  > 对于使用无线规格 OTAA v1.1 的 LoRaWAN 设备，AppKey 的用途和 GenAppKey 相同。

  要设置参数来启动数据传输，适用于 LoRaWAN 的 AWS IoT Core 与终端设备一起分发会话密钥。有关 LoRaWAN 版本的更多信息，请参阅 LoRaWAN 版本。

> **Note**
>
> 适用于 LoRaWAN 的 AWS IoT Core 存储了您以加密格式提供的 GenAppKey 信息。

- FPorts：根据对 FUOTA 和多播组的 LoRaWAN 规范，适用于 LoRaWAN 的 AWS IoT Core 为 FPorts 参数的以下字段分配默认值。如果您已经分配了下列任意 FPort 值，您可以选择一个不同的可用值，从 1 到 223。
  - Multicast：200

    该 FPort 值用于多播组。
  - FUOTA：201

    该 FPort 值用于 FUOTA。
  - ClockSync：202

    该 FPort 值用于时钟同步。

多播和 FUOTA 的设备配置文件

在多播会话开始时，使用 B 类或 C 类分发窗口向组中的设备发送下行链路消息。为多播和 FUOTA 添加的设备必须支持 B 类或 C 类操作模式。根据您的设备支持的设备类别，选择启用 B 类或 C 类模式之一或两种模式的设备配置文件。

有关设备配置文件的更多信息，请参阅 将配置文件添加到 适用于 LoRaWAN 的 AWS IoT Core。

使用控制台为多播和 FUOTA 做好设备准备

要使用控制台指定多播设置的 FPort 和 GenAppKey 参数和 FUOTA，请执行以下操作：

1. 导航到 AWS IoT 控制台的设备中心然后选择添加无线设备。
2. 选择无线设备规格。您的设备必须使用 OTAA 进行激活。选择 OTAA v1.0.x 或 OTAA v1.1 时，出现 FUOTA 配置-可选。
3. 输入无线设备的 EUI（扩展唯一标识符）参数。
4. 展开FUOTA 配置-可选部分然后选择该设备支持无线固件更新 (FUOTA)。现在，您可以输入多播、FUOTA 和时钟同步的 FPort 值。如果针对无线设备规格您选择了 OTAA v1.0.x，请输入 GenAppKey。

5. 通过选择您的个人资料和路由消息目标，将您的设备添加到 适用于 LoRaWAN 的 AWS IoT Core。对于链接到设备的设备配置文件，请确保选择支持 B 类或支持 C 类模式，或选择两种模式。

> **Note**
>
> 要指定 FUOTA 配置参数，您必须使用 AWS IoT 控制台的设备中心。如果您使用 AWS IoT 控制台的简介页面来登记设备，则不会出现这些参数。

有关无线设备规格和登记设备的更多信息，请参阅 将您的无线设备添加到 适用于 LoRaWAN 的 AWS IoT Core。

> **Note**
>
> 您只能在创建无线设备时指定这些参数。更新现有设备时无法更改或指定参数。

使用 API 操作为多播和 FUOTA 准备设备

要使用多播组或执行 FUOTA，请使用 `CreateWirelessDevice` API 操作或 `create-wireless-device` CLI 命令确认这些参数。除了指定应用程序密钥和 FPort 参数之外，请确保链接到设备的设备配置文件支持 B 类或 C 类模式或两者都支持。

您可以将 `input.json` 文件作为 `create-wireless-device` 命令的输入。

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

其中：

input.json 的内容

```
{
    "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
}
```

有关您可以使用的 CLI 的信息，请参阅 AWS CLI 参考。

> **Note**
>
> 指定这些参数的值之后，无法通过 UpdateWirelessDevice API 操作进行更新。相反，您可以使用参数 GenAppKey 和 FPorts 的值创建新的设备。

要获取这些参数指定值的信息，可以使用 GetWirelessDevice API 操作或 get-wireless-device CLI 命令。

后续步骤

配置参数后，您可以创建多播组和 FUOTA 任务来发送下行链路有效负载或更新 LoRaWAN 设备的固件。

- 有关创建多播组的更多信息，请参阅 创建多播组并将设备添加到组。
- 有关创建 FUOTA 任务的更多信息，请参阅 创建 FUOTA 任务并提供固件映像。

创建多播组向多台设备发送下行链路有效负载

要向多台设备发送下行链路有效负载，请创建一个多播组。借助多播，源可以将数据发送到单个多播地址，然后分发到整组接收设备。

多播组中的设备共享同一个多播地址、会话密钥和帧计数器。通过使用相同的会话密钥，多播组中的设备可以在启动下行链路传输时解密消息。多播组仅支持下行链路。它不能确认设备是否接收了下行链路有效负载。

使用 适用于 LoRaWAN 的 AWS IoT Core 多播组，您可以：

- 使用设备配置文件、RFRegion 或设备类别筛选您设备列表，然后将这些设备添加到多播组中。
- 在 48 小时的分发窗口内，调度一条或多条下行负载消息并向多播组中的设备发送一条或多条下行负载消息。
- 让设备在组播会话开始时临时切换到 B 类或 C 类模式接收下行链路消息。
- 监控多播组的设置及其设备的状态，并对发现的问题进行故障排除。
- 使用无线固件更新 (FUOTA) 将固件更新安全地部署到多播组中的设备。

下面的视频介绍了如何创建 适用于 LoRaWAN 的 AWS IoT Core 多播组，并指导您完成将设备添加到多播组并调度下行链路消息到该组的整个过程。

下面介绍了如何创建多播组并调度下行链路消息。

主题

创建多播组并将设备添加到组

您可以使用控制台或 CLI 创建多播组。如果您是第一次创建多播组，我们建议您使用控制台添加多播组。如果要管理多播组并在组中添加或删除设备时，可以使用 CLI。

在与添加的终端设备交换信令之后，适用于 LoRaWAN 的 AWS IoT Core 与终端设备建立共享密钥并设置数据传输的参数。

先决条件

在创建多播组并将设备添加到组之前，请执行以下操作：

- 通过指定 FUOTA 配置参数 `GenAppKey` 和 `FPorts`，为设备多播和 FUOTA 设置做好准备。有关更多信息，请参阅为组播和 FUOTA 配置准备设备。

- 检查设备是否支持 B 类或 C 类操作模式。根据您的设备支持的设备类别，选择支持 B 类或支持 C 类或两者都支持的的设备配置文件。有关设备配置文件的更多信息，请参阅 将配置文件添加到 适用于 LoRaWAN 的 AWS IoT Core。

  在多播会话开始时，使用 B 类或 C 类分发窗口向组中的设备发送下行链路消息。

使用控制台查创建多播组

要使用控制台创建多播组，请转至 AWS IoT 控制台的 Multicast groups（多播组）页面并选择 Create multicast group（创建多播组）。

1. 创建多播组

   要创建您的多播组，请为组指定多播属性和标签。

   1. 指定多播属性

      要指定多播属性，请为多播组输入下列信息。

      - 名称：为您的多播组输入唯一的名称。名称只能包含字母、数字、连字符和下划线。它不能包含空格。
      - 说明：您可以为多播组提供可选描述。描述的长度最多为 2,048 个字符。

   2. 多播组的标签

      您可以选择提供任何键值作为多播组的 标签。要继续创建多播组，请选择 Next（下一步）。

2. 将设备添加到多播组

   您可以将单个设备或一组设备添加到多播组中。要添加设备：

   1. 指定 RFRegion

      为您的多播组指定RfRegion或频段。的您的多播组RfRegion必须匹配您在多播组中添加设备的RfRegion。有关 RfRegion 的更多信息，请参阅 为您的网关和设备连接选择 LoRa 频带。

   2. 选择多播设备类别

      在多播会话开始时，选择您希望多播组中的设备切换到 B 类还是 C 类模式。B 类会话可以在常规的下行链路槽接收下行链路消息，C 类会话可以随时接收下行链路消息。

   3. 选择您想要添加在组中的设备。

      选择您要在多播组中单独添加设备还是批量添加设备。

- 要单独添加设备，请输入要添加到组的每台设备的无线设备 ID。
- 要批量添加设备，您可以按设备配置文件或标签筛选要添加的设备。对于设备配置文件，您可以添加带支持 B 类、C 类或支持两者的配置文件的设备。

4. 要创建多播组，请选择 Create （创建）。

   组 中出现多播组详细信息和您添加的设备 。有关多播组和设备的状态以及对任何问题的故障排除信息，请参阅 监控多播组和组中设备的状态并对进行故障排除。

创建多播组后，您可以选择 Action（操作）编辑、删除或将设备添加到多播组中。添加设备后，您可以调度会话，将下行链路负载发送到组中的设备。

使用 API 创建多播组

要使用 API 创建多播组并将设备添加到组中，请执行以下操作：

1. 创建多播组

   要创建多播组，请使用CreateMulticastGroup API 操作或 create-multicast-groupCLI 命令。您可以将 input.json 文件作为 create-multicast-group 命令的输入。

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

   其中：

   input.json 的内容

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

   创建多播组后，您可以使用下列 API 操作或 CLI 命令来更新、删除或获取有关多播组的信息。

- UpdateMulticastGroup 或 update-multicast-group

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

2. 将设备添加到多播组

   您可以将设备单独添加或批量添加到多播组。

   - 要将设备批量添加到多播组，请使用 StartBulkAssociateWirelessDeviceWithMulticastGroup API 操作或 start-bulk-associate-wireless-device-with-multicast-groupCLI 命令。要筛选批量添加到多播组的设备，请提供查询字符串。下面说明了如何通过相关的特定 ID 添加一组具有配置文件的设备。

     ```
     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 的内容

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

     在这里，multicast-groups/d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf/bulk 是用于将设备与组进行关联的 URL。

   - 要将设备单独添加到多播组中，请使用 AssociateWirelessDeviceWithMulticastGroup API 操作或 associate-wireless-device-with-multicast-group CLI。为要添加到组的每个设备提供无线设备 ID。

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

创建多播组后，您可以使用以下 API 操作或 CLI 命令获取有关多播组的信息或取消设备的关联。

- 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 操作可用于列出一般的无线设备，以及与多播组或 FUOTA 任务关联的无线设备。
>
> - 要列出与多播组关联的无线设备，请将 ListWirelessDevices API 操作和 MulticastGroupID 用作筛选器。
> - 要列出与 FUOTA 任务关联的无线设备，请将 ListWirelessDevices API 操作和 FuotaTaskID 用作筛选器。

后续步骤

创建多播组并添加设备后，您可以继续添加设备并监控多播组和设备的状态。如果您的设备已成功添加到组中，可以配置并调度向设备发送下行链路消息。在发送下行链路消息之前，设备的状态必须为多播设置已就绪。调度下行链路消息后，状态将变为尝试会话。有关更多信息，请参阅调度向多播组中的设备发送下行链路消息。

如果要更新多播组中设备的固件，可以通过以下方式用 适用于 LoRaWAN 的 AWS IoT Core 执行无线固件更新 (FUOTA)。有关更多信息，请参阅对 适用于 LoRaWAN 的 AWS IoT Core 设备进行无线固件更新（FUOTA）。

如果没有添加设备，或如果在多播组或设备状态中看到错误，可以将鼠标悬停在错误上获取更多信息并解决问题。如果错误仍然存在，有关如何排查和解决问题的信息，请参阅 监控多播组和组中设备的状态并对进行故障排除。

监控多播组和组中设备的状态并对进行故障排除

添加设备并创建多播组后，请打开 AWS Management Console。导航到 AWS IoT 控制台Multicast groups（多播组）的页面并选择您创建的多播组查看详细信息。您将看到有关多播组的信息、已添加的设备数量以及 设备 状态的详细信息。您可以使用状态信息跟踪多播会话的进度并对任何错误进行故障排除。

多播组状态

您的多播组可以在 AWS Management Console 中显示如下状态消息。

- 待定

  此状态表示您已经创建了多播组，但还没有多播会话。创建群组后，您将看到显示此状态消息。在此期间，您可以更新多播组，并将设备与组关联或取消关联。在状态从 Pending（待定）更改后，其他设备不能添加到组中。
- 尝试会话

  您的设备成功添加到多播组之后，如果您的组有已调度的多播会话，您将看到显示此状态消息。在此期间，您无法更新或将设备添加到您的多播组。如果取消了多播会话，组状态将更改为 Pending（待定）。
- 在会话中

  出现最早多播会话时间时，您将看到显示此状态消息。当多播组与 FUOTA 任务（正在进行的固件更新会话）相关联时，多播组会保持在这个状态。

  如果会话中没有关联的 FUOTA 任务，且由于会话时间超时而取消多播会话或您取消了多播会话，则组状态将更改为 Pending（待定）。
- 等待删除

  如果删除多播组，组状态将更改为 Delete waiting（等待删除）。删除操作是永久性的，无法撤消。此操作可能需要时间，在多播组删除之前，组的状态是 Delete_Waiting（等待删除）。多播组进入此状态后，无法转换到其他状态。

多播组中设备的状态

多播组中的设备可以在 AWS Management Console 中显示下列状态消息。您可以将鼠标悬停在每条状态消息上获取表示内容的更多信息

- 尝试程序包

设备与多播组关联后，设备状态为尝试程序包。这种状态表明 适用于 LoRaWAN 的 AWS IoT Core 尚未确认设备是否支持多播设置和操作。

- 不支持的程序包

  在您的设备与多播组关联后，适用于 LoRaWAN 的 AWS IoT Core 检查设备的固件是否能够进行多播设置和操作。如果您的设备没有支持的多播软件包，其状态为不支持程序包。要纠正错误，请检查设备的固件是否能够进行多播设置和操作。

- 尝试多播设置

  如果与多播组关联的设备能够进行多播设置和操作，状态为尝试多播设置。此状态表示设备尚未完成组播设置。

- 多播设置就绪

  您的设备已完成组播设置并已添加到多播组中。此状态表示设备已准备好进行多播会话，并且可以向这些设备发送下行链路消息。状态还表示您可以使用 FUOTA 更新组中设备固件的时间。

- 尝试会话

  已为多播组中的设备调度了多播会话。在多播组会话开始时，设备状态为 Session attempting（尝试会话），并且请求已发送，以了解是否可以为会话启动 B 类或 C 类分发窗口。如果设置多播会话所需的时间超时或取消了多播会话，状态将更改为多播设置完成。

- 在会话中

  此状态表示 B 类或 C 类分发窗口已启动，您的设备有正在进行的多播会话。在此期间，适用于 LoRaWAN 的 AWS IoT Core 中的下行链路消息可以发送到多播组中的设备。如果更新会话时间，会覆盖当前会话，状态更改为尝试会话。当会话时间结束或取消多播会话时，状态将更改为多播设置就绪。

后续步骤

现在，您已经了解了多播组和组中设备的不同状态，以及如何解决问题（例如设备无法设置多播时），您可以计划向设备发送下行链路消息，然后您的多播组将处于会话中的状态。有关调度下行链路消息的信息，请参阅 调度向多播组中的设备发送下行链路消息。

调度向多播组中的设备发送下行链路消息

成功将设备添加到多播组后，您可以启动多播会话并配置要发送到这些设备的下行链路消息。必须在 48 小时之内调度下行链路消息，组播的开始时间必须至少比当前时间晚 30 分钟。

> **Note**
>
> 多播组中的设备无法确认收到下行链路消息的时间。

先决条件

在发送下行链路消息之前，您必须已创建多播组并成功将设备添加到要发送下行链路消息的组中。在为多播会话调度了开始时间后，您无法添加更多的设备。有关更多信息，请参阅创建多播组并将设备添加到组。

如果没有成功添加任何设备，多播组和设备状态将包含帮助您纠正错误的信息。如果错误仍然存在，有关纠正这些错误的信息，请参阅 监控多播组和组中设备的状态并对进行故障排除。

使用控制台调度下行链路消息

要使用控制台发送下行链路消息，请转至 AWS IoT 控制台的多播组的页面，然后选择您创建的多播组。在多播组详细信息页面中，选择调度下行链路消息然后选择调度下行会话。

1. 调度下行链路消息窗口

   您可以设置一个时间窗口，以便向多播组中的设备发送下行链路消息。必须在 48 小时内调度下行链路消息。

   要调度多播会话，请指定以下参数：

   - Start date（开始日期）和 Start time（开始时间）：开始日期和时间必须至少为当前时间之后的 30 分钟和当前时间之前的 48 小时。

     > **Note**
     >
     > 您指定的时间是 UTC，因此在调度下行链路窗口时，请考虑检查与时区的时差。

   - 会话超时：如果没有收到下行链路消息，您希望多播会话超时之后的时间。最短超时时间为 60 秒。B 类多播组的最长超时时间为 2 天，C 类多播组的最长超时时间为 18 小时。

2. 配置下行链路消息

   要配置下行链路消息，请指定以下参数：

   - 数据速率：为下行链路消息选择数据速率。数据速率取决于 RFRegion 和有效载荷大小。US915 区域的默认数据速率为 8，EU868 区域的默认数据速率为 0。

- 频率：选择发送下行链路消息的频率。为避免消息冲突，请根据 RFRegion 选择可用频率。
- FPort：选择可用的频率端口将下行链路消息发送到设备。
- 有效负载：根据数据速率指定的最大有效负载。使用默认数据速率，您可以在 US915 RFRegion 中拥有 33 个字节的最大有效负，在 EU868 RFRegion 中为 51 个字节。使用较大的数据速率，您可以传输最多 242 字节的有效负载。

要调度下行链路消息，请选择 Schedule（调度）。

使用 API 调度下行链路消息

要使用 API 调度下行链路消息，请使用 StartMulticastGroupSession API 操作或 start-multicast-group-sessionCLI 命令。

您可以使用以下 API 操作或 CLI 命令获取有关多播组的信息并删除多播组。

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

要在会话启动后向多播组发送数据，请使用SendDataToMulticastGroup API 操作或 send-data-to-multicast-groupCLI 命令。

后续步骤

将下行链路消息配置为发送到设备后，该消息将在会话开始时发送。多播组中的设备无法确认是否已收到消息。

配置其他下行链路消息

您还可以配置其他下行链路消息发送到多播组中的设备：

- 要从控制台配置其他下行链路消息：
  1. 转至 AWS IoT 控制台的 Multicast groups（多播组）的页面，然后选择您创建的多播组。
  2. 在多播组详细信息页面中，选择 Schedule downlink message（调度下行链路消息）然后选择 Configure additional downlink message（配置额外的下行链路消息）。
  3. 指定 Data rate（数据速率）、Frequency（频率）、FPort 和 Payload（有效负载）等参数，类似于为第一条下行链路消息配置这些参数的方式。

- 要使用 API 或 CLI 配置其他下行链路消息，请调用 SendDataToMulticastGroup API 操作或 send-data-to-multicast-group每条附加下行链路消息的 CLI 命令。

更新会话调度

您还可以更新会话调度，为多播会话使用新的开始日期和时间。新的会话调度将覆盖之前调度的会话。

> **Note**
>
> 仅在需要时更新您的多播会话。这些更新可能会导致一组设备长时间唤醒并耗尽电池。

- 要从控制台更新会话调度：
  1. 转至 AWS IoT 控制台的 Multicast groups（多播组）的页面，然后选择您创建的多播组。
  2. 在多播组详细信息页面中，选择 Schedule downlink message（安排下行链路消息）然后选择Update session schedule（更新会话调度）。
  3. 指定状态日期、开始时间和会话超时等参数，类似于您为第一条下行链路消息指定这些参数的方式。
- 要在 API 或 CLI 更新会话调度，请使用StartMulticastGroupSession API 操作或 start-multicast-group-sessionCLI 命令。

对 适用于 LoRaWAN 的 AWS IoT Core 设备进行无线固件更新（FUOTA）

使用无线固件更新 (FUOTA) 将固件更新部署到 适用于 LoRaWAN 的 AWS IoT Core 设备。

借助 FUOTA，您可以将固件更新发送到单个设备或一组设备。您还可以通过创建多播组向多台设备发送固件更新。首先将设备添加到多播组，然后将固件更新映像发送到所有这些设备。我们建议您对固件映像进行数字签名，以便接收映像的设备可以验证它们来自正确的来源。

借助 适用于 LoRaWAN 的 AWS IoT Core 的 FUOTA，您可以：

- 将新的固件映像或 delta 映像部署到单个设备、一组设备中。
- 在新固件部署到设备之后，验证其真实性和完整性。
- 监控部署进度并在部署失败的情况下调试问题。

适用于 LoRaWAN 的 AWS IoT Core 对 FUOTA 和多播组的支持基于 LoRa Alliance 的以下规范：

- LoRaWAN 远程组播设置规范，TS005-2.0.0
- loraWAN 碎片化数据块传输规范，TS004-2.0.0
- LoRaWAN 应用层时钟同步规范，TS003-2.0.0

> **Note**
>
> 适用于 LoRaWAN 的 AWS IoT Core 根据 LoRa Alliance 规范自动执行时钟同步。它使用的是函数 `AppTimeReq` 将服务器端时间回复使用 ClockSync 信令请求时间的设备。

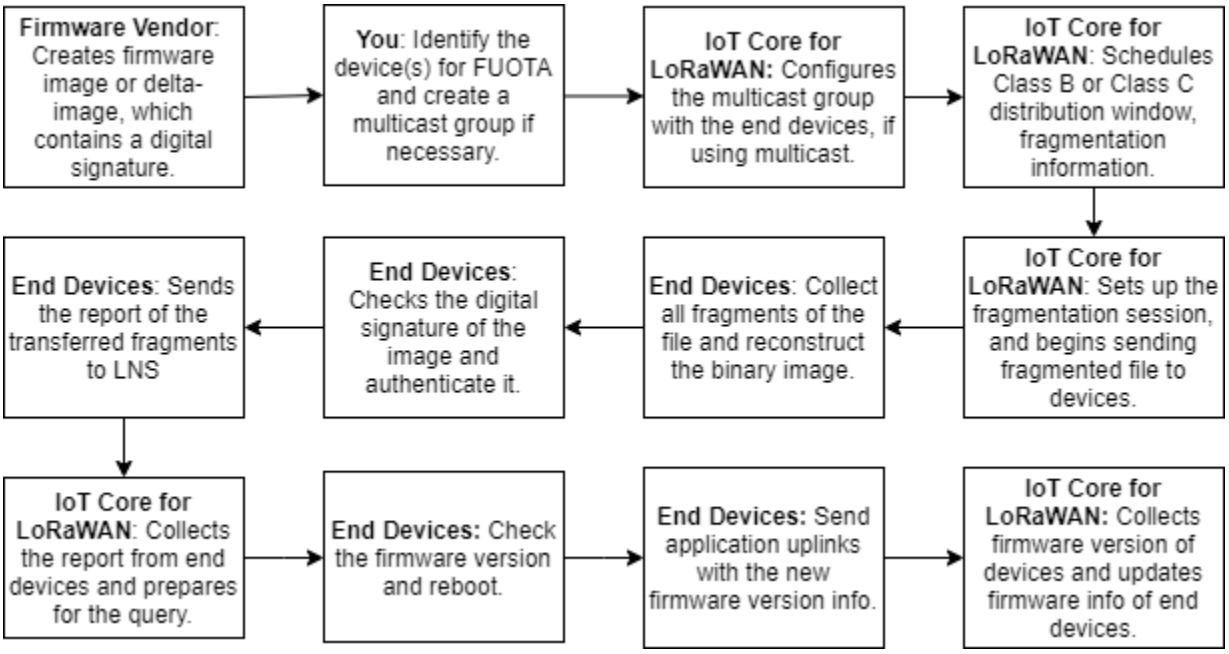
以下视频描述了如何创建 适用于 LoRaWAN 的 AWS IoT Core FUOTA 任务，并引导您完成向任务中添加设备和安排 FUOTA 任务的过程。

以下主题演示了如何执行 FUOTA。

- FUOTA 过程概述
- 创建 FUOTA 任务并提供固件映像
- 将设备和多播组添加到 FUOTA 任务并调度 FUOTA 会话
- 监控 FUOTA 任务和添加到该任务的设备的状态并进行故障排除

FUOTA 过程概述

下图说明了 适用于 LoRaWAN 的 AWS IoT Core 如何为终端设备执行 FUOTA 流程。如果要向 FUOTA 会话添加单个设备，可以跳过创建和配置多播组的步骤。您可以直接将设备添加到 FUOTA 会话中，然后 适用于 LoRaWAN 的 AWS IoT Core 将启动固件更新过程。



要对设备执行 FUOTA，请首先创建数字签名固件映像，然后配置要添加到 FUOTA 任务中的设备和多播组。启动 FUOTA 会话后，您的终端设备会收集所有碎片，从碎片中重建映像，将状态报告至 适用于 LoRaWAN 的 AWS IoT Core，然后应用新的固件映像。

下面说明了 FUOTA 流程的不同步骤：

1. 使用数字签名创建固件映像或增量映像

   要通过 适用于 LoRaWAN 的 AWS IoT Core 对 LoRaWAN 设备执行 FUOTA，建议您在通过无线方式发送固件更新时对固件映像或增量映像进行数字签名。然后，接收映像的设备可以验证映像来自正确的来源。

   固件映像的大小不得超过 1 兆字节。固件越大，更新过程完成所需的时间就越长。要更快地传输数据，或如果您的新映像大于 1 兆字节，请使用增量映像，这是新映像的一部分，即新固件映像和上一个映像之间的增量。

   > **Note**
   >
   > 适用于 LoRaWAN 的 AWS IoT Core 不提供数字签名生成工具和固件版本管理系统。您可以使用任何第三方工具为固件映像生成数字签名。建议您使用数字签名工具，例如嵌入 ARM mbed GitHub 存储库的工具，其中还包括用于生成增量映像和设备使用该映像的工具。

2. 识别和配置 FUOTA 的设备

   识别 FUOTA 的设备后，将固件更新发送到单个或多个设备。

   - 要将固件更新发送到多台设备，请创建多播组，然后使用终端设备配置多播组。有关更多信息，请参阅创建多播组向多台设备发送下行链路有效负载。
   - 要向各个设备发送固件更新，请将这些设备添加到 FUOTA 会话中，然后执行固件更新。

3. 调度分发窗口并设置碎片会话

   如果创建了多播组，可以指定 B 类或 C 类分发窗口，确定设备可以从 适用于 LoRaWAN 的 AWS IoT Core 接收碎片的时间。在切换到 B 类或 C 类模式之前，您的设备可能在 A 类中运行。您还必须指定会话的开始时间。

   B 类或 C 类设备会在指定的分发窗口唤醒，然后开始接收下行链路数据包。在 C 类模式下运行的设备可以消耗比 B 类设备更多的电量。有关更多信息，请参阅设备类。

4. 终端设备将状态报告至 适用于 LoRaWAN 的 AWS IoT Core 并更新固件映像

   设置碎片会话后，您的终端设备和 适用于 LoRaWAN 的 AWS IoT Core 执行以下步骤以更新设备的固件。

   1. 由于 LoRaWAN 设备的数据速率较低，要启动 FUOTA 流程，适用于 LoRaWAN 的 AWS IoT Core 设置碎片会话以对固件映像进行碎片化。 然后 会将这些碎片发送到终端设备。
   2. 适用于 LoRaWAN 的 AWS IoT Core 发送映像碎片后，LoRaWAN 终端设备执行以下任务。
      a. 收集碎片，然后从这些碎片中重建二进制图像。
      b. 检查重建后映像的数字签名以验证映像并验证来自正确的来源。
      c. 比较来自的固件版本 适用于 LoRaWAN 的 AWS IoT Core 转至当前版本。
      d. 报告转移到 适用于 LoRaWAN 的 AWS IoT Core 的碎片映像状态，然后应用新的固件映像。

      > **Note**
      > 在某些情况下，在检查固件映像的数字签名之前，终端设备报告传输到 适用于 LoRaWAN 的 AWS IoT Core 的碎片映像的状态。

现在您已经了解了 FUOTA 流程，您可以创建 FUOTA 任务并将设备添加到任务中以更新固件。有关更多信息，请参阅创建 FUOTA 任务并提供固件映像。

创建 FUOTA 任务并提供固件映像

要更新 LoRaWAN 设备的固件，首先创建一个 FUOTA 任务并提供要用于更新的数字签名固件映像。然后，您可以将设备和多播组添加到任务中并调度 FUOTA 会话。会话开始时，适用于 LoRaWAN 的 AWS IoT Core 设置碎片会话，最终设备收集碎片、重建映像并应用新固件。有关 FUOTA 过程的更多信息，请参阅 FUOTA 过程概述。

下面说明了如何创建 FUOTA 任务并上传要存储在 S3 存储桶中的固件映像或增量映像。

先决条件

在执行 FUOTA 之前，必须对固件映像进行数字签名，这样您的终端设备才可以在应用映像时验证映像的真实性。您可以使用任何第三方工具为固件映像生成数字签名。建议您使用数字签名工具，例如嵌入 ARM mbed GitHub 存储库的工具，其中还包括用于生成增量映像和设备使用该映像的工具。

使用控制台创建 FUOTA 任务并上传固件映像

要创建 FUOTA 任务并使用控制台上传固件映像，请转到控制台的 FUOTA 任务 的选项卡，然后选择 Create FUOTA task（创建 FUOTA 任务）。

1. 创建 FUOTA 任务

   要创建 FUOTA 任务，请指定任务属性和标签。

   1. 指定 FUOTA 任务属性

      要指定 FUOTA 任务属性，请为 FUOTA 任务输入以下信息。

      - 名称：为 FUOTA 任务输入唯一的名称。名称只能包含字母、数字、连字符和下划线。它不能包含空格。
      - 说明：您可以为多播组提供可选描述。说明字段最多为 2,048 个字符。
      - RFRegion：为 FUOTA 任务设置频段。频段必须与您用于配置无线设备或多播组的频段相匹配。

   2. FUOTA 任务的标签

      您可以选择提供任何键值对作为的 FUOTA 任务的标签。要继续创建您的映像，选择 Next(下一步)。

2. 上传固件映像

选择要用来更新添加到 FUOTA 任务设备固件的固件映像文件。固件映像文件存储在 S3 存储桶中。您可以为 适用于 LoRaWAN 的 AWS IoT Core 提供代表您访问固件映像的权限。我们建议您对固件映像进行数字签名，以便在执行固件更新时验证其真实性。

1. 选择固件映像文件

您可以将新的固件映像文件上传到 S3 存储桶，也可以选择已上传到 S3 存储桶的现有映像。

> **Note**
> 固件映像文件不得超过 1 兆字节。固件越大，更新过程完成所需的时间就越长。

- 要使用现有映像，选择 Select an existing firmware image（选择现有的固件映像），选择 Browse S3（浏览 S3），然后选择要使用的固件映像文件。

  适用于 LoRaWAN 的 AWS IoT Core 填充 S3 URL，该 URL 是 S3 存储桶中固件映像文件的路径。`s3://bucket_name/file_name` 路径的格式为：要在 Amazon Simple Storage Service控制台中查看文件，选择 View（查看）。
- 要上传新的固件映像。
  a. 选择 Upload a new firmware image（上载新的固件映像），然后上载固件映像。映像文件不得大于 1 兆字节。
  b. 要创建 S3 存储桶并输入要存储固件映像文件的Bucket 名称，请选择 Create S3 bucket（创建 S3 存储桶）。

2. 要访问存储桶的权限

您可以创建新的服务角色或选择现有角色，让 适用于 LoRaWAN 的 AWS IoT Core 代表您访问 S3 存储桶中的固件映像文件。选择下一步。

要创建新角色，您可以输入角色名称或将其留白以便自动生成随机名称。要查看授予对 S3 存储桶访问权限的策略权限，选择 View policy permissions（查看策略权限）。

有关使用 S3 存储桶存储映像和授予 适用于 LoRaWAN 的 AWS IoT Core 访问权限的更多信息，请参阅 将固件文件上载到 S3 存储桶并添加 IAM 角色。

3. 审核和创建

   要创建 FUOTA 任务，请查看您指定的 FUOTA 任务和配置详细信息，然后选择 Create task(创建任务)。

使用 API 创建 FUOTA 任务并上传固件映像

要使用 API创建 FUOTA 任务并指定固件映像文件，请使用 CreateFuotaTask API 操作或 create-fuota-taskCLI 命令。您可以将 input.json 文件作为 create-fuota-task 命令的输入。使用 API 或 CLI 时，作为输入提供的固件映像文件必须已上传到 S3 存储桶。您还可以指定让 适用于 LoRaWAN 的 AWS IoT Core 访问S3 存储桶中固件映像的 IAM 角色。

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

其中：

input.json 的内容

```
{
   "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"
}
```

创建 FUOTA 任务后，您可以使用以下 API 操作或 CLI 命令来更新、删除或获取有关 FUOTA 任务的信息。

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

后续步骤

现在，您已创建了 FUOTA 任务并提供了固件映像，您可以将设备添加到任务中更新固件。您可以将单个设备或多播组添加到任务中。有关更多信息，请参阅将设备和多播组添加到 FUOTA 任务并调度 FUOTA 会话。

将设备和多播组添加到 FUOTA 任务并调度 FUOTA 会话

创建 FUOTA 任务后，您可以将设备添加到要更新固件的任务中。将设备成功添加到 FUOTA 任务后，您可以调度 FUOTA 会话更新设备固件。

- 如果您只有少数设备，可以将这些设备直接添加到 FUOTA 任务中。
- 如果有大量设备 要更新固件 ，您可以将这些设备添加到多播组中，然后将多播组添加到 FUOTA 任务中。有关创建和使用多播组的信息，请参阅 创建多播组向多台设备发送下行链路有效负载。

> **Note**
>
> 您可以将单个设备或多播组添加到 FUOTA 任务中。您不能将设备和多播组添加到任务中。

添加设备或多播组后，您可以启动固件更新会话。适用于 LoRaWAN 的 AWS IoT Core 收集固件映像，对映像进行碎片化，然后以加密格式存储碎片。您的终端设备收集碎片并应用新的固件映像。固件更新所需的时间取决于映像的大小和碎片方式。固件更新完成后，适用于 LoRaWAN 的 AWS IoT Core 存储的固件映像的加密片段已删除。您仍然可以在 S3 存储桶中找到固件映像。

先决条件

在将设备或多播组添加到 FUOTA 任务之前，请执行以下操作。

- 您必须已创建 FUOTA 任务并提供固件映像。有关更多信息，请参阅创建 FUOTA 任务并提供固件映像。
- 配置要更新设备固件的无线设备。有关登记设备的更多消息，请参阅 将您的设备登记到 适用于 LoRaWAN 的 AWS IoT Core。
- 要更新多台设备的固件，您可以将它们添加到多播组。有关更多信息，请参阅创建多播组向多台设备发送下行链路有效负载。
- 您将设备登记到 适用于 LoRaWAN 的 AWS IoT Core 时，指定 FUOTA 配置参数 `FPorts`。如果您使用的是 LoRaWan v1.0.x 设备，您还必须指定 `GenAppKey`。有关 FUOTA 配置参数的更多信息，请参阅 为组播和 FUOTA 配置准备设备。

将设备添加到 FUOTA 任务并使用控制台调度 FUOTA 会话

要使用控制台添加设备或多播组并调度 FUOTA 会话，请转到控制台的 FUOTA 任务选项卡。然后，选择要向其添加设备的 FUOTA 任务并执行固件更新。

添加设备和多播组

1. 您可以将单个设备或多播组添加到 FUOTA 任务中。但是，您不能将单个设备和多播组添加到同一个 FUOTA 任务中。要使用控制台添加设备，请执行以下操作。

   1. 在FUOTA 任务详细信息中，选择 Add device（添加设备）。
   2. 选择您添加到任务中的设备频段或 RFRegion。此值必须与您为 FUOTA 任务选择的RFRegion匹配。
   3. 选择是否要将单个设备或多播组添加到任务。
      - 要添加单个设备，请选择 Add individual devices （添加单个设备），然后输入要添加到 FUOTA 任务中的设备 ID。
      - 要添加多播组，请选择 Add multicast groups（添加多播组），然后将多播组添加到任务中。您可以使用设备配置文件或标签筛选要添加到任务中的多播组。按设备配置文件进行筛选时，可以选择带配置文件设备的多播组，已启用支持 B 类或支持 C 类。
2. 调度 FUOTA 会话

   成功添加设备或多播组后，您可以调度 FUOTA 会话。要调度会话，请执行以下操作。

   1. 选择要更新其设备固件的 FUOTA 任务，然后选择 Schedule FUOTA session.（调度 FUOTA 会话）
   2. 为您的 FUOTA 会话指定开始日期和开始时间。确保开始时间是当前时间 30 分钟之后。

将设备添加到 FUOTA 任务并使用 API 调度 FUOTA 会话

您可以使用 AWS IoT Wireless API 或 CLI 将无线设备或多播组添加到 FUOTA 任务中。然后，您可以调度 FUOTA 会话。

1. 添加设备和多播组

   您可以将无线设备或多播组与 FUOTA 任务关联起来。

   - 要将各单台设备与 FUOTA 任务关联，请使用 `AssociateWirelessDeviceWithFuotaTask` API 操作或 `associate-wireless-device-with-fuota-task` CLI 命令，然后将 `WirelessDeviceID` 作为输入。

```
aws iotwireless associate-wireless-device-with-fuota-task \
    --id "01a23cde-5678-4a5b-ab1d-33456808ecb2"
    --wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
```

- 要将多播组与 FUOTA 任务关联，请使用 AssociateMulticastGroupWithFuotaTask API 操作或 associate-multicast-group-with-fuota-task CLI 命令，然后将 MulticastGroupID 作为输入。

```
aws iotwireless associate-multicast-group-with-FUOTA-task \
    --id 01a23cde-5678-4a5b-ab1d-33456808ecb2"
    --multicast-group-id
```

将无线设备或多播组与 FUOTA 任务关联后，请使用以下 API 操作或 CLI 命令列出您的设备或多播组或将它们与任务取消关联。

- 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：
>
> - MulticastGroupID 作为筛选条件时，ListWirelessDevices 可以列出一般无线设备以及与多播组关联的设备。FuotaTaskID 作为筛选条件时，API 列出了与 FUOTA 任务关联的无线设备。
> - FuotaTaskID 作为筛选条件时，ListMulticastGroups 可以在下列情况下列出一般多播组和与 FUOTA 任务关联的多播组。

2. 调度 FUOTA 会话

设备或多播组成功添加到 FUOTA 任务后，您可以启动 FUOTA 会话更新设备固件。开始时间离当前时间必须至少有 30 分钟。要使用 API 或 CLI 调度 FUOTA 会话，请使用StartFuotaTask API 操作或 start-fuota-taskCLI 命令。

启动 FUOTA 会话后，您无法再向任务添加设备或多播组。您可以使用 GetFuotaTask API 操作或 get-fuota-taskCLI 命令来获取有关 FUOTA 会话状态的信息。

监控 FUOTA 任务和添加到该任务的设备的状态并进行故障排除

在配置无线设备并创建了可能要使用的任何多播组之后，可以通过执行以下步骤启动 FUOTA 会话。

FUOTA 任务状态

您的 FUOTA 任务可以在 AWS Management Console 中显示下列状态消息。

- 待定

  此状态表示您已经创建了 FUOTA 任务，但还没有固件更新会话。创建任务后，您将看到显示此状态消息。在此期间，您可以更新 FUOTA 任务， 关联 或者取消设备或多播组与任务的关联。在状态从Pending（待定）更改后，无法将其他设备添加到任务中。

- 等待 FUOTA 会话

  您的设备已经成功 添加 到 FUOTA 任务，当您的任务有已调度固件更新会话时，您将看到显示此状态消息。在此期间，您无法更新或将设备添加到 FUOTA 会话。如果您取消 FUOTA 会话，群组状态将更改为 Pending（待定。）

- 在 FUOTA 会话中

  当 FUOTA 会话开始时，您将看到显示此状态消息。碎片会话开始，您的终端设备收集碎片、重建固件映像、将新固件版本与原始版本进行比较，然后应用新映像。

- Fuota 已完成

  在您的终端设备向 适用于 LoRaWAN 的 AWS IoT Core 报告已应用新的固件映像，或会话超时时，FUOTA 会话被标记为已完成，您将看到显示此状态。

  在以下任何情况下，您还会显示看到此状态，因此请务必检查固件更新是否已正确应用于设备。

  - 当 FUOTA 任务状态为等待 FUOTA 会话，并且存在 S3 存储桶错误时，例如指向 S3 存储桶中映像文件的链接不正确或 适用于 LoRaWAN 的 AWS IoT Core 没有足够的权限来访问存储桶中的文件。
  - 当 FUOTA 任务状态为等待 FUOTA 会话，并且有启动 FUOTA 会话的请求时，但是没有从 FUOTA 任务中的设备或多播组收到响应。

  - 当 FUOTA 任务状态为 FUOTA 会话中时，并且设备或多播组在一段时间内没有发送任何碎片，这导致会话超时。
- 等待删除

  如果删除处于任何其他状态的 FUOTA 任务，会显示此状态。这是永久性操作，无法撤消。此操作可能需要时间，在 FUOTA 任务删除之前，任务状态为 Delete waiting（删除等待）。FUOTA 任务进入此状态后，无法转换到其他状态。

FUOTA 任务中设备的状态

FUOTA 任务中的设备可以在 AWS Management Console 中显示下列状态消息。您可以将鼠标悬停在每条状态消息上获取表示内容的更多信息

- 初次

  在 FUOTA 会话的开始时间，适用于 LoRaWAN 的 AWS IoT Core 检查您的设备是否有固件更新支持的软件包。如果您的设备有支持的软件包，该设备的 FUOTA 会话将启动。固件映像已碎片化，碎片将发送到您的设备。看到显示此状态时，表示设备的 FUOTA 会话尚未启动。
- 不支持的程序包

  如果设备不支持 FUOTA 软件包，您将看到显示此状态。如果不支持固件更新包，设备的 FUOTA 会话将无法启动。要纠正该错误，请检查设备的固件是否可以使用 FUOTA 接收固件更新。
- 不受支持碎片算法

  FUOTA 会话开始时，适用于 LoRaWAN 的 AWS IoT Core 为设备设置碎片会话。如果您看到显示此状态，表示使用的碎片算法类型无法应用于设备的固件更新。出现错误的原因是您的设备没有支持的 FUOTA 软件包。要纠正该错误，请检查设备的固件是否可以使用 FUOTA 接收固件更新。
- 内存不足

  适用于 LoRaWAN 的 AWS IoT Core 发送映像碎片之后，您的终端设备收集映像碎片并从这些碎片中重建二进制映像。当设备没有足够的内存来组装固件映像的传入碎片时，会显示此状态，这可能会导致固件更新会话过早结束。要纠正错误，请检查设备的硬件是否可以接收此更新。如果您的设备无法接收此更新，请使用增量映像更新固件。
- 不受支持碎片索引

  碎片索引标识了四个同时可能出现的碎片会话中的一个。如果您的设备不支持指定的碎片索引值，会显示此状态。要纠正这个错误，可以执行下列操作。
  - 为设备启动新的 FUOTA 任务。

  - 如果错误仍然存在，请将从单播模式切换到多播模式。
  - 如果错误仍未纠正，请检查设备固件。
- 内存错误

  此状态表示您的设备在接收来自 适用于 LoRaWAN 的 AWS IoT Core 传入的片段时出现了内存错误。如果发生此错误，您的设备可能无法接收此更新。要纠正错误，请检查设备的硬件是否可以接收此更新。如果需要，请使用增量映像更新设备固件。
- 错误描述符

  您的设备不支持指定的描述符。描述符是描述在碎片会话期间传输的文件的字段。如果您看到此错误，请联系 AWS Support 中心。
- 会话计数重播

  此状态表示您的设备之前使用过此会话计数。要纠正该错误，请启动设备的 FUOTA 新任务。
- 缺少碎片

  当您的设备从 适用于 LoRaWAN 的 AWS IoT Core 中收集图像碎片时，它从独立的编码碎片中重建新的固件映像。如果您的设备尚未收到所有碎片，无法重建新映像，您将看到此状态。要纠正该错误，请启动设备的 FUOTA 新任务。
- MIC 错误

  当您的设备从收集的碎片中重建新固件映像时，会执行 MIC（消息完整性检查）来验证映像的真实性以及是否来自正确的来源。如果您的设备在重新组装碎片后检测到麦克风中不匹配，会显示此状态。要纠正该错误，请启动设备的 FUOTA 新任务。
- 成功

  您的设备 FUOTA 会话成功。

> **Note**
>
> 尽管此状态消息表示设备已从碎片中重建映像并进行了验证，但设备将状态报告到时，设备固件可能尚未更新到 适用于 LoRaWAN 的 AWS IoT Core。检查您的设备固件是否已更新。

后续步骤

您已经了解了 FUOTA 任务及其设备的不同状态以及如何解决问题。有关每个状态的更多信息，请参阅 loraWAN 碎片化数据块传输规范，TS004-1.0.0。

使用网络分析器实时监控无线资源机群

网络分析器使用默认 WebSocket 连接接收无线连接资源的实时跟踪消息日志。通过使用网络分析器，您可以添加要监控的资源、激活跟踪消息传递会话以及开始实时接收跟踪消息。

要监控您的资源，还可以使用 Amazon CloudWatch。要使用 CloudWatch，您需要设置 IAM 角色来配置日志记录，然后等待日志条目在控制台中显示。网络分析器显著缩短了建立连接和开始接收跟踪消息所需的时间，为您的资源机群提供了即时日志信息。有关使用 CloudWatch 监控的信息，请参阅 使用 Amazon CloudWatch Logs 监控 AWS IoT Wireless 资源。

通过缩短设置时间并使用跟踪消息中的信息，您可以更有效地监控资源、获得有意义的见解并排查错误。您可以监控 LoRaWAN 设备和 LoRaWAN 网关。例如，在登记其中一台 LoRaWAN 设备时，您可以快速识别加入错误。要调试错误，请使用提供的跟踪消息日志中的信息。

如何使用网络分析器

要监控资源机群并开始接收跟踪消息，请执行以下步骤

1. 创建网络分析器配置并添加资源

   在激活跟踪消息收发之前，请创建网络分析器配置并向配置添加资源。首先，指定配置设置，其中包括日志级别和无线设备帧信息。然后，通过使用无线网关和无线设备标识符添加要监控的无线资源。

2. 使用 WebSockets 流式传输跟踪消息

   您可以使用 IAM 角色的凭据生成预签名的请求 URL，使用 WebSocket 协议流式传输网络分析器跟踪消息。

3. 激活跟踪消息收发会话并监控跟踪消息

   要开始接收跟踪消息，请激活跟踪消息会话。为了避免产生额外费用，您可以停用或关闭网络分析器跟踪消息会话。

下面的视频不仅会介绍 适用于 LoRaWAN 的 AWS IoT Core 网络分析器的工作原理，还会全程指导您使用网络分析器添加资源和跟踪加入活动。

下面的主题介绍了如何创建配置、添加资源和激活跟踪消息收发会话。

主题

- 为网络分析器添加必要的 IAM 角色

- 创建网络分析器配置并添加资源
- 使用 WebSockets 串流网络分析器跟踪消息
- 实时查看和监控网络分析器跟踪消息日志
- 使用网络分析器对多播组和 FUOTA 任务进行调试和故障排除

为网络分析器添加必要的 IAM 角色

使用网络分析器时，必须授予用户使用 API 操作 UpdateNetworkAnalyzerConfiguration 和 GetNetworkAnalyzerConfiguration 访问网络分析器资源的权限。下面显示了您用于授予权限的 IAM 策略。

网络分析器的 IAM 策略

请使用以下任一项：

- 完全访问无线策略

  通过将策略 AWSIoTWirelessFullAccess 附加到您的角色，授予 适用于 LoRaWAN 的 AWS IoT Core 完全访问策略。有关更多信息，请参阅 AWSIoTWirelessFullAccess 策略摘要。
- 用于获取和更新 API 的限定范围的 IAM 策略

  通过转到 IAM 控制台的 Create policy（创建策略）页面，并在 Visual editor（可视化编辑器）选项卡上创建以下 IAM 策略：

  1. 对于 Service（服务），选择 IoTWireless。
  2. 在 Access level（访问级别）下，展开 Read（读取）并选择 GetNetworkAnalyzerConfiguration，然后展开 Write（写入）并选择 UpdateNetworkAnalyzerConfiguration。
  3. 选择 Next:Tags（下一步：标签），然后输入策略的 Name（名称），例如 IoTWirelessNetworkAnalyzerPolicy。选择创建策略。

  以下内容显示您创建的策略 IoTWirelessNetworkAnalyzerPolicy。有关创建策略的更多信息，请参阅创建 IAM 策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
```

```
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

用于访问特定资源的限定范围的策略

要配置更精细的访问控制，您必须将无线网关和设备添加到 Resource（资源）字段中。以下策略使用通配符 ARN 授予对所有网关和设备的访问权限。您可以使用 `WirelessGatewayId` 和 `WirelessDeviceId` 来控制对特定网关和设备的访问权限。

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

要授予用户使用网络分析器但不使用任何无线网关或设备的权限，请使用以下策略。除非指定，否则隐式拒绝使用资源的权限。

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

```
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": [
                "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
            ]
        }
    ]
}
```

后续步骤

现在您已创建策略，您可以将资源添加到网络分析器配置中，并接收这些资源的跟踪消息收发信息。有关更多信息，请参阅创建网络分析器配置并添加资源。

创建网络分析器配置并添加资源

在流式传输跟踪消息之前，请创建网络分析器配置并将要监控的资源添加到此配置中。创建配置时，您可以：

- 指定配置名称和可选说明。
- 自定义配置设置，例如帧信息和日志消息的详细级别。
- 确定您要监控的资源。资源可以是无线网关和/或无线设备。

您指定的配置设置将决定您将针对添加到配置的资源收到的跟踪消息收发信息。您可能还需要根据监控用例创建多个配置。

下面的内容说明了如何创建配置和添加资源。

主题

- 创建网络分析器配置
- 添加资源并更新网络分析器配置

创建网络分析器配置

必须先创建网络分析器配置，然后才能监控无线网关或无线设备。创建配置时，只需指定配置名称。即使在创建配置之后，您也可以自定义配置设置并将希望监控的资源添加到配置中。配置设置决定了您将针对这些资源收到的跟踪消息收发信息。

根据要监控的资源以及希望为其接收的信息级别，您可能需要创建多个配置。例如，您可以创建一个仅显示您的 AWS 账户中一组网关的错误信息的配置。您还可以创建一个配置，以显示有关要监控的无线设备的所有信息。

以下各节介绍了各种配置设置以及如何创建配置。

配置设置

在创建或更新网络分析器配置时，还可以自定义以下参数以筛选日志流信息。

- 帧信息

  此设置是跟踪消息的无线设备资源的帧信息。帧信息可用于调试网络服务器与终端设备之间的通信。该功能默认已启用。

- 日志级别

  您可以查看信息或错误日志，也可以关闭日志记录。

  - 信息

    日志级别为 Info（信息）的日志更详细，其中包含错误日志流和信息日志流。信息日志可用于查看设备或网关状态的变化。

    > **Note**
    >
    > 收集更多详细的日志流可能会产生额外的成本。有关定价的更多信息，请参阅 AWS IoT Core 定价。

  - 错误

    日志级别为 Error 的日志不那么详细，只显示错误信息。当应用程序出错（例如设备连接错误）时，可以使用这些日志。通过使用日志流中的信息，您可以识别机群中资源的错误并对其进行故障排除。

使用控制台创建配置

通过使用 AWS IoT 控制台或 AWS IoT Wireless API，您可以创建网络分析器配置和自定义可选参数。还可以创建多个配置，之后删除任何不再使用的配置。

创建网络分析器配置

1. 打开 AWS IoT 控制台的网络分析器中心并选择 Create configuration（创建配置）。
2. 指定配置设置。
   - 名称、描述和标签

     指定一个唯一的 Configuration name（配置名称），其中仅包含字母、数字、连字符或下划线。使用可选的 Description（说明）字段以提供有关配置的信息，并使用 Tags（标签）字段以添加有关配置的元数据的键值对。有关命名和描述资源的更多信息，请参阅 描述您的 AWS IoT Wireless 资源。
   - 配置设置

     选择是否禁用帧信息，并使用选择日志级别来选择希望用来跟踪消息日志的日志级别。选择下一步。
3. 向配置添加资源。您可以立即添加资源，也可以选择 Create（创建）并在以后添加您的资源。要以后添加资源，请选择 Create（创建）。

   在 Network Analyzer hub page（网络分析器中心页面）中，您将看到创建的配置及其设置。要查看新配置的详细信息，请选择配置名称。

删除网络分析器配置

您可以创建多个网络分析器配置，具体取决于要监视的资源以及希望针对这些资源接收的跟踪消息收发信息的级别。

从控制台中删除配置

1. 转至 AWS IoT 控制台的网络分析器中心，然后选择要删除的配置。
2. 选择操作，然后选择删除。

使用 API 创建控制台

要使用 API 创建网络分析器配置，请使用 CreateNetworkAnalyzerConfiguration API 操作或 create-network-analyzer-configuration CLI 命令。

创建配置时，只需指定配置名称。还可以使用此 API 操作在创建配置时指定配置设置和添加资源。或者，您可以在以后使用 UpdateNetworkAnalyzerConfiguration API 操作或 update-network-analyzer-configuration CLI 来指定它们。

- 创建配置

  创建配置时，必须指定名称。例如，以下命令通过仅提供名称和可选描述来创建配置。默认情况下，配置激活了帧信息，并使用 INFO 日志级别。

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_Network_Analyzer_Config \
    --description "My first network analyzer configuration"
```

  运行此命令将显示网络分析器配置的 ARN 和 ID。

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

- 使用资源创建配置

  要自定义这些配置设置，请使用 trace-content 参数。要添加资源，请使用 WirelessDevices 和 WirelessGateways 参数指定要添加到配置中的网关和/或设备。例如，以下命令自定义配置设置并将无线资源（由其 WirelessGatewayID 和 WirelessDeviceID 指定）添加到配置中。

```
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"
```

  下面的示例显示了运行此命令的输出：

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

列出网络分析器配置

您可以创建多个网络分析器配置，具体取决于要监视的资源以及要针对这些资源接收的跟踪消息收发信息的详细程度。创建这些配置后，可以使用 ListNetworkAnalyzerConfigurations API 操作或 list-network-analyzer-configuration CLI 命令以获取这些配置的列表。

```
aws iotwireless list-network-analyzer-configurations
```

运行此命令会显示您的 AWS 账户 中的所有网络分析器配置。也可以使用 `max-results` 参数来指定要显示多少个配置。下面的内容显示运行此命令的输出。

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

删除网络分析器配置

您可以通过 DeleteNetworkAnalyzerConfiguration API 操作或 delete-network-analyzer-configuration CLI 命令删除不再使用的配置。

```
aws iotwireless delete-network-analyzer-configuration \
```

```
    --configuration-name My_NetworkAnalyzer_Config
```

运行此命令不会生成任何输出。要查看可用配置，可以使用 `ListNetworkAnalyzerConfigurations` API 操作。

后续步骤

现在您已创建网络分析器配置，可以向配置中添加资源或更新配置设置。有关更多信息，请参阅添加资源并更新网络分析器配置。

添加资源并更新网络分析器配置

在激活跟踪消息传递之前，您必须向配置中添加资源。您只能使用单个默认的网络分析器配置。适用于 LoRaWAN 的 AWS IoT Core 将名称 NetworkAnalyzerConfig_Default 分配给此配置，因此该字段无法编辑。当您在控制台使用网络分析器时，此配置会自动添加到您的 AWS 账户 中。

您可以将想要监控的资源添加到此原定设置配置中。资源可以是 LoRaWAN 设备和 LoRaWAN 网关之一，也可以是两者。要将每个单独的资源添加到配置中，请使用无线网关和无线设备标识符。

配置设置

要配置设置，首先将资源添加到原定设置配置中，然后激活跟踪消息。收到跟踪消息日志后，您还可以自定义以下参数更新原定设置配置并筛选日志流。

- 帧信息

  此设置是跟踪消息的无线设备资源的帧信息。预设情况下，帧信息处于启用状态，可用于调试网络服务器与终端设备之间的通信。
- 日志级别

  您可以查看信息或错误日志，也可以关闭日志记录。
  - 信息

    日志级别为 Info 的日志更详细，包含信息丰富和错误的日志流。信息日志可用于查看设备或网关状态的更改。

    > **Note**
    >
    > 收集更多详细的日志流可能会产生额外的成本。有关定价的更多信息，请参阅 AWS IoT Core 定价。

- 错误

  日志级别为 Error 的日志不那么详细，只显示错误信息。当应用程序出错（例如设备连接错误）时，可以使用这些日志。通过使用日志流中的信息，您可以识别机群中资源的错误并对其进行故障排除。

先决条件

在添加资源之前，必须将您要监控的网关和设备登记到 适用于 LoRaWAN 的 AWS IoT Core。有关更多信息，请参阅将网关和设备连接到 适用于 LoRaWAN 的 AWS IoT Core。

使用控制台添加资源并更新网络分析器配置

通过使用 AWS IoT 控制台或 AWS IoT Wireless API，您可以添加资源和自定义可选参数。除了资源之外，您还可以编辑配置设置并保存更新的配置。

向配置添加资源（控制台）

1. 打开 AWS IoT控制台的网络分析器中心然后选择网络分析器配置，Network Analyszerconfig_DEFAULT。
2. 选择 Add resource (添加资源)。
3. 使用无线网关和无线设备标识符添加要监控的资源。您可以添加最多 250 个无线网关或无线设备。添加资源：
   a. 使用查看网关或查看设备选项卡查看已添加到 AWS 账户的网关和设备列表。
   b. 复制要监控的设备或网关的 `WirelessDeviceID`或 `WirelessGatewayID`，然后输入相应资源的标识符值。
   c. 要继续添加资源，请选择 Add gateway （添加网关）或 Add device（添加设备），然后添加无线网关或设备。如果您添加了不再想监控的资源，请选择 Remove resource（删除资源）。
4. 在添加所有资源后，请选择Add（添加）。

   您将在网络分析器中心页面中看到添加的网关和设备的数量。在激活跟踪消息会话之前，您仍然可以继续添加网关和设备。激活会话后，要添加资源，您必须停用会话。

编辑网络分析器配置（控制台）

您还可以编辑网络分析器配置，并选择是否禁用帧信息和跟踪消息日志的日志级别。

1. 打开 AWS IoT控制台的网络分析器中心然后选择网络分析器配置，Network Analyszerconfig_DEFAULT。
2. 选择编辑。
3. 选择是否禁用帧信息并使用选择日志级别来选择希望用来跟踪消息日志的日志级别。选择保存。

   您将看到在网络分析器配置的详细信息页面中指定的配置设置。

使用 API 添加资源并更新网络分析器配置

您可以使用 AWS IoT Wireless API 操作或 AWS IoT Wireless CLI 命令添加资源并更新网络分析器配置的配置设置。

- 要添加资源和更新网络分析器配置，请使用 UpdateNetworkAnalyzerConfiguration API 或 update-network-analyzer-configuration CLI。
  - 添加资源

    对于要添加的无线设备，请使用 `WirelessDevicesToAdd`输入设备的 `WirelessDeviceID`，作为字符串数组。对于要添加的无线网关，请使用 `WirelessGatewaysToAdd`输入 网关 `WirelessGatewayID`，作为字符串数组。
  - 编辑配置

    要编辑网络分析器配置，请使用 `TraceContent`参数来指定是否 `WirelessDeviceFrameInfo`应为 `ENABLED`或 `DISABLED`，以及 `LogLevel`参数是否应为 `INFO`、`ERROR` 或者 `DISABLED`。

```
{
   "TraceContent": {
      "LogLevel": "string",
      "WirelessDeviceFrameInfo": "string"
   },
   "WirelessDevicesToAdd": [ "string" ],
   "WirelessDevicesToRemove": [ "string" ],
   "WirelessGatewaysToAdd": [ "string" ],
   "WirelessGatewaysToRemove": [ "string" ]
}
```

- 要获取有关您添加的配置和资源的信息，请使用 GetNetworkAnalyzerConfiguration API 操作或 get-network-analyzer-configuration 命令。提供网络分析器配置的名称，`NetworkAnalyzerConfig_Default`，作为输入。

后续步骤

现在，您已经添加了资源并为配置指定了任何可选的配置设置，接下来，您可以使用 WebSocket 协议通过与 适用于 LoRaWAN 的 AWS IoT Core 建立连接来使用网络分析器。然后，您可以激活跟踪消息并开始接收资源的跟踪消息。有关更多信息，请参阅使用 WebSockets 串流网络分析器跟踪消息。

使用 WebSockets 串流网络分析器跟踪消息

当使用 WebSocket 协议时，您可以实时流式传输网络分析器跟踪消息。当您发送请求时，该服务会使用 JSON 结构响应。激活跟踪消息后，您可以使用消息日志获取有关资源的信息并排查错误。有关更多信息，请参阅WebSocket Numbers。

下面展示了如何使用 WebSockets 流式传输网络分析器跟踪消息。

主题

- 使用 WebSocket 库生成预签名请求
- WebSocket 消息和状态码

使用 WebSocket 库生成预签名请求

下面显示如何生成预签名请求，这样，您就可以使用 WebSocket 库向服务发送请求。

将 WebSocket 请求的策略添加到您的 IAM 角色

要使用 WebSocket 协议调用网络分析器，您需要将以下策略附加到发出请求的 AWS Identity and Access Management(IAM) 角色。

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

创建预签名 URL

为 WebSocket 请求构造一个 URL，该请求中包含在应用程序和网络分析器 之间建立通信所需的信息。要验证请求的标识，WebSocket 流式处理使用 Amazon 签名版本 4 流程处理签名请求。有关签名版本4的更多信息，请参阅 Amazon Web Services 一般参考中的签名 AWSAPI 请求。

要调用网络分析器，请使用 StartNetworkAnalyzerStream请求 URL。使用之前提到的 IAM 角色凭据对请求进行签名。URL的格式如下，出于可读性目的，添加了换行符。

```
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
```

对于签名版本 4 参数，请使用以下值：

- X-Amz-Algorithm – 您在签名过程中使用的算法。唯一有效值为 AWS4-HMAC-SHA256。
- X-Amz-Credential – 以斜杠（“/”）分隔的字符串，它通过将您的访问密钥 ID 和凭证范围组件串联起来而形成。凭证范围包括采用 YYYYMMDD 格式的日期、AWS 区域、服务名称和终止字符串 (aws4_request)。
- X-Amz-Date – 创建签名的日期和时间。根据 Amazon Web Services 一般参考中签名版本 4 中处理日期的说明生成日期和时间。
- X-Amz-Expires – 凭证到期之前的时间长度（以秒为单位）。最大值为 300 秒（5 分钟）。
- X-Amz-Security-Token – 临时凭证的签名版本 4 令牌。如果您指定此参数，则将其包含在规范请求中。更多信息，请参阅 AWSIdentity and Access Management 中的请求临时安全凭据。
- X-Amz-Signature – 您为请求生成的签名版本 4 签名。
- X-Amz-SignedHeaders – 在为请求创建签名时签名的标头。唯一有效值为 host。

构建请求 URL 并创建签名版本 4 签名

要构造请求的 URL 并创建签名版本 4 签名，请使用以下步骤。这些示例是伪代码。

任务 1：创建规范请求

创建一个字符串，其中包含来自标准格式的请求的信息。这可确保 AWS在收到请求时，计算出的签名与您在 任务 3：计算签名中计算出的签名相同。有关更多信息，请参阅 Amazon Web Services 一般参考中的 创建签名版本 4 的规范请求。

1. 为您应用程序中的请求定义变量。

```
# 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. 创建规范 URI（统一资源标识符）。规范 URI 是域与查询字符串之间的 URI 部分。

```
canonical_uri = "/start-network-analyzer-stream"
```

3. 创建规范标头和签名标头。请注意规范标头尾部的 \n。

   - 追加小写标头名称，后跟冒号。
   - 追加该标头的值的逗号分隔列表。请勿对有多个值的标头进行值排序。
   - 追加一个新行(\n)。

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. 将算法与哈希算法匹配。您必须使用 SHA-256。

```
algorithm = "AWS4-HMAC-SHA256"
```

5. 创建凭证范围，该范围将派生密钥范围限定为发出请求的日期以及将请求发送到的区域和服务。

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. 创建规范查询字符串。查询字符串值必须是 URL 编码，并且按名称排序。

   - 按字符代码点以升序顺序对参数名称进行排序。具有重复名称的参数应按值进行排序。例如，以大写字母 F 开头的参数名称排在以小写字母 b 开头的参数名称之前。
   - 请勿对 RFC 3986 定义的任何非预留字符进行 URI 编码，这些字符包括：A-Z、a-z、0-9、连字符 (-)、下划线 (_)、句点 (.) 和波形符 (~)。
   - 使用 %XY 对所有其他字符进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F）。例如，空格字符必须编码为 %20（不像某些编码方案那样使用“+”），扩展 UTF-8 字符必须采用格式 %XY%ZA%BC。
   - 对参数值中的任何等于 (=) 字符进行双重编码。

```
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. 创建负载的哈希。对于 GET 请求，负载为空字符串。

```
payload_hash = HashSHA256(("").Encode("utf-8")).HexDigest()
```

8. 组合所有元素以创建规范请求。

```
canonical_request = method + '\n'
   + canonical_uri + '\n'
   + canonical_querystring + '\n'
   + canonical_headers + '\n'
   + signed_headers + '\n'
   + payload_hash
```

任务 2：创建待签字符串

待签字符串包含您的请求的元信息。在计算请求签名时，您可以使用该字符串登录下一步。有关更多信息，请参阅《Amazon Web Services 一般参考》中的创建签名版本 4 签名字符串。

```
string_to_sign=algorithm + "\n"
   + amz_date + "\n"
   + credential_scope + "\n"
   + HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

任务 3：计算签名

您可以从您的 AWS秘密访问密钥派生签名密钥。为了实现更高程度的保护，派生密钥特定于日期、服务和 AWS区域。您可以使用派生密钥签名请求。有关更多信息，请参阅《Amazon Web Services 一般参考》https://docs.aws.amazon.com/general/latest/gr/sigv4-calculate-signature.html中的为 AWS 签名版本 4 计算签名。

该代码假定您实施了函数 GetSignatureKey来派生签名密钥。有关更多信息，请参阅 Amazon Web Services 一般参考中的如何为签名版本 4 派生签名密钥的示例。

函数 HMAC(key, data)表示以二进制格式返回结果的 HMAC-SHA256 函数。

```
#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
```

任务 4：将签名信息添加到请求中并创建请求 URL

在计算签名之后，将它添加到查询字符串。有关更多信息，请参阅 Amazon Web Services 一般参考中的向请求添加签名。

```
#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
```

后续步骤

现在，您可以在 WebSocket 库中使用请求 URL 向服务发出请求并观察消息。有关更多信息，请参阅WebSocket 消息和状态码。

WebSocket 消息和状态码

创建预签名请求后，您可以将请求 URL 与 WebSocket 库或适合您的编程语言的库一起使用，向服务发出请求。有关如何生成此预签请求的更多信息，请参阅 使用 WebSocket 库生成预签名请求。

WebSocket 消息

可以使用 WebSocket 协议建立双向连接 消息可以从客户端传输到服务器，从服务器传输到客户端。但是，网络分析器仅支持从服务器发送到客户端的消息。从客户端收到的任何消息都是意外的，如果从客户端收到消息，服务器将自动关闭 WebSocket 连接。

收到请求并启动跟踪消息会话时，服务器将使用 JSON 结构（即有效负载）进行响应。有关有效负载以及如何从 AWS Management Console激活跟踪消息的更多信息，请参阅 实时查看和监控网络分析器跟踪消息日志。

WebSocket 状态代码

下面显示了从服务器到客户端的通信的 WebSocket 状态码。WebSocket 状态码遵循 RFC 正常关闭连接标准。

下面显示了支持的状态代码：

- 1000

  此状态代码表示正常关闭，这意味着 WebSocket 连接已建立并且请求已完成。会话处于空闲状态时，可以观察到此状态，从而导致连接超时。

- 1002

  此状态代码表示端点由于协议错误而终止连接。

- 1003

  此状态代码表示错误状态，其中端点以无法接受的格式接收数据而终止连接。端点仅支持文本数据，如果它收到来自客户端的二进制消息或使用不受支持的格式的消息，可能会显示此状态代码。

- 1008

此状态代码表示错误状态，端点因收到违反其策略的消息而终止连接。此状态为通用状态，当其他状态代码（例如 1003 或 1009）不适用时显示。如果需要隐藏策略或授权失败（例如签名过期），您还会看到此状态显示。

- 1011

  此状态代码表示错误状态，其中服务器因遇到意外情况或内部错误而导致无法完成请求而终止连接。

后续步骤

现在您已经学会了如何生成预签名请求以及如何使用 WebSocket 连接观察来自服务器的消息，现在您可以激活跟踪消息并开始接收无线网关和无线设备资源的消息日志。有关更多信息，请参阅实时查看和监控网络分析器跟踪消息日志。

实时查看和监控网络分析器跟踪消息日志

如果已将资源添加到网络分析器配置中，可以激活跟踪消息开始接收资源的跟踪消息。您可以使用 AWS Management Console、AWS IoT Wireless API 或 AWS CLI。

先决条件

在使用网络分析器激活跟踪消息传送之前，您必须：

- 将希望监控的资源添加到默认网络分析器配置中。有关更多信息，请参阅添加资源并更新网络分析器配置。
- 通过使用 `StartNetworkAnalyzerStream`请求 URL 生成预签请求。将使用发出此请求的 AWS Identity and Access Management角色的凭据对请求进行签名。有关更多信息，请参阅创建预签名 URL。

使用控制台激活跟踪消息

要激活跟踪消息

1. 打开 AWS IoT控制台的网络分析器中心然后选择网络分析器配置，Network Analyszerconfig_DEFAULT。
2. 在网络分析仪配置的详细信息页面中，选择Activate trace messaging（激活跟踪消息），然后选择Activate（激活）。

   您将开始接收跟踪消息，其中最新的跟踪消息首先出现在控制台中。

> **Note**
>
> 消息传递会话开始后，在您停用会话或退出跟踪会话之前，接收跟踪消息可能会产生额外费用。有关定价的更多信息，请参阅 AWS IoT Core 定价。

查看和监控跟踪消息

激活跟踪消息后，将建立 WebSocket 连接，并且跟踪消息开始实时出现，首先是最新消息。您可以自定义首选项以指定要在每个页面中显示的跟踪消息的数量，并仅显示每条消息的相关字段。例如，您可以自定义跟踪消息日志以仅显示具有日志级别设置为 `ERROR`，以便您可以快速识别和调试网关的错误。跟踪消息包含以下信息。

- 消息编号：显示首先收到的最后一条消息的唯一编号。
- 资源 ID：资源的无线网关或无线设备 ID。
- 时间戳：收到消息的时间。
- 消息 ID：适用于 LoRaWAN 的 AWS IoT Core 分配给每条收到的消息的标识符。
- FPort：使用 WebSocket 连接与设备通信的频率端口。
- DevEui：无线设备的扩展唯一标识符 (EUI)。
- 资源：受监控的资源是无线设备还是无线网关。
- 事件：无线设备的日志消息的事件，可以是 Join、Rejoin、Uplink_Data、Downlink_Data 或 Registration。
- 日志级别：有关设备的 `INFO` 或 `ERROR` 日志流的信息。

网络分析器 JSON 日志消息

您还可以一次选择一条跟踪消息来查看该消息的 JSON 有效负载。根据在跟踪消息日志中选择的消息，您将在 JSON 负载中看到指示包含两部分的信息：CustomerLog 和 LoRaFrame。

CustomerLog

JSON 的客户日志部分显示接收消息的资源的类型和标识符、日志级别和消息内容。以下示例显示了 CustomerLog 日志消息。您可以使用 JSON 中的 `message`字段来获取有关错误以及如何解决错误的更多信息。

LoRaFrame

JSON 的 lorRaFrame 部分有消息 ID 并包含有关设备的物理负载和无线元数据的信息。

以下示例显示了追踪消息的结构。

```
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;
  },
};
```

回顾和后续步骤

在本节中，您查看了跟踪消息，并了解了如何使用这些信息调试错误。查看所有消息后，您可以：

- 停用跟踪消息

  为避免产生任何额外费用，您可以停用跟踪消息传送会话。取消激活会话会断开 WebSocket 连接的连接，因此您将不会收到任何其他跟踪消息。您仍然可以继续在控制台中查看现有消息。

- 编辑配置的帧信息

您可以编辑网络分析器配置，然后选择是否停用帧信息并选择消息的日志级别。在更新配置之前，请考虑停用跟踪消息传送会话。要进行这些编辑，请打开 AWS IoT控制台中的网络分析器详细信息页面，然后选择 Edit（编辑）。然后，您可以使用新的配置设置更新配置，并激活跟踪消息查看更新的消息。

- 向配置添加资源

  您还可以向网络分析器配置添加更多资源并实时监控。您最多可以合计 250 个无线网关和无线设备资源。要添加资源，请在 AWS IoT控制台的网络分析器详细信息页面上，选择 Resources（资源）选项卡，然后选择 Add resources（添加资源）。然后，您可以使用新资源更新配置并激活跟踪消息查看其他资源的更新消息。

有关通过编辑配置设置和添加资源来更新网络分析器配置的详细信息，请参阅 添加资源并更新网络分析器配置。

使用网络分析器对多播组和 FUOTA 任务进行调试和故障排除

您可以监控的无线资源包括 LoRaWAN 设备、LoRaWAN 网关和多播组。您还可以使用网络分析器来调试和排查 FUOTA 任务的任何问题。您还可以在 FUOTA 任务进行时，监控和跟踪与设置、数据传输和状态查询相关的消息。

要监控您的 FUOTA 任务，如果该任务包含多播组，则必须将多播组和组中的设备添加到网络分析器配置中。您还必须激活帧信息和组播帧信息，以跟踪在 FUOTA 任务进行过程中与多播组和设备交换的单播和组播上行和下行消息。

要监控多播组，可以将它们添加到网络分析器配置中，并使用组播帧信息，排查发送到这些组的组播下行链路消息的问题。要对尝试加入使用单播通信的组的设备进行故障排除，还必须将这些设备包括在网络分析器配置中。要仅监控与组中设备的单播通信，请激活无线设备的帧信息。这种方法可确保对多播组和加入该组的设备进行全面的监控和诊断。

以下各节介绍如何使用网络分析器对多播组和 FUOTA 任务进行调试和故障排除。

主题

调试仅包含设备的 FUOTA 任务

您可以使用网络分析器调试其中只添加了 LoRaWAN 设备的 FUOTA 任务。有关向 FUOTA 任务添加设备的信息，请参阅将设备和多播组添加到 FUOTA 任务并调度 FUOTA 会话。要调试 FUOTA 任务，请执行以下步骤：

1. 通过激活无线设备的帧信息，创建网络分析器配置，这样您就可以监控在任务进行时与设备交换的 FUOTA 上行链路和下行链路消息。
2. 使用 FUOTA 任务中设备的无线设备标识符，将这些设备添加到网络分析仪配置中。
3. 激活跟踪消息，以开始接收网络分析器配置中的设备的跟踪消息。

在跟踪消息信息的 `applicationCommandType` 列中，您将开始接收与数据传输和分段设置相关的单播下行链路消息。

> **Note**
>
> 如果您在跟踪消息表中看不到 `applicationCommandType` 列，则可以调整表设置以在表中显示该列。

您还可以在 JSON 日志消息的 WirelessMetadata > ApplicationInfo 下查看 `applicationCommandType` 和其他详细消息。

使用多播组调试 FUOTA 任务

您可以使用网络分析器调试具有多播组且已将 LoRaWAN 设备添加到该组的 FUOTA 任务。有关向 FUOTA 任务添加设备的信息，请参阅将设备和多播组添加到 FUOTA 任务并调度 FUOTA 会话。要调试 FUOTA 任务，请执行以下步骤：

1. 通过激活无线设备和多播组的帧信息和组播帧信息设置，创建网络分析器配置。
2. 使用 FUOTA 任务中多播组的多播组标识符，将该多播组添加到网络分析器配置中。通过启用组播帧信息，您可以调试在 FUOTA 任务进行时发送给该组的固件数据消息和 FUOTA 状态查询消息。
3. 使用多播组中设备的无线设备标识符，将这些设备添加到网络分析器配置中。通过激活帧信息，您可以监控在 FUOTA 任务进行时直接与设备交换的上行链路和下行链路消息。
4. 激活跟踪消息，以开始接收网络分析器配置中的设备和多播组的跟踪消息。

然后，如 调试仅包含设备的 FUOTA 任务 中所述，您可以使用跟踪消息表的 `applicationCommandType` 列以及 JSON 日志消息中的详细信息来查看跟踪消息并对其进行调试。

调试试图加入多播组的设备

您可以使用网络分析器来调试试图加入多播组的设备。有关向多播组添加设备的信息，请参阅创建多播组并将设备添加到组。要调试多播组，请执行以下步骤：

1. 通过激活无线设备的帧信息来创建网络分析器配置。
2. 使用要监控的设备的无线设备标识符，将这些设备添加到网络分析器配置中。
3. 激活跟踪消息，以开始接收网络分析器配置中的设备的跟踪消息。
4. 为多播组中的设备激活跟踪消息后，开始将设备与多播组相关联。

调试多播组会话

您可以使用网络分析器调试多播组会话。有关更多信息，请参阅调度向多播组中的设备发送下行链路消息。要调试多播组会话，请执行以下步骤：

1. 通过激活多播组的组播帧信息，创建网络分析器配置。
2. 使用要监控的多播组的多播组标识符，将此多播组添加到网络分析器配置中。
3. 在组播会话开始之前，激活跟踪消息以开始接收多播组会话的跟踪消息。
4. 启动多播组会话，并通过查看跟踪消息表中显示的消息和 JSON 日志消息来监控状态。

在跟踪消息表中，`MulticastAddr` 将显示在 `DevAddr` 列中。在 JSON 日志消息中，您可以查看详细信息，例如 WirelessMetadata > ApplicationInfo 下的 `MulticastGroupId`。

适用于 LoRaWAN 的 AWS IoT Core 和接口 VPC 端点（AWS PrivateLink）

您可以通过虚拟私有云（VPC）中的 接口 VPC 端点（AWS PrivateLink） 直接连接到 适用于 LoRaWAN 的 AWS IoT Core，而不是通过公共互联网进行连接。当您使用 VPC 接口端点时，VPC 与适用于 LoRaWAN 的 AWS IoT Core 之间的通信完全在 AWS 网络内安全进行。

适用于 LoRaWAN 的 AWS IoT Core 支持 Amazon Virtual Private Cloud 接口端点，这些端点由 AWS PrivateLink 提供支持。每个 VPC 端点都由您的 VPC 子网中一个或多个使用私有 IP 地址的弹性网络接口代表。有关更多信息，请参阅《Amazon VPC 用户指南》中的接口 VPC 端点 (AWS PrivateLink)。

有关 VPC 和端点的更多信息，请参阅什么是 Amazon VPC。

有关 AWS PrivateLink 的更多信息，请参阅 AWS PrivateLink 和 VPC 端点。

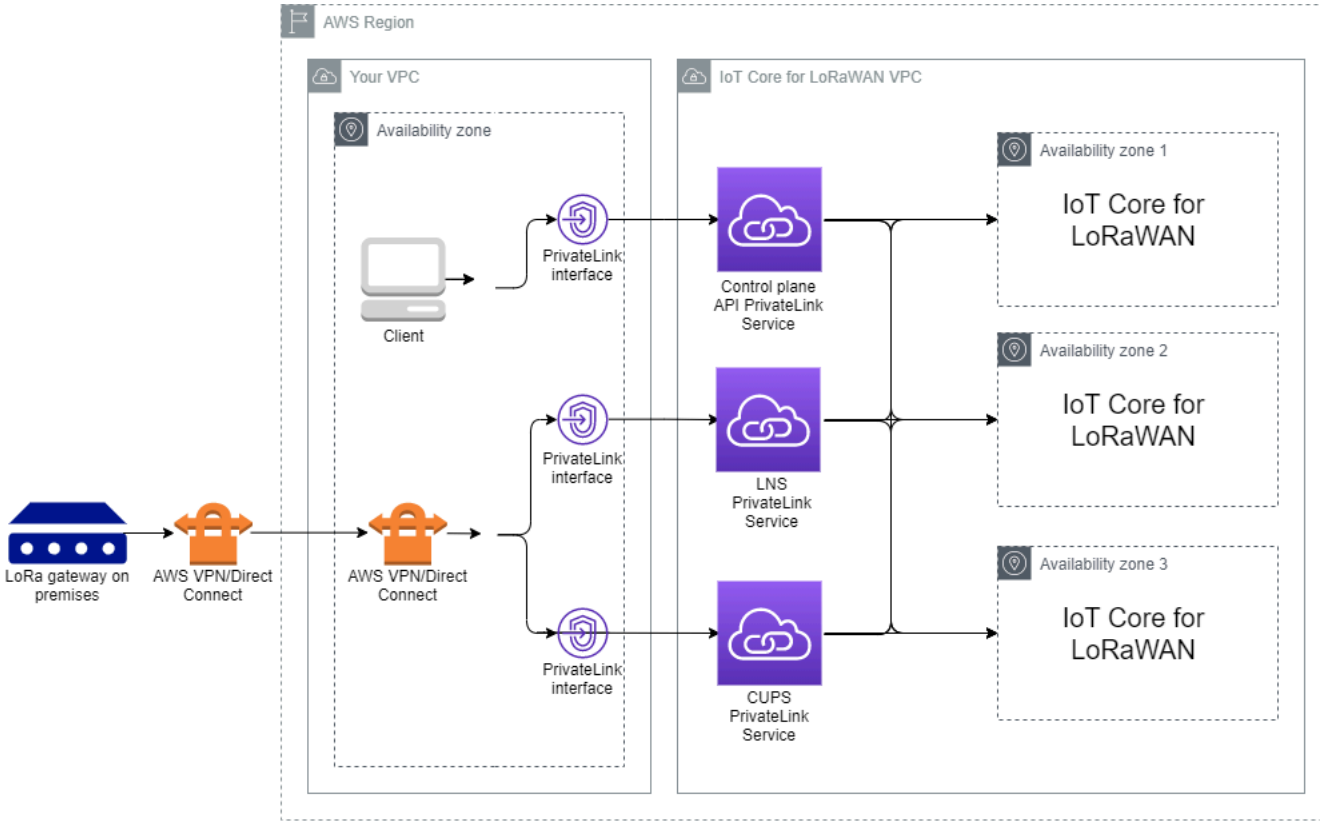
有关 AWS IoT Wireless VPC 端点的考虑事项

请务必先查看《Amazon VPC 用户指南》中的接口端点属性和限制，然后再为 AWS IoT Wireless 设置接口 VPC 端点。

AWS IoT Wireless 支持从 VPC 调用它的所有 API 操作。AWS IoT Wireless 不支持 VPC 端点策略。默认情况下，允许通过端点对 AWS IoT Wireless 进行完全访问。有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 VPC 端点控制对服务的访问权限。

适用于 LoRaWAN 的 AWS IoT Core 私有链接架构

下图显示了 适用于 LoRaWAN 的 AWS IoT Core 私有链路架构 该架构使用 Transit Gateway 和 Route 53 Resolver 在您的 VPC、适用于 LoRaWAN 的 AWS IoT Core VPC 以及本地部署环境之间共享 AWS PrivateLink 接口端点。在设置与 VPC 接口端点的连接时，您将找到更详细的架构图。



适用于 LoRaWAN 的 AWS IoT Core 端点

适用于 LoRaWAN 的 AWS IoT Core 有三个公有端点。每个公有端点都有相应的 VPC 接口端点。公有端点可分为控制面板和数据面板端点。有关这些端点的信息，请参阅 适用于 LoRaWAN 的 AWS IoT Core API 端点。

- 控制面板 API 端点

  您可以使用控制面板 API 端点与 AWS IoT Wireless API 交互。这些端点可以使用 AWS PrivateLink 从托管在 Amazon VPC 中的客户端进行访问。

- 数据面板 API 端点

  数据面板 API 端点是 LoRaWAN Network Server（LNS）和 Configuration and Update Server（CUPS）端点，您可以使用它们与 适用于 LoRaWAN 的 AWS IoT Core LNS 和 CUPS 端点进行交互。这些端点可以使用 AWS VPN 或 AWS Direct Connect 从本地部署的 LoRa 网关访问。您可以在将网关登记到 适用于 LoRaWAN 的 AWS IoT Core 时获取这些端点。有关更多信息，请参阅将网关添加到 适用于 LoRaWAN 的 AWS IoT Core。

主题

- 登记 适用于 LoRaWAN 的 AWS IoT Core 控制面板 API 端点
- 登记 适用于 LoRaWAN 的 AWS IoT Core 数据面板 API 端点

登记 适用于 LoRaWAN 的 AWS IoT Core 控制面板 API 端点

您可以使用 适用于 LoRaWAN 的 AWS IoT Core 控制面板 API 端点与 AWS IoT Wireless API 交互。例如，您可以使用此端点运行 SendDataToWirelessDevice API 以将数据从 AWS IoT 发送到 LoRaWAN 设备。有关更多信息，请参阅适用于 LoRaWAN 的 AWS IoT Core 控制面板 API 端点。

您可以使用 Amazon VPC 中托管的客户端访问由 AWS PrivateLink 提供支持的控制面板端点。您可以通过虚拟私有云（VPC）中的接口端点直接连接到 AWS IoT Wireless API，而不是通过公共互联网进行连接。

登记控制面板端点：

- 创建您的 Amazon VPC 和子网
- 将 Amazon EC2 实例启动到您的子网中
- 创建 Amazon VPC 接口端点

- 测试与接口端点的连接

创建您的 Amazon VPC 和子网

您必须先创建 VPC 和子网，然后才能连接到接口端点。然后，您将在子网中启动 EC2 实例，您可以使用该实例连接到接口端点。

要创建您的 VPC：

1. 导航 Amazon VPC 控制台的 VPC 页面上，然后选择 Create VPC（创建 VPC）。
2. 在 Create VPC（创建 VPC）页面：
   - 为 VPC Name tag - optional（VPC 名称标签 — 可选）输入名称（例如，**VPC-A**）。
   - 在 IPv4 CIDR 块中输入 VPC 的 IPv4 地址范围（例如 **10.100.0.0/16**）。
3. 保留其他字段的默认值，然后选择创建 VPC。

要创建子网：

1. 导航 Amazon VPC 控制台的 Subnets（子网）页面上，然后选择 Create subnet（创建子网）。
2. 在 Create subnet（创建子网）页面中：
   - 对于 VPC ID，请选择之前创建的 VPC（例如，VPC-A）。
   - 在 Subnet name（子网名称）中输入名称（例如，**Private subnet**)。
   - 为您的子网选择可用区。
   - 在 IPv4 CIDR 块中输入子网的 IP 地址数据块（例如，**10.100.0.0/24**)。
3. 要创建子网并将其添加到 VPC 中，请选择 Create subnet（创建子网）。

有关更多信息，请参阅使用 VPC 和子网。

将 Amazon EC2 实例启动到您的子网中

要启动 EC2 实例：

1. 导航到 Amazon EC2 控制台并选择 Launch Instance（启动实例）。
2. 对于 AMI，请选择 Amazon Linux 2 AMI (HVM)，SSD Volume Type，然后选择 t2 micro 实例类型。要配置实例详情，请选择 Next（下一步）。
3. 在 Configure Instance Details（配置实例详细信息）页面：

- 对于 Network（网络），请选择您之前创建的 VPC（例如：`VPC-A`）。
- 对于 Subnet（子网），请选择您之前创建的子网（例如，**`Private subnet`**）。
- 对于 IAM role（IAM 角色），请选择角色 AWSIoTWirelessFullAccess 以授予 适用于 LoRaWAN 的 AWS IoT Core 完全访问策略。有关更多信息，请参阅 `AWSIoTWirelessFullAccess 策略摘要`。
- 对于假定私有 IP，请使用 IP 地址，例如 10.100.0.42。

4. 选择 Next: Add Storage（下一步：添加存储），然后选择 Next: Add Tags（下一步：添加标签）。您可以选择添加要与 EC2 实例关联的任何标签。选择 Next: Configure Security Group。
5. 在 Configure Security Group（配置安全组）页面上，将安全组配置为允许执行以下操作：
   - 对源 `10.200.0.0/16` 打开 All TCP。
   - 对源 `10.200.0.0/16` 打开 All ICMP - IPV4。
6. 要查看实例详细信息并启动 EC2 实例，请选择 Review and Launch（审核和启动）。

有关更多信息，请参阅 Amazon EC2 Linux 实例入门。

创建 Amazon VPC 接口端点

您可以为您的 VPC 创建 VPC 端点，然后可以通过 EC2 API 访问该端点。要创建端点：

1. 导航到 VPC 端点控制台并选择 Create Endpoint（创建端点）。
2. 在 Create Endpoint（创建端点）页面上，指定以下信息。
   - 为 Service category（服务类别）选择AWS 服务。
   - 对于 Service Name（服务名称），通过输入关键字 **`iotwireless`** 进行搜索。在显示的 `iotwireless` 服务列表中，请为您的区域选择控制面板 API 端点。端点的格式为 `com.amazonaws.region.iotwireless.api`。
   - 对于 VPC 和 Subnets（子网），选择要在其中创建端点的 VPC 和要在其中创建端点网络的可用区 (AZ)。

     > **Note**
     >
     > 并非所有可用区都支持 `iotwireless` 服务。

   - 对于 Enable DNS Name（启用 DNS 名称），请选择 Enable for this endpoint（为此端点启用）。

选择此选项将自动解析 DNS 并在 Amazon Route 53 Public Data Plane 创建一个路由，以便稍后用于测试连接的 API 能够通过私密链路端点。

- 对于 Security group（安全组），选择要与端点网络接口关联的安全组。
- 您可以选择添加或删除标签。标签是用于与端点关联的名称-值对。

3. 要创建 VPC 端点，请选择 Create endpoint（创建端点）。

测试与接口端点的连接

您可以使用 SSH 访问您的 Amazon EC2 实例，然后使用 AWS CLI 连接到私有链路接口端点。

在连接到接口端点之前，请根据在 Linux 上安装、更新和卸载 AWS CLI 版本 2 中所描述的方法下载最新的 AWS CLI 版本。

以下示例介绍如何使用 CLI 测试到接口端点的连接。

```
aws iotwireless create-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com  \
    --name='test-privatelink'
```

以下是运行命令的示例。

```
Response:
{
 "Arn": "arn:aws:iotwireless:region:acct_number:ServiceProfile/1a2345ba-4c5d-67b0-ab67-e0c8342f2857",
 "Id": "1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
}
```

同样，您可以运行以下命令来获取服务配置文件信息或列出所有服务配置文件。

```
aws iotwireless get-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
    --id="1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
```

以下是 list-device-profiles 命令的示例。

```
aws iotwireless list-device-profiles \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
```

登记 适用于 LoRaWAN 的 AWS IoT Core 数据面板 API 端点

适用于 LoRaWAN 的 AWS IoT Core 数据面板端点由以下端点组成。将网关添加到 适用于 LoRaWAN 的 AWS IoT Core 时，您将获得这些端点。有关更多信息，请参阅将网关添加到 适用于 LoRaWAN 的 AWS IoT Core。

- LoRaWAN 网络服务器 (LNS) 端点

  LNS 端点的格式为 *account-specific-prefix*.lns.lorawan.*region*.amazonaws.com。您可以使用此端点建立用于交换 LoRa 上行链路和下行链路消息的连接。

- Configuration and Update Server (CUPS) 端点

  CUPS 端点的格式为 *account-specific-prefix*.cups.lorawan.*region*.amazonaws.com。您可以将此端点用于网关的凭证管理、远程配置和固件更新。

有关更多信息，请参阅使用 CUPS 和 LNS 协议。

要查找适用于您的 AWS 账户 和区域的数据面板 API 端点，请使用在此显示的 get-service-endpoint CLI 命令，或是 GetServiceEndpoint REST API。有关更多信息，请参阅适用于 LoRaWAN 的 AWS IoT Core 数据面板 API 端点。

您可以在本地部署连接 LoRaWAN 网关，以便与 适用于 LoRaWAN 的 AWS IoT Core 端点通信。要建立此连接，请首先使用 VPN 连接将本地部署网关连接到您 VPC 中的AWS 账户。然后，您可以与 适用于 LoRaWAN 的 AWS IoT Core VPC 中的数据面板接口端点通信，该 VPC 由私有链路提供支持。

以下是如何登记这些端点的演示。

- 创建 VPC 接口端点和私有托管区域
- 使用 VPN 以将 LoRa 网关连接到 AWS 账户。

创建 VPC 接口端点和私有托管区域

适用于 LoRaWAN 的 AWS IoT Core 有两个数据面板端点： Configuration and Update Server (CUPS) 端点和 LoRaWAN Network Server (LNS) 端点。建立到两个端点的私有链路连接的设置过程是相同的，因此我们可以使用 LNS 端点进行说明。

对于数据面板端点，LoRa 网关首先连接到您在 Amazon VPC 中的 AWS 账户，然后它会连接到 适用于 LoRaWAN 的 AWS IoT Core VPC 中的 VPC 端点。

连接到端点时，DNS 名称可以在一个 VPC 内解析，但无法跨多个 VPC 解析。要在创建端点时禁用私有 DNS，请禁用 Enable DNS name（启用 DNS 名称）设置。您可以使用私有托管区域提供有关您希望 Route 53 响应 VPC 的 DNS 查询的信息。要与本地部署环境共享您的 VPC，您可以使用 Route 53 Resolver 来支持混合 DNS。

要完成本流程，请执行以下步骤。

- 创建 Amazon VPC 和子网
- 创建 Amazon VPC 接口端点
- 配置私有托管区域
- 配置 Route 53 入站解析程序
- 后续步骤

创建 Amazon VPC 和子网

您可以重复使用您在登记控制面板端点时创建的 Amazon VPC 和子网。有关信息，请参阅 创建您的 Amazon VPC 和子网。

创建 Amazon VPC 接口端点

您可以为 VPC 创建 VPC 端点，这与为控制面板端点创建 VPC 端点的方式类似。

1. 导航到 VPC 端点控制台并选择 Create Endpoint（创建端点）。
2. 在 Create Endpoint（创建端点）页面上，指定以下信息。
   - 为 Service category（服务类别）选择AWS 服务。
   - 对于 Service Name（服务名称），通过输入关键字 **lns** 进行搜索。在显示的 `lns` 服务列表中，请选择您所在区域的 LNS 数据面板 API 端点。端点的格式为 `com.amazonaws.region.lorawan.lns`。

     > **Note**
     >
     > 如果您正在为 CUPS 端点执行此流程，请搜索 `cups`。端点的格式为 `com.amazonaws.region.lorawan.cups`。

   - 对于 VPC 和 Subnets（子网），选择要在其中创建端点的 VPC 和要在其中创建端点网络的可用区 (AZ)。

> **Note**
>
> 并非所有可用区都支持 `iotwireless` 服务。

- 对于 Enable DNS name（启用 DNS 名称），请确保未选择 Enable for this endpoint（为此端点启用）。

  如果不选择此选项，您可以禁用 VPC 端点的私有 DNS，并改为使用私有托管区域。
- 对于 Security group（安全组），选择要与端点网络接口关联的安全组。
- 您可以选择添加或删除标签。标签是用于与端点关联的名称-值对。

3. 要创建 VPC 端点，请选择 Create endpoint（创建端点）。

配置私有托管区域

创建私有链路端点后，在端点的 Details（详细信息）选项卡下，您将看到 DNS 名称列表。您可以使用这些 DNS 名称来配置私有托管区域。DNS 名称将采用 `vpce-xxxx.lns.lorawan.region.vpce.amazonaws.com` 格式。

创建私有托管区域

要创建私有托管区域：

1. 导航到 Route 53 托管区域控制台并选择 Create hosted zone（创建托管区域）。
2. 在 Create hosted zone（创建托管区域）页面上，指定以下信息。
   - 对于 Domain name（域名），输入 LNS 端点的完整服务名称，**`lns.lorawan.region.amazonaws.com`**。

     > **Note**
     >
     > 如果您正在为 CUPS 端点执行此流程，请输入 **`cups.lorawan.region.amazonaws.com`**。

   - 对于 Type（类型），选择 Private hosted zone（私有托管区域）。
   - 或者，您可以添加或删除要与托管区域关联的标签。
3. 要创建您的私有托管区域，请选择 Create hosted zone（创建托管区域）。

有关更多信息，请参阅创建私有托管区域。

创建私有托管区域后，您可以创建一个记录，告诉 DNS 如何将流量路由到该域。

创建记录

创建私有托管区域后，您可以创建一个记录，告诉 DNS 如何将流量路由到该域。要创建记录：

1. 在显示的托管区域列表中，选择您之前创建的私有托管区域，然后选择 Create record（创建记录）。
2. 使用向导提供的方法创建记录。如果控制台为您提供了 Quick create（快速创建）的方法，选择 Switch to wizard（切换到向导）。
3. 在 Routing policy（路由策略）中选择 Simple Routing（简单路由），然后选择 Next（下一步）。
4. 在 Configure records（配置记录）下，选择 Define simple record（定义简单记录）。
5. 在 Define simple record（定义简单记录）页面:
   - 对于 Record name（记录名称），输入您的AWS 账户账号别名。您可以在登记网关时获取此值，或者通过使用 `GetServiceEndpoint` REST API 获取此值。
   - 对于 Record type（记录类型），请将该值保持为 `A - Routes traffic to an IPv4 address and some AWS resources`。
   - 对于 Value/Route traffic to（值/流量路由至），选择 Alias to VPC endpoint（向 VPC 添加别名）。然后选择您的 Region（区域），接着从显示的端点列表中选择您之前创建的端点，如 创建 Amazon VPC 接口端点 中所述。
6. 选择 Define simple record（定义简单记录）以创建您的记录。

配置 Route 53 入站解析程序

若要将 VPC 端点共享到本地部署环境，可以使用 Route 53 Resolver 来支持混合 DNS。入站解析程序将允许您将流量从本地部署网络路由到数据面板端点，而无需通过公共互联网。要为服务返回私有 IP 地址值，请在 VPC 端点所在的 VPC 中创建 Route 53 Resolver。

创建入站解析程序时，您只需指定您之前在可用区 (AZ) 中创建的 VPC 以及子网即可。Route 53 Resolver 使用此信息自动分配 IP 地址，将流量路由到每个子网。

要创建入站解析程序：

1. 导航到 Route 53 Inbound endpoint（Route 53 入站端点）控制台并选择 Create inbound endpoint（创建入站端点）。

> **Note**
>
> 请确保您使用的是与您在创建端点和私有托管区域时相同的AWS 区域。

2. 在 Create inbound endpoint（创建入站端点）页面，请指定以下信息。
   - 为 Endpoint name（端点名称）输入名称（例如，**`VPC_A_Test`**）。
   - 对于 VPC in the region（该区域中的 VPC），选择您在创建 VPC 端点时使用的相同 VPC。
   - 配置 Security group for this endpoint（适用于此端点的安全组）以允许来自本地部署网络的传入流量。
   - 对于 IP 地址，选择 Use an IP adress that is selected automatically（使用自动选择的 IP 地址）。
3. 选择 Submit（提交）以创建您的入站解析程序。

在这个例子中，让我们假设 IP 地址 `10.100.0.145` 和 `10.100.192.10` 被分配给了路由流量的入站 Route 53 Resolver。

后续步骤

您已创建私有托管区域和入站解析程序，以路由 DNS 条目的流量。现在，您可以使用 Site-to-Site VPN 或 Client VPN 端点。有关更多信息，请参阅使用 VPN 以将 LoRa 网关连接到 AWS 账户。。

使用 VPN 以将 LoRa 网关连接到 AWS 账户。

要将本地部署网关连接到您的AWS 账户，可以使用 Site-to-Site VPN 连接或 Client VPN 端点。

在连接本地部署网关之前，您必须已创建 VPC 端点，并配置私有托管区和入站解析程序，以避免来自网关的流量通过公共网络。有关更多信息，请参阅创建 VPC 接口端点和私有托管区域。

Site-to-Site VPN 端点

如果您没有网关硬件或希望使用其他AWS 账户测试 VPN 连接，则可以使用 Site-to-Site VPN 连接。您可以使用 Site-to-Site VPN 连接来自同一个 AWS 账户的 VPC 端点；或如果您打算在不同 AWS 区域中使用，则可以选择来自其他 AWS 账户的 VPC 端点。

> **Note**
>
> 如果您拥有网关硬件并希望建立 VPN 连接，我们建议您改用 Client VPN。有关说明，请参阅Client VPN 端点。

要设置 Site-to-Site VPN：

1. 在要从中设置连接的站点中创建另一个 VPC。对于 VPC-A，您可以重复使用之前创建的 VPC。要创建另一个 VPC（例如，VPC-B），请使用与您之前创建的 VPC 的 CIDR 块不重叠的 CIDR 块。

   有关设置 VPC 的信息，请按照AWS设置 Site-to-Site VPN 连接中所述操作执行。

   > **Note**
   >
   > 文档中描述的 Site-to-Site VPN VPN 方法使用 OpenSWAN 进行 VPN 连接，该连接仅支持一个 VPN 隧道。如果您为 VPN 使用了不同的商业软件，则可在站点之间设置两个通道。

2. 设置 VPN 连接后，请添加您的AWS 账户中的入站解析程序 IP 地址，以更新 `/etc/resolv.conf` 文件。您将为名称服务器使用此 IP 地址。有关如何获取此 IP 地址的信息，请参阅 配置 Route 53 入站解析程序。在这个示例中，我们可以使用您在创建 Route 53 Resolver 时分配的 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. 现在，我们可以测试 VPN 连接是否使用了 AWS PrivateLink 端点，而无需使用 `nslookup` 命令经由公共互联网进行测试。以下是运行命令的示例。

   ```
   nslookup account-specific-prefix.lns.lorawan.region.amazonaws.com
   ```

   下面显示了运行命令的示例输出，该输出显示了一个私有 IP 地址，表示与 AWS PrivateLink LNS 端点的连接已建立。

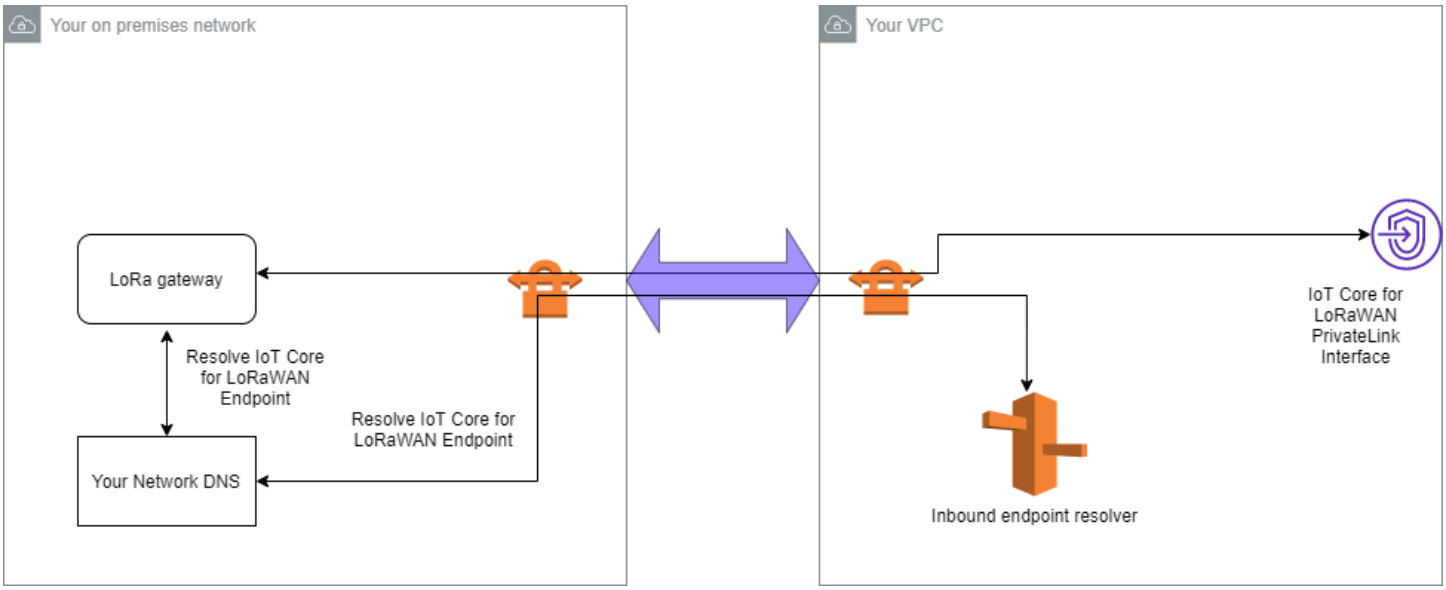
   ```
   Server: 10.100.0.145
   Address: 10.100.0.145
   ```

```
Non-authoritative answer:
Name: https://xxxxx.lns.lorawan.region.amazonaws.com
Address: 10.100.0.204
```

有关使用 Site -to-Site VPN 连接的信息，请参阅 Site-to-Site VPN 的工作原理。

Client VPN 端点

AWS Client VPN 是一种基于客户端的托管 VPN 服务，让您能够安全地访问 AWS 资源和本地网络中的资源。下面显示了 Client VPN 服务的架构。



要建立到 Client VPN 端点的 VPN 连接：

1. 按照 AWS Client VPN 入门中描述的说明创建 Client VPN 端点。
2. 使用该路由器的访问 URL（例如 `192.168.1.1`），登录到本地网络（例如，Wi-Fi 路由器），然后查找根名称和密码。
3. 按照网关文档中的说明设置 LoRaWAN 网关，然后将您的网关添加到 适用于 LoRaWAN 的 AWS IoT Core。有关如何添加网关的信息，请参阅 将您的网关登记到 适用于 LoRaWAN 的 AWS IoT Core。
4. 检查您的网关固件是否为最新版本。如果固件过期，您可以按照本地部署网络中提供的说明更新网关的固件。有关更多信息，请参阅使用 适用于 LoRaWAN 的 AWS IoT Core 的 CUPS 服务更新网关固件。
5. 检查 OpenVPN 是否已启用。如果已启用，请跳到下一步以配置本地部署网络中的 OpenVPN 客户端。如果尚未启用，请遵循为 OpenWrt 安装 OpenVPN 的指南。

> **Note**
>
> 在此示例中，我们使用 OpenVPN。您可以使用其他 VPN 客户端，例如 AWS VPN 或者 AWS Direct Connect 以设置 Client VPN 连接。

6. 根据客户端配置中的信息配置 OpenVPN 客户端，并了解您可以如何使用使用 LuCi 的 OpenVPN 客户端。
7. 将入站解析程序的 IP 地址添加到AWS 账户 (`10.100.0.145`)，从而 SSH 连接到您的本地部署网络，并更新`/etc/resolv.conf`文件。
8. 对于要使用AWS PrivateLink连接到端点的网关流量，请将网关的第一个 DNS 条目替换为入站解析程序的 IP 地址。

有关使用 Site-to-Site VPN 连接的信息，请参阅 Client VPN 入门。

连接到 LNS 和 CUPS VPC 端点

下面显示了如何测试与 LNS 和 CUPS VPC 端点的连接。

测试 CUPS 端点

要测试您的 LoRa 网关与 CUPS 端点的 AWS PrivateLink 连接，请运行以下命令：

```
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
```

测试 LNS 端点

要测试您的 LNS 端点，请首先预置一个可与您的无线网关配合使用的 LoRaWAN 设备。然后，您可以添加您的设备并执行加入流程，随后您便可以开始发送上行链路消息了。

适用于 Amazon Sidewalk 的 AWS IoT Core

适用于 Amazon Sidewalk 的 AWS IoT Core 提供云服务，您可以使用这些服务将 Sidewalk 终端设备连接到 AWS Cloud 并使用其他 AWS 服务。

Amazon Sidewalk 是一个安全的共享网络，使您社区中的设备能够连接并保持连接。Amazon Sidewalk 在 Sidewalk 终端设备和 Sidewalk 网关之间以及 Sidewalk 网关和 Sidewalk 云之间传输数据。

访问适用于 Amazon Sidewalk 的 AWS IoT Core

您可以使用控制台或 AWS IoT Wireless API 操作将 Sidewalk 终端设备登记到 AWS IoT。登记设备后，其消息将发送至 AWS IoT Core。然后，您可以开始在 AWS 云上开发您的业务应用程序，它们将使用来自您的 Amazon Sidewalk 终端设备中的数据。

使用控制台

要登记 Sidewalk 终端设备，请登录 AWS Management Console并导航到 AWS IoT 控制台上的设备页面。登记设备后，您可以在 IoT 控制台的此页面上查看和管理它们。

使用 API 或 CLI

您可以使用 AWS IoT Wireless API 操作登记 Sidewalk 和 LoRaWAN 设备。用于构建 AWS IoT Core 的 AWS IoT Wireless API 由 AWS SDK 提供支持。有关更多信息，请参阅 AWS SDK 和工具包。

您可以使用 AWS CLI 运行用于登记和管理 Sidewalk 终端设备的命令。有关更多信息，请参阅 AWS IoT Wireless CLI 参考。

适用于 Amazon Sidewalk 的 AWS IoT Core 区域和端点

Amazon Sidewalk 仅在 `us-east-1` AWS 区域可用。适用于 Amazon Sidewalk 的 AWS IoT Core 为该区域中的控制面板和数据面板 API 端点提供支持。数据面板 API 端点特定于您的 AWS 账户。有关更多信息，请参阅《AWS General Reference》中的 AWS IoT Wireless Service endpoints。

对于在设备与 AWS Cloud之间传输的设备数据以及 AWS IoT Wireless API 操作的最大 TPS，适用于 Amazon Sidewalk 的 AWS IoT Core 具有相应的限额。有关更多信息，请参阅《AWS General Reference》中的 AWS IoT Wireless quotas。

适用于 Amazon Sidewalk 的 AWS IoT Core 的定价

注册 AWS 后，您可以通过 AWS 免费套餐开始免费使用适用于 Amazon Sidewalk 的 AWS IoT Core。

有关一般产品概述和定价的更多信息，请参阅 AWS IoT Core定价。

什么是适用于 Amazon Sidewalk 的 AWS IoT Core？

借助适用于 Amazon Sidewalk 的 AWS IoT Core，您可以将 Amazon Sidewalk 终端设备登记到 AWS IoT 并对其进行管理和监控。它还负责管理向其他 AWS 服务发送设备数据的目标。

适用于 Amazon Sidewalk 的 AWS IoT Core 的功能

借助适用于 Amazon Sidewalk 的 AWS IoT Core，您可以：

- 使用 AWS IoT 控制台、适用于 Amazon Sidewalk 的 AWS IoT Core API 操作或 AWS CLI 命令将 Sidewalk 终端设备登记到 AWS IoT。
- 利用 AWS Cloud提供的功能。
- 创建一个使用 AWS IoT 规则来处理传入的有效负载消息并与其他 AWS 服务进行交互的目标。
- 启用事件通知以接收有关此类事件的消息：例如，当已预置或注册了 Sidewalk 终端设备时，或者下行链路消息是否已成功传送到您的设备。
- 实时记录和监控 Sidewalk 终端设备，获取有用的见解，并识别和排除错误。
- 将 Sidewalk 终端设备与 AWS IoT 事物关联，这可以帮助您在云端存储设备的表示形式。AWS IoT 中的事物使您能够更轻松地搜索和管理特征以及访问其他 AWS IoT Core 功能。

以下主题将帮助您了解 Amazon Sidewalk 和适用于 Amazon Sidewalk 的 AWS IoT Core。

主题

什么是 Amazon Sidewalk？

Amazon Sidewalk 是一个安全的社区网络，它使用 Amazon Sidewalk 桥接器（例如兼容的 Amazon Echo 和 Ring 设备）为 IoT 设备提供云连接。Amazon Sidewalk 使用蓝牙 LE 进行短距离通信，使用频率为 900MHz 的 LoRa 和 FSK 无线电协议覆盖更长的距离，从而在家内外实现低带宽和远程连接。

启用 Amazon Sidewalk 后，该网络可以在社区中支持其他 Sidewalk 终端设备，并可用于感知环境等应用。Amazon Sidewalk 可帮助设备进行连接并保持连接。

Amazon Sidewalk 的功能

以下是 Amazon EKS 的功能。

- Amazon Sidewalk 使用包含 Ring 和精选 Echo 设备的 Sidewalk 网关创建低带宽网络。使用网关，您可以共享部分互联网带宽，然后使用这部分带宽来将终端设备连接到网络。
- Amazon Sidewalk 提供了一种具有多层加密和安全性的安全联网机制。
- Amazon Sidewalk 提供了一种简单的机制来启用或禁用参与 Sidewalk。

Amazon Sidewalk 概念

以下是 Amazon Sidewalk 的一些关键概念。

Sidewalk 网关

Sidewalk 网关或 Amazon Sidewalk 桥接器可在 Sidewalk 终端设备和云端之间路由数据。网关是 Amazon 设备（例如 Echo 设备或 Ring Floodlight Cam），它们支持 SubG-CSS（异步、LDR）、SubG-FSK（同步、HDR）或蓝牙 LE 进行 Sidewalk 通信。Sidewalk 网关与 Sidewalk 社区共享您的部分互联网带宽，以提供与一组支持 Sidewalk 的设备的连接。

Sidewalk 终端设备

Sidewalk 终端设备通过连接到 Sidewalk 网关在 Amazon Sidewalk 上漫游。终端设备是低带宽、低功耗的智能产品，例如支持 Sidewalk 的灯或门锁。

> **Note**
>
> 某些 Sidewalk 网关也可以充当终端设备。

Sidewalk 网络服务器

由 Amazon 运营的 Sidewalk 网络服务器验证传入的数据包，并将上行链路和下行链路消息路由到所需的目标，同时使 Sidewalk 网络保持时间同步。

了解有关 Amazon Sidewalk 的更多信息

有关 Amazon Sidewalk 的更多信息，请参阅以下网页：

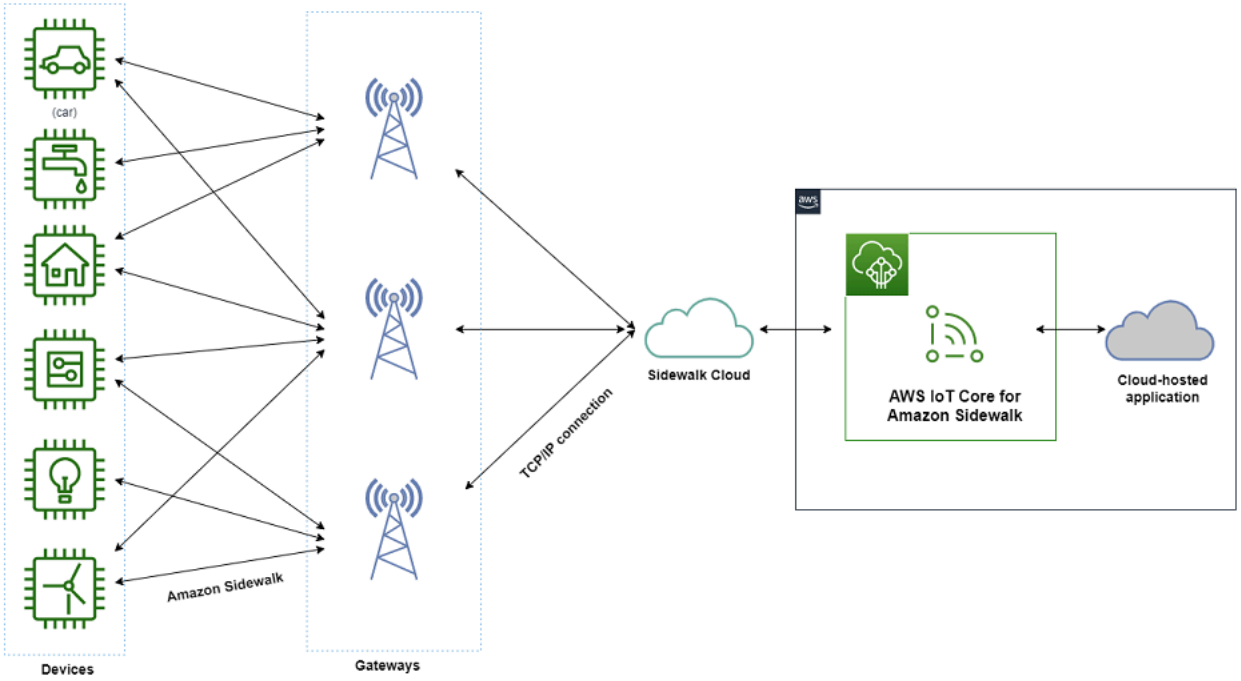
- Amazon Sidewalk
- Amazon Sidewalk 文档
- 适用于 Amazon Sidewalk 的 AWS IoT Core

适用于 Amazon Sidewalk 的 AWS IoT Core 的工作原理

借助适用于 Amazon Sidewalk 的 AWS IoT Core，您可以将 Amazon Sidewalk 终端设备登记到 AWS IoT 并对其进行管理和监控。它还负责管理向其他 AWS 服务发送设备数据的目标。

适用于 Amazon Sidewalk 的 AWS IoT Core 提供云服务，您可以使用这些服务将 Sidewalk 终端设备连接到 AWS Cloud 并使用其他 AWS 服务。您还可以使用适用于 Amazon Sidewalk 的 AWS IoT Core 管理 Sidewalk 设备，并在这些设备上监控和构建应用程序。

Sidewalk 终端设备通过 Sidewalk 网关与 AWS IoT Core 通信。适用于 Amazon Sidewalk 的 AWS IoT Core 负责管理 AWS IoT Core 用来管理 Sidewalk 终端设备和网关以及与它们进行通信所需的服务和设备策略。它还负责管理向其他 AWS 服务发送设备数据的目标。



开始使用适用于 Amazon Sidewalk 的 AWS IoT Core

您可以使用 AWS IoT 控制台、适用于 Amazon Sidewalk 的 AWS IoT Core API 或 AWS CLI 来创建和登记 Sidewalk 终端设备，并将它们连接到 Sidewalk 网络。有关开始使用 Amazon Sidewalk 以及将终端设备登记到 AWS IoT 的信息，请参阅以下主题。

- 开始使用适用于 Amazon Sidewalk 的 AWS IoT Core

  本主题介绍登记 Sidewalk 终端设备的先决条件，说明使用传感器监控应用程序的工作流，并概述如何使用 AWS CLI 命令登记设备。

- 连接到适用于 Amazon Sidewalk 的 AWS IoT Core

  本节介绍登记工作流简介中的不同步骤，并演练如何使用控制台和 API 操作登记终端设备。您还将连接设备并查看在设备与适用于 Amazon Sidewalk 的 AWS IoT Core 之间交换的消息。

- 使用适用于 Amazon Sidewalk 的 AWS IoT Core 批量预置设备

  本节提供了详细的分步教程，介绍如何使用适用于 Amazon Sidewalk 的 AWS IoT Core 批量预置 Sidewalk 终端设备。您将了解如何批量预置工作流，以及如何登记大量 Sidewalk 设备。

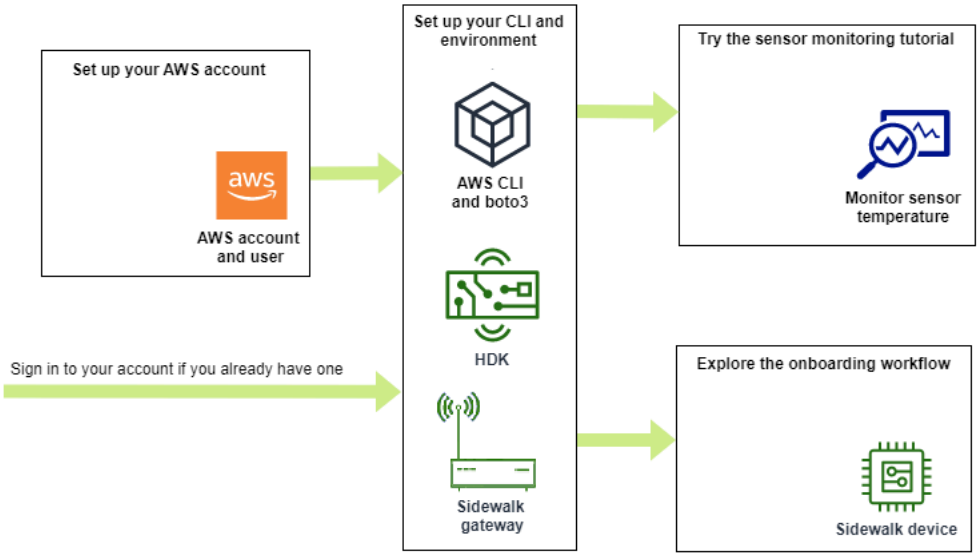
了解有关适用于 Amazon Sidewalk 的 AWS IoT Core 的更多信息

有关适用于 Amazon Sidewalk 的 AWS IoT Core 的更多信息，请参阅以下网页：

- Amazon Sidewalk
- Amazon Sidewalk 文档
- 适用于 Amazon Sidewalk 的 AWS IoT Core

开始使用适用于 Amazon Sidewalk 的 AWS IoT Core

本节向您展示如何开始将 Sidewalk 终端设备连接到适用于 Amazon Sidewalk 的 AWS IoT Core，并且解释了如何将终端设备连接到 Amazon Sidewalk 并在它们之间传递消息。您还将大致了解 Sidewalk 应用程序示例，以及如何使用适用于 Amazon Sidewalk 的 AWS IoT Core 执行传感器监控。该应用程序示例为您提供了一个控制面板，用于查看和监控传感器温度的变化。



以下主题将帮助您开始使用适用于 Amazon Sidewalk 的 AWS IoT Core。

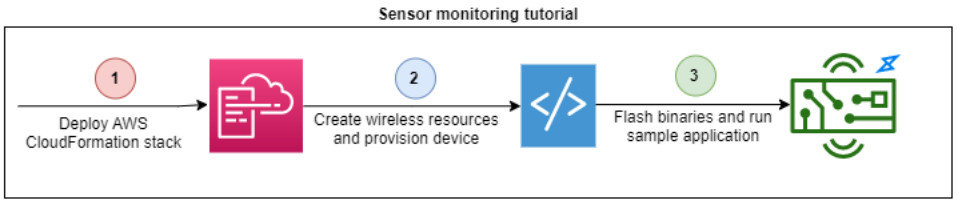
主题

- 试用传感器监控教程
- 登记 Sidewalk 设备简介

试用传感器监控教程

本节概述了 GitHub 上的 Amazon Sidewalk 示例应用程序，该应用程序向您展示如何监控传感器的温度。在本教程中，您使用脚本以编程方式创建所需的无线资源，预置终端设备并刷写二进制文件，然后将终端设备连接到应用程序。使用 AWS CLI 和 Python 命令的脚本会创建 AWS CloudFormation 堆栈和无线资源，然后刷写二进制文件，并将应用程序部署到您的硬件开发工具包（HDK）上。

下图显示了您运行示例应用程序并将 Sidewalk 终端设备连接到该应用程序时所涉及的步骤。有关包括本教程的先决条件和配置在内的详细说明，请参阅 GitHub 中的自述文件文档。



登记 Sidewalk 设备简介

本节向您展示如何将 Sidewalk 终端设备登记到适用于 Amazon Sidewalk 的 AWS IoT Core。要登记您的设备，请先添加 Sidewalk 设备，接着预置和登记设备，然后将硬件连接到云应用程序。在运行本教程之前，请仔细阅读并完成安装 Python 和 AWS CLI。

以下步骤向您展示了如何登记 Sidewalk 终端设备并将其连接到适用于 Amazon Sidewalk 的 AWS IoT Core。如果要使用 AWS CLI 登记设备，可以参考本节提供的命令示例。有关使用 AWS IoT 控制台登记设备的信息，请参阅连接到适用于 Amazon Sidewalk 的 AWS IoT Core。

> **Important**
>
> 要执行整个登记工作流，您还需要预置和登记终端设备，并连接硬件开发工具包（HDK）。有关更多信息，请参阅《Amazon Sidewalk 文档》中的预置和注册终端设备。

主题

步骤 1：将 Sidewalk 设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core

下面概述了将 Sidewalk 终端设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core 需要执行的步骤。存储您获得的有关您创建的设备配置文件和无线设备的信息。您将使用此信息来预置和注册终端设备。有关这些步骤的更多信息，请参阅将设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core。

1. 创建设备配置文件

   创建包含 Sidewalk 设备的共享配置的设备配置文件。创建配置文件时，请将配置文件的 *name* 指定为字母数字字符串。要创建配置文件，请转到 AWS IoT 控制台的配置文件中心的 Sidewalk 选项卡并选择创建配置文件，或者使用 `CreateDeviceProfile` API 操作或 `create-device-profile` CLI 命令，如本示例所示。

   ```
   // Add your device profile using a name and the sidewalk object.
   aws iotwireless create-device-profile --name sidewalk_profile --sidewalk {}
   ```

2. 创建 Sidewalk 终端设备

使用适用于 Amazon Sidewalk 的 AWS IoT Core 创建 Sidewalk 终端设备。指定目标名称和从上一步获得的设备配置文件的 ID。要添加设备，请转到 AWS IoT 控制台的设备中心的 Sidewalk 选项卡并选择预置设备，或者使用 CreateWirelessDevice API 操作或 create-wireless-device CLI 命令，如本示例所示。

> **Note**
>
> 为目标指定一个对您的 AWS 账户和 AWS 区域具有唯一性的名称。将目标添加到适用于 Amazon Sidewalk 的 AWS IoT Core 中时，您将使用相同的目标名称。

```
// 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. 获取设备配置文件和无线设备信息

以 JSON 格式获取设备配置文件和无线设备信息。JSON 将包含有关设备详细信息、设备证书、私有密钥、DeviceTypeId 和 Sidewalk 制造序列号（SMSN）的信息。

- 如果您使用的是 AWS IoT 控制台，则可以使用设备中心的 Sidewalk 选项卡为 Sidewalk 终端设备下载组合的 JSON 文件。
- 如果您使用的是 API 操作，请将从 API 操作 GetDeviceProfile 和 GetWirelessDevice 获得的响应存储为单独的 JSON 文件，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
```

步骤 2：为 Sidewalk 终端设备创建目标

下面概述了将目标添加到适用于 Amazon Sidewalk 的 AWS IoT Core 需要执行的步骤。使用 AWS Management Console、AWS IoT Wireless API 操作或 AWS CLI，您可以运行以下步骤来创建 AWS IoT 规则和目标。然后，您可以连接到硬件平台，并查看和交换消息。有关本节中用于 AWS CLI 示例的 IAM 角色和 AWS IoT 规则，请参阅为您的目标创建 IAM 角色和 IoT 规则。

1. 创建 IAM 角色

   创建一个 IAM 角色来授予适用于 Amazon Sidewalk 的 AWS IoT Core 向 AWS IoT 规则发送数据的权限。要创建角色，请使用 CreateRole API 操作或 create-role CLI 命令。您可以将角色命名为 *SidewalkRole*。

   ```
   aws iam create-role --role-name lambda-ex \
       --assume-role-policy-document file://lambda-trust-policy.json
   ```

2. 为目标创建规则

   创建一条 AWS IoT 规则，用于处理设备数据并指定要向其发布消息的主题。连接到硬件平台后，您将观察到有关此主题的消息。使用 AWS IoT Core API 操作 CreateTopicRule 或 AWS CLI 命令 create-topic-rule 为目标创建规则。

   ```
   aws iot create-topic-rule --rule-name Sidewalkrule \
       --topic-rule-payload file://myrule.json
   ```

3. 创建目标

   创建一个目标以将 Sidewalk 设备与 IoT 规则相关联，此 IoT 规则对设备进行处理，以便用于其他 AWS 服务。您可以使用 AWS IoT 控制台的目标中心、CreateDestination API 操作或 create-destination CLI 命令添加目标。

   ```
   aws iotwireless create-destination --name SidewalkDestination \
       --expression-type RuleName --expression SidewalkRule \
       --role-arn arn:aws:iam::123456789012:role/SidewalkRole
   ```

步骤 3：预置和注册终端设备

使用 Python 命令，您可以预置和注册您的终端设备。预置脚本使用您获得的设备 JSON 数据生成制造二进制映像，然后将其刷写到硬件板上。然后，您可以注册终端设备以连接到硬件平台。有关更多信息，请参阅《Amazon Sidewalk 文档》中的预置和注册终端设备。

> **Note**
>
> 注册您的 Sidewalk 终端设备时，您的网关必须选择加入 Amazon Sidewalk，并且网关和设备必须位于彼此的范围内。

步骤 4：连接到终端设备并交换消息

注册终端设备后，您可以连接终端设备并开始交换消息和设备数据。

1. 连接 Sidewalk 终端设备

   将 HDK 连接到您的计算机，然后按照供应商文档提供的说明连接到您的 HDK。有关更多信息，请参阅《Amazon Sidewalk 文档》中的预置和注册终端设备。

2. 查看和交换消息

   使用 MQTT 客户端订阅在规则中指定的主题并查看收到的消息。您还可以使用 `SendDataToWirelessDevice` API 操作或 `send-data-to-wireless-device` CLI 命令向设备发送下行链路消息并验证连接状态。

   （可选）您可以启用消息传送状态事件来检查是否已成功接收下行链路消息。

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

连接到适用于 Amazon Sidewalk 的 AWS IoT Core

本节介绍如何登记 Sidewalk 终端设备，然后将设备连接到 Sidewalk 网络。它描述了您在登记教程中执行的步骤，如登记 Sidewalk 设备简介中所述。您将了解如何使用 AWS IoT 控制台和适用于 Amazon Sidewalk 的 AWS IoT Core API 操作来登记设备。您还将了解执行这些操作的 AWS CLI 命令。

先决条件

要将终端设备和目标添加到适用于 Amazon Sidewalk 的 AWS IoT Core，必须设置您的 AWS 账户。要使用 AWS IoT Wireless API 或 AWS CLI 命令执行这些操作，还必须设置 AWS CLI。有关先决条件和设置的更多信息，请参阅安装 Python 和 AWS CLI。

> **Note**
>
> 要执行预置和登记终端设备以及连接到硬件开发工具包（HDK）的整个登记工作流，还必须设置 Sidewalk 网关和 HDK。有关更多信息，请参阅《Amazon Sidewalk 文档》中的设置硬件开发工具包（HDK）和设置 Sidewalk 网关。

描述 Sidewalk 资源

在您开始并创建资源之前，建议您考虑 Sidewalk 终端设备、设备配置文件和目标的命名约定。适用于 Amazon Sidewalk 的 AWS IoT Core 为您创建的资源分配唯一标识符。但是，您可以为它们提供更具描述性的名称、添加描述或添加可选标签来帮助标识和管理它们。

> **Note**
>
> 目标名称在创建目标后就无法更改。使用对于您的 AWS 账户和 AWS 区域具有唯一性的名称。

有关更多信息，请参阅描述您的 AWS IoT Wireless 资源。

主题

- 将设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core
- 为 Sidewalk 终端设备添加目标
- 连接您的 Sidewalk 设备并查看上行链路元数据格式

将设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core

在创建无线设备之前，请先创建设备配置文件。设备配置文件定义了 Sidewalk 设备的设备功能和其他参数。单个设备配置文件可以与多个设备关联。

创建设备配置文件后，当您检索有关该配置文件的信息时，它会返回 DeviceTypeId。在预置终端设备时，您将使用此 ID、设备证书、应用程序服务器公钥和 SMSN。

如何创建和添加设备

1. 为您的 Sidewalk 终端设备创建设备配置文件。以字母数字字符串形式指定要用于 Sidewalk 设备的配置文件名称。配置文件将帮助标识要与之关联的设备。
   - （控制台）添加 Sidewalk 设备时，还可以创建新的配置文件。这可以帮助您快速将设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core 并将其与配置文件关联。
   - （API）通过指定配置文件名称和 Sidewalk 对象 `sidewalk {}` 来使用 `CreateDeviceProfile` API 操作。API 响应将包含配置文件 ID 和 ARN（Amazon 资源名称）。
2. 将无线设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core 指定目标名称并选择您在上一步中创建的设备配置文件。
   - （控制台）添加 Sidewalk 设备时，输入目标名称，然后选择您创建的配置文件。
   - （API）使用 `CreateWirelessDevice` API 操作。指定目标名称和先前获取的设备配置文件的 ID。

   无线设备参数

参数	描述	备注
目标名称	目标的名称，描述用于处理其他 AWS 服务将使用的设备数据的 AWS IoT 规则。	如果您尚未创建目标，可以提供任何字符串值。适用于 Amazon Sidewalk 的 AWS IoT Core 将在创建设备时创建一个空目标，然后您可以在添加目标时对其进行更新。
设备配置文件	您之前创建的设备配置文件。	–

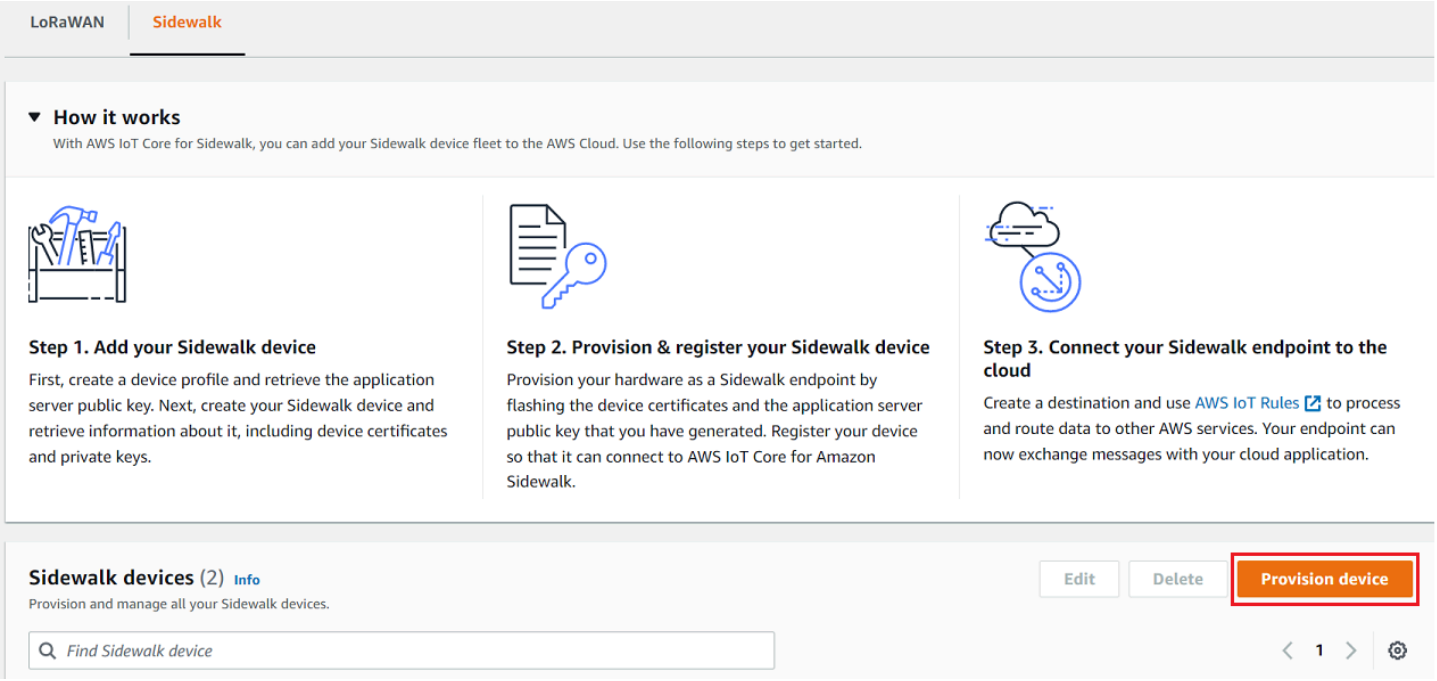
3. 获取包含预置终端设备所需信息的 JSON 文件。
   - （控制台）从您创建的 Sidewalk 设备的详细信息页面下载此文件。
   - （API）使用 `GetDeviceProfile` 和 `GetWirelessDevice` API 操作检索有关设备配置文件和无线设备的信息。将 API 响应信息存储为 JSON 文件，例如 *`device_profile.json`* 和 *`wireless_device.json`*。

添加设备配置文件和 Sidewalk 终端设备

本节向您介绍如何创建设备配置文件。它还演示如何使用 AWS IoT 控制台和 AWS CLI 将 Sidewalk 终端设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core。

添加 Sidewalk 设备（控制台）

要使用 AWS IoT 控制台添加 Sidewalk 设备，请转到设备中心的 Sidewalk 选项卡，选择预置设备，然后执行以下步骤。



1. 指定设备详细信息

   指定 Sidewalk 设备的配置信息。您也可以创建新的设备配置文件，或者为 Sidewalk 设备选择现有的配置文件。

   a. 指定设备名称和可选描述。描述长度最多为 2048 个字符。创建设备后，可以编辑这些字段。

   b. 选择要与 Sidewalk 设备关联的设备配置文件。如果您有任何现有的设备配置文件，则可以选择您的配置文件。要创建新的配置文件，请选择创建新的配置文件，然后输入配置文件的名称。

   > **Note**
   >
   > 要将标签附加到设备配置文件，请在创建配置文件后，转至配置文件中心，然后编辑配置文件以添加此信息。

c. 指定将消息从您的设备路由到其他 AWS 服务的目标的名称。如果您尚未创建目标，请前往目标中心以创建目标。然后，您可以为 Sidewalk 设备选择目标。有关更多信息，请参阅为 Sidewalk 终端设备添加目标。

d. 选择下一步以继续添加 Sidewalk 设备。

2. 将 Sidewalk 设备与 AWS IoT 事物关联（可选）

   您可以选择将 Sidewalk 设备与 AWS IoT 事物关联。IoT 事物是 AWS IoT 设备注册表中的条目。事物可让您更轻松地搜索和管理您的设备。将事物与设备关联将允许设备访问其他 AWS IoT Core 特征。

   要将设备与事物关联，请选择自动事物注册。

   a. 为要与 Sidewalk 设备关联的 IoT 事物输入唯一名称。事物名称区分大小写，并且在您的 AWS 账户和 AWS 区域中必须是唯一的。

   b. 为 IoT 事物提供任何其他配置，例如使用事物类型或可用于从事物列表中筛选的可搜索属性。

   c. 选择下一步并验证有关 Sidewalk 设备的信息，然后选择创建。

添加 Sidewalk 设备（CLI）

要添加 Sidewalk 设备并下载将用于预置 Sidewalk 设备的 JSON 文件，请执行以下 API 操作。

主题

步骤 1：创建设备配置文件

要在 AWS 账户中创建设备配置文件，请使用 CreateDeviceProfile API 操作或 create-device-profile CLI 命令。创建设备配置文件时，请指定名称并以名称/值对的形式提供任何可选标签。

例如，以下命令为 Sidewalk 设备创建设备配置文件：

```
aws iotwireless create-device-profile \
    --name sidewalk_profile --sidewalk {}
```

运行此命令会返回 Amazon 资源名称（ARN）和设备配置文件的 ID 作为输出。

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

步骤 2：添加 Sidewalk 设备

要针对适用于 Amazon Sidewalk 的 AWS IoT Core 将 Sidewalk 设备添加到您的账户中，请使用 CreateWirelessDevice API 操作或 create-wireless-device CLI 命令。创建设备时，除了 Sidewalk 设备的可选名称和描述外，还请指定以下参数。

> **Note**
>
> 如果您要将 Sidewalk 设备与 AWS IoT 事物关联，请使用 AssociateWirelessDeviceWithThing API 操作或 associate-wireless-device-with-thing CLI 命令。

以下命令显示创建 Sidewalk 设备的示例：

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

下面显示的是 device.json 文件的内容。

device.json 的内容

```
{
  "Type": "Sidewalk",
  "Name": "SidewalkDevice",
  "DestinationName": "SidewalkDestination",
  "Sidewalk": {
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
    }
}
```

运行此命令会返回设备 ID 和 Amazon 资源名称（ARN）作为输出。

```
{
```

```
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/23456789-
abcd-0123-bcde-fabc012345678",
    "Id": "23456789-abcd-0123-bcde-fabc012345678"
}
```

获取用于预置的设备 JSON 文件

将 Sidewalk 设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core 后，下载包含预置终端设备所需信息的 JSON 文件。您可以使用 AWS IoT 控制台或 AWS CLI 来检索此信息。有关如何预置设备的更多信息，请参阅《Amazon Sidewalk 文档》中的预置和注册终端设备。

获取 JSON 文件（控制台）

要获取用于预置 Sidewalk 设备的 JSON 文件，请执行以下操作：

1. 转至 Sidewalk 设备中心。
2. 选择您添加到适用于 Amazon Sidewalk 的 AWS IoT Core 的设备以查看其详细信息。
3. 在您添加的设备的详细信息页面中，选择下载设备 JSON 文件，获取 JSON 文件。

   将下载一个 `certificate.json` 文件，其中包含预置终端设备所需的信息。下面是示例 JSON 文件。它包含设备证书、私有密钥、Sidewalk 制造序列号（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 设备的详细信息页面中，您还将看到有关以下内容的信息：

- 设备 ID、其 Amazon 资源名称（ARN）以及与设备关联的任何 AWS IoT 事物的详细信息。
- 设备配置文件和目标详细信息。
- 从设备收到最后一条上行链路消息的时间。
- 表示您的设备是已预置还是已注册的状态。

获取 JSON 文件（CLI）

要使用适用于 Amazon Sidewalk 的 AWS IoT Core API 或 AWS CLI 获取用于预置 Sidewalk 终端设备的 JSON 文件，请将检索有关设备配置文件和无线设备的信息时获得的 API 响应临时保存为 JSON 文件，例如 *wireless_device.json* 和 *device_profile.json*。您将使用它们来预置 Sidewalk 设备。

以下内容说明如何检索这些 JSON 文件。

主题

步骤 1：以 JSON 文件格式获取设备配置文件信息

使用 GetDeviceProfile API 操作或 get-device-profile CLI 命令，以获取有关您针对适用于 Amazon Sidewalk 的 AWS IoT Core 添加到您的账户中的设备配置文件的信息。要检索有关设备配置文件的信息，请指定配置文件 ID。

然后，API 将返回有关与指定标识符和设备 ID 匹配的设备配置文件的信息。您可以将此响应信息保存为文件，并为其命名，例如 *device_profile.json*。

下面显示了 CLI 命令示例：

```
aws iotwireless get-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" > device_profile.json
```

运行此命令会返回设备配置文件的参数、应用程序服务器公有密钥和 DeviceTypeID。下面显示了一个 JSON 文件，其中包含来自 API 的示例响应信息。有关 API 响应中的参数的更多信息，请参阅 GetDeviceProfile。

GetDeviceProfile API 响应（***device_profile.json*** 的内容）

```
{
    "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
    }
}
```

步骤 2：以 JSON 文件形式获取 Sidewalk 设备信息

使用 GetWirelessDevice API 操作或 get-wireless-device CLI 命令，以获取有关您针对适用于 Amazon Sidewalk 的 AWS IoT Core 添加到您的账户中的 Sidewalk 设备的信息。要获取有关您的终端设备的信息，请提供您在添加设备时获得的无线设备的标识符。

然后，API 将返回有关与指定标识符和设备 ID 相匹配的设备的信息。将此响应信息保存为 JSON 文件。为文件指定一个有意义的名称，例如 *wireless_device.json*。

下面显示了使用 CLI 运行命令的示例：

```
aws iotwireless get-wireless-device --identifier-type WirelessDeviceId \
    --identifier "23456789-abcd-0123-bcde-fabc012345678" > wireless_device.json
```

运行此命令会返回设备详细信息、设备证书、私有密钥和 Sidewalk 制造序列号（SMSN）。以下内容显示了运行此命令的示例输出。有关 API 响应中的参数的更多信息，请参阅 GetWirelessDevice。

GetWirelessDevice API 响应（***wireless_device.json*** 的内容）

```
{
```

```
    "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"
    },

    ...

}
```

后续步骤

临时存储 JSON 文件 *wireless_device.json* 和 *device_profile.json*，因为您将在下一步中使用它们来预置和注册终端设备以连接到硬件平台。有关更多信息，请参阅《Amazon Sidewalk 文档》中的预置和注册终端设备。

为 Sidewalk 终端设备添加目标

使用 AWS IoT 规则处理数据和设备消息，并将其路由到其他服务。您还可以定义规则来处理从设备接收的二进制消息，并将消息转换为其他格式，使其他服务易于使用这些消息。目标将 Sidewalk 终端设备与处理要发送到其他 AWS 服务的设备数据的规则相关联。

如何创建和使用目标

1. 为目标创建 AWS IoT 规则和 IAM 角色。AWS IoT 规则指定了将处理设备的数据并路由数据以供其他 AWS 服务和您的应用程序使用的规则。IAM 角色授予访问此规则的权限。
2. 使用 `CreateDestination` API 操作为 Sidewalk 设备创建目标。指定目标名称、规则名称、角色名称和任何可选参数。API 将返回目标的唯一标识符，您可以在将终端设备添加到适用于 Amazon Sidewalk 的 AWS IoT Core 时指定该标识符。

以下内容显示了如何创建目标，以及为该目标创建 AWS IoT 规则和 IAM 角色。

主题

为 Sidewalk 设备创建目标

您可以使用目标中心或使用 `CreateDestination` 针对适用于 Amazon Sidewalk 的 AWS IoT Core 将目标添加到您的账户。创建目标时，请指定：

- 要用于 Sidewalk 终端设备的目标的唯一名称。

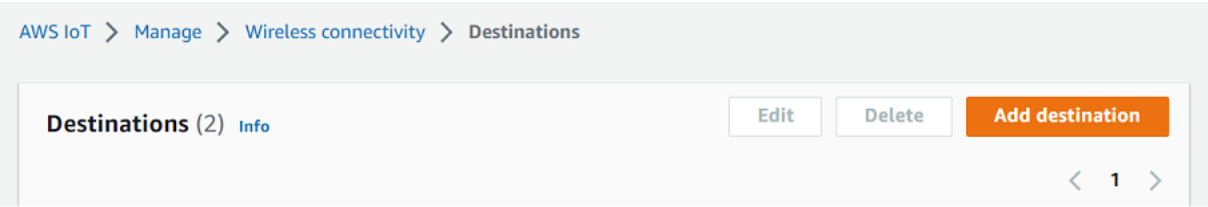
> **Note**
>
> 如果您已经使用目标名称添加设备，则在创建目标时必须使用该名称。有关更多信息，请参阅步骤 2：添加 Sidewalk 设备。

- 将用于处理设备数据的 AWS IoT 规则的名称以及要向其发布消息的主题。
- IAM 角色，向设备的数据授予访问规则的权限。

以下各节介绍如何为目标创建 AWS IoT 规则和 IAM 角色。

创建目标（控制台）

要使用 AWS IoT 控制台创建目标，请转到目标中心并选择添加目标。



要处理设备的数据，请在创建目标时指定以下字段，然后选择添加目标。

- 目标详细信息

  输入 Destination name（目标名称）以及您的目标描述（可选）。

- Rule name（规则名称）

  被配置为评估设备发送的消息并处理设备数据的 AWS IoT 规则。规则名称将映射到您的目标。目标要求规则来处理收到的消息。您可以通过调用 AWS IoT 规则或通过发布到 AWS IoT 消息代理。

- 如果选择 Enter a rule name（输入规则名称），请输入名称，然后选择 Copy（复制）以复制创建 AWS IoT 规则时要输入的规则名称。您可以选择 Create rule（创建规则）以立即创建规则，或导航到 AWS IoT 控制台的 Rule（规则）中心并使用该名称创建规则。

  您也可以输入规则并使用高级设置以指定主题名称。主题名称是在规则调用期间提供的，可通过使用 `topic` 规则中的表达式访问。有关 AWS IoT 规则的更多信息，请参阅 AWS IoT 规则。

- 如果选择 Publish to AWS IoT message broker（发布到消息代理），输入主题名称。然后，您可以复制 MQTT 主题名称，多个订阅者可以订阅此主题以接收发布到该主题的消息。有关更多信息，请参阅 MQTT 主题。

  有关目标的 AWS IoT 规则的更多信息，请参阅创建规则以处理 LoRaWAN 设备消息。

- 角色名称

  IAM 角色，该角色授予设备数据权限以访问在 Rule name（规则名称）中命名的规则。在控制台中，您可以创建新的服务角色或选择现有的服务角色。如果要创建新的服务角色，可以输入角色名称（例如，**`SidewalkDestinationRole`**），或将其留白 适用于 LoRaWAN 的 AWS IoT Core 以生成新的角色名称。然后 适用于 LoRaWAN 的 AWS IoT Core 将代表您自动创建具有适当权限的 IAM 角色。

创建目标（CLI）

要创建目标，请使用 `CreateDestination` API 操作或 `create-destination` CLI 命令。例如，以下命令为您的 Sidewalk 终端设备创建目标：

```
aws iotwireless create-destination --name SidewalkDestination \
    --expression-type RuleName --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

运行此命令会返回目标详细信息，其中包括 Amazon 资源名称（ARN）和目标名称。

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/SidewalkDestination",
    "Name": "SidewalkDestination"
}
```

有关创建目标的更多信息，请参阅创建规则以处理 LoRaWAN 设备消息。

为您的目标创建 IAM 角色和 IoT 规则

AWS IoT 规则将设备消息发送到其他服务。AWS IoT 规则还可以处理从 Sidewalk 终端设备接收的二进制消息，以供其他服务使用。适用于 Amazon Sidewalk 的 AWS IoT Core 目标将无线设备与处理设备的消息数据以发送到其他服务的规则相关联。在适用于 Amazon Sidewalk 的 AWS IoT Core 收到设备的数据后，该规则会立即对其进行处理。对于将数据发送到同一服务的所有设备，您可以创建一个可供所有设备共享的目标。您还必须创建一个 IAM 角色来授予向规则发送数据的权限。

为目标创建 IAM 角色

创建一个 IAM 角色来授予适用于 Amazon Sidewalk 的 AWS IoT Core 向 AWS IoT 规则发送数据的权限。要创建角色，请使用 `CreateRole` API 操作或 `create-role` CLI 命令。您可以将角色命名为 *`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"}]}'
```

您还可以使用 JSON 文件为角色定义信任策略。

```
aws iam create-role --role-name SidewalkRole \
    --assume-role-policy-document file://trust-policy.json
```

下面显示的是 JSON 文件的内容。

trust-policy.json 的内容

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

为目标创建规则

使用 AWS IoT Core API 操作 CreateTopicRule 或 AWS CLI 命令 create-topic-rule 来创建规则。您的目标将使用主题规则将从 Sidewalk 终端设备接收到的数据路由到其他 AWS 服务。例如，您可以创建向 Lambda 函数发送消息的规则操作。您可以定义 Lambda 函数，使其从您的设备接收应用程序数据，并使用 base64 对有效负载数据进行解码，以便其他应用程序可以使用这些数据。

以下步骤显示了如何创建 Lambda 函数，以及如何创建向该函数发送消息的主题规则。

1. 创建执行角色和策略

   创建一个 IAM 角色，向您的函数授予访问 AWS 资源的权限。您还可以使用 JSON 文件为角色定义信任策略。

   ```
   aws iam create-role --role-name lambda-ex \
       --assume-role-policy-document file://lambda-trust-policy.json
   ```

   下面显示的是 JSON 文件的内容。

   lambda-trust-policy.json 的内容

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

2. 创建并测试 Lambda 函数

   执行以下步骤以创建 base64 对有效负载数据进行解码的 AWS Lambda 函数。

   a. 编写用于解码有效负载数据的代码。例如，您可以使用以下示例 Python 代码。为脚本指定一个名称，例如 *base64_decode.py*。

      base64_decode.py 的内容

```
// ------------------------------------------------------------
// ----- 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. 将部署包创建为包含 Python 文件的 zip 文件，并将其命名为 *base64_decode.zip*。使用 `CreateFunction` API 或 `create-function` CLI 命令为示例代码创建 Lambda 函数 *base64_decode.py*。

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
```

您应当看到如下输出。创建主题规则时，您将使用输出中的 Amazon 资源名称（ARN）值 `FunctionArn`。

```
{
    "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. 要从命令行获取调用的日志，请将 `--log-type` 选项与 `invoke` 命令一起使用。响应包含一个 LogResult 字段，该字段包含来自调用的多达 4KB 的 base64 编码日志。

```
aws lambda invoke --function-name my-function out --log-type Tail
```

您应该会收到 `StatusCode` 为 200 的响应。有关从 AWS CLI 创建和使用 Lambda 函数的更多信息，请参阅将 Lambda 与 AWS CLI 结合使用。

3. 创建主题规则

使用 `CreateTopicRule` API 或 `create-topic-rule` CLI 命令创建向此 Lambda 函数发送消息的主题规则。您还可以添加第二个规则操作，以重新发布到 AWS IoT 主题。将此主题规则命名为 *Sidewalkrule*。

```
aws iot create-topic-rule --rule-name Sidewalkrule \
    --topic-rule-payload file://myrule.json
```

您可以使用 `myrule.json` 文件来指定有关规则的更多详细信息。例如，以下 JSON 文件显示了如何重新发布到 AWS IoT 主题以及如何向 Lambda 函数发送消息。

```
{
    "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/SidewalkRepublishRole",
                 "topic": "project/sensor/observed"
             }
```

```
        }
    ],
}
```

连接您的 Sidewalk 设备并查看上行链路元数据格式

在本教程中，您将使用 MQTT 测试客户端来测试连接，并查看在终端设备与 AWS Cloud之间交换的消息。要接收消息，请在 MQTT 测试客户端中，订阅在为目标创建 IoT 规则时指定的主题。也可以使用 SendDataToWirelessDevice API 操作从适用于 Amazon Sidewalk 的 AWS IoT Core 向您的设备发送下行链路消息。您可以通过启用消息传送状态事件通知来验证消息是否已送达。

> **Note**
>
> 有关连接和设置硬件平台的信息，请参阅《Amazon Sidewalk 文档》中的预置和注册终端设备和设置硬件开发工具包（HDK）。

向终端设备发送下行链路消息

使用 SendDataToWirelessDevice API 操作或 send-data-to-wireless-device CLI 命令从适用于 Amazon Sidewalk 的 AWS IoT Core 将下行链路消息发送到您的 Sidewalk 终端设备。下面显示了如何运行此命令的示例。有效负载数据是要发送的二进制文件，以 base64 编码。

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

以下显示了运行此命令的示例输出，也即发送到设备的下行链路消息的 ID。

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

> **Note**
>
> SendDataToWirelessDevice API 可以返回消息 ID，但消息可能未能成功传送。要检查发送到设备的消息的状态，您可以为您 Sidewalk 账户和设备启用消息传送状态事件。有关如何

启用此事件的信息，请参阅Sidewalk 资源的事件通知。有关此事件类型的更多信息，请参阅消息传送事件。

查看来自设备的上行链路消息的格式

在连接设备之后，您可以订阅在创建目标规则时指定的主题（例如，*project/sensor/observed*），并观察来自设备的上行链路消息。

如果您在创建目标时指定了主题名称，则可以订阅该主题以监控来自终端设备的上行链路消息。转到 AWS IoT 控制台的测试页面上的 MQTT 测试客户端，输入主题名称（例如，*project/sensor/observed*），然后选择订阅。

下面的示例显示了从 Sidewalk 设备发送到 AWS IoT 的上行链路消息格式。WirelessMetadata 包含有关消息请求的元数据。

```
{
   "PayloadData":"ZjRlNjY1ZWNlNw==",
   "WirelessDeviceId":"wireless_device_id",
   "WirelessMetadata":{
      "Sidewalk":{
         "CmdExStatus":"Cmd",
         "SidewalkId":"device_id",
         "Seq":0,
         "MessageType":"messageType"
      }
    }
}
```

下表显示了上行链路元数据中不同参数的定义。*device-id* 是无线设备的 ID（例如 *ABCDEF1234*），而 *messageType* 是从设备接收的上行链路消息的类型。

Sidewalk 上行链路元数据参数

参数	描述	类型	必需
PayloadData	从无线设备发送的消息有效负载。	字符串	是
WirelessDeviceID	发送数据的无线设备的标识符	字符串	是
Sidewalk.CmdExStatus	命令运行时状态。响应类型的消息应包含状态代码，COMMAND_E	枚举	否

参数	描述	类型	必需
	XEC_STATUS_SUCCESS 。但是，通知可能不包含状态代码。		
Sidewalk.NackExStatus	响应 nack 状态，可以为 RADIO_TX_ERROR 或者 MEMORY_ERROR 。	字符串数组	否

使用适用于 Amazon Sidewalk 的 AWS IoT Core 批量预置设备

您可以使用批量预置功能，将大量终端设备批量登记到适用于 Amazon Sidewalk 的 AWS IoT Core。批量预置非常有用，特别是当您在工厂制造大量设备并希望将这些设备登记到 AWS IoT 时。有关制造设备的更多信息，请参阅《Amazon Sidewalk 文档》中的制造 Amazon Sidewalk 设备。

下列主题向您展示了批量预置的工作原理。

- Amazon Sidewalk 批量预置工作流

  本主题向您展示批量预置的一些关键概念及其工作原理。其中还显示了将 Sidewalk 设备导入适用于 Amazon Sidewalk 的 AWS IoT Core 必须执行的步骤。
- 在工厂支持下创建设备配置文件

  本主题说明如何创建设备配置文件并获得工厂支持。您还将了解如何检索 YubiHSM 密钥并将其发送给制造商，以便在设备制造完成后获取控制日志。
- 使用导入任务预置 Sidewalk 设备

  本主题介绍如何通过创建和使用导入任务来批量预置 Sidewalk 设备。您还将了解如何更新或删除导入任务，以及查看导入任务及任务中设备的状态。

主题

- Amazon Sidewalk 批量预置工作流
- 在工厂支持下创建设备配置文件
- 使用导入任务预置 Sidewalk 设备

Amazon Sidewalk 批量预置工作流

以下各节向您展示了批量预置的关键概念及其工作原理。批量预置所涉及的步骤包括：

1. 使用适用于 Amazon Sidewalk 的 AWS IoT Core 创建设备配置文件。
2. 向 Amazon Sidewalk 团队请求 YubiHSM 密钥，并在工厂支持下更新您的设备配置文件。
3. 将 YubiHSM 密钥发送给您的制造商，以便适用于 Amazon Sidewalk 的 AWS IoT Core 可以在设备制造完成后获取控制日志。
4. 创建导入任务并提供要登记到适用于 Amazon Sidewalk 的 AWS IoT Core 的设备的序列号（SMSN）。

批量预置的组件

以下概念向您展示了批量预置的一些关键组件，以及如何在批量预置 Sidewalk 设备时使用这些组件。

YubiHSM 密钥

Amazon 会为您的每个 Sidewalk 产品创建一个或多个 HSM（硬件安全模块）。每个 HSM 都有一个唯一的序列号，称为 YubiHSM 密钥，印在硬件模块上。可以从 Yubico 网页购买此密钥。

该密钥对每个 HSM 都是唯一的，并与您使用适用于 Amazon Sidewalk 的 AWS IoT Core 创建的每个设备配置文件相关联。要获取 YubiHSM 密钥，请联系 Amazon Sidewalk 团队。如果您将 YubiHSM 密钥发送给制造商，则在工厂中制造 Sidewalk 设备后，适用于 Amazon Sidewalk 的 AWS IoT Core 将收到包含设备序列号的控制日志文件。然后，它将此信息与您的输入 CSV 文件进行比较，以便将设备登记到 AWS IoT。

设备认证密钥（**DAK**）

当 Sidewalk 终端设备加入 Sidewalk 网络时，必须为其预置 Sidewalk 设备证书。用于设置设备的证书包括私有设备特定的证书和公有设备证书，后者对应于 Sidewalk 证书链。当 Sidewalk 设备完成制造时，YubiHSM 会对设备证书进行签名。

下面显示了一个包含设备证书和私有密钥的 JSON 文件示例。有关更多信息，请参阅获取用于预置的设备 JSON 文件。

```
{
  "p256R1": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbH ... DANKkOKoNT3bUGz+/f/pyTE
+xMRdIUBZ1Bw==",
  "eD25519": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbHD ... UiZmntHiUr1GfkTOFMYqRB+Aw==",
  "metadata": {
    "devicetypeid": "fe98",
```

```
    ...

    "devicePrivKeyP256R1":
 "3e704bf8d319b3a475179f1d68c60737b28c708f845d0198f2d00d00c88ee018",
    "devicePrivKeyEd25519":
 "17dacb3a46ad9a42d5c520ca5f47f0167f59ce54d740aa13918465faf533b8d0"
  },
  "applicationServerPublicKey":
 "5ce29b89c2e3ce6183b41e75fe54e45f61b8bb320efbdd2abd7aefa5957a316b"
}
```

设备认证密钥（DAK）是您在创建设备配置文件时获得的私有密钥。它对应于产品证书，这是颁发给每个 Sidewalk 产品的唯一证书。当您联系 Amazon Sidewalk 团队时，您将收到 Sidewalk 证书链、YubiHSM 密钥和预置了产品设备认证密钥（DAK）的 HSM。

您的设备配置文件还会使用新的设备认证密钥（DAK）进行更新，并启用工厂支持。设备配置文件的 DAK 元数据信息提供诸如 DAK 名称、证书 ID、ApId（公告产品 ID）、是否启用了工厂支持以及 DAK 可以签名的最大签名数等详细信息。

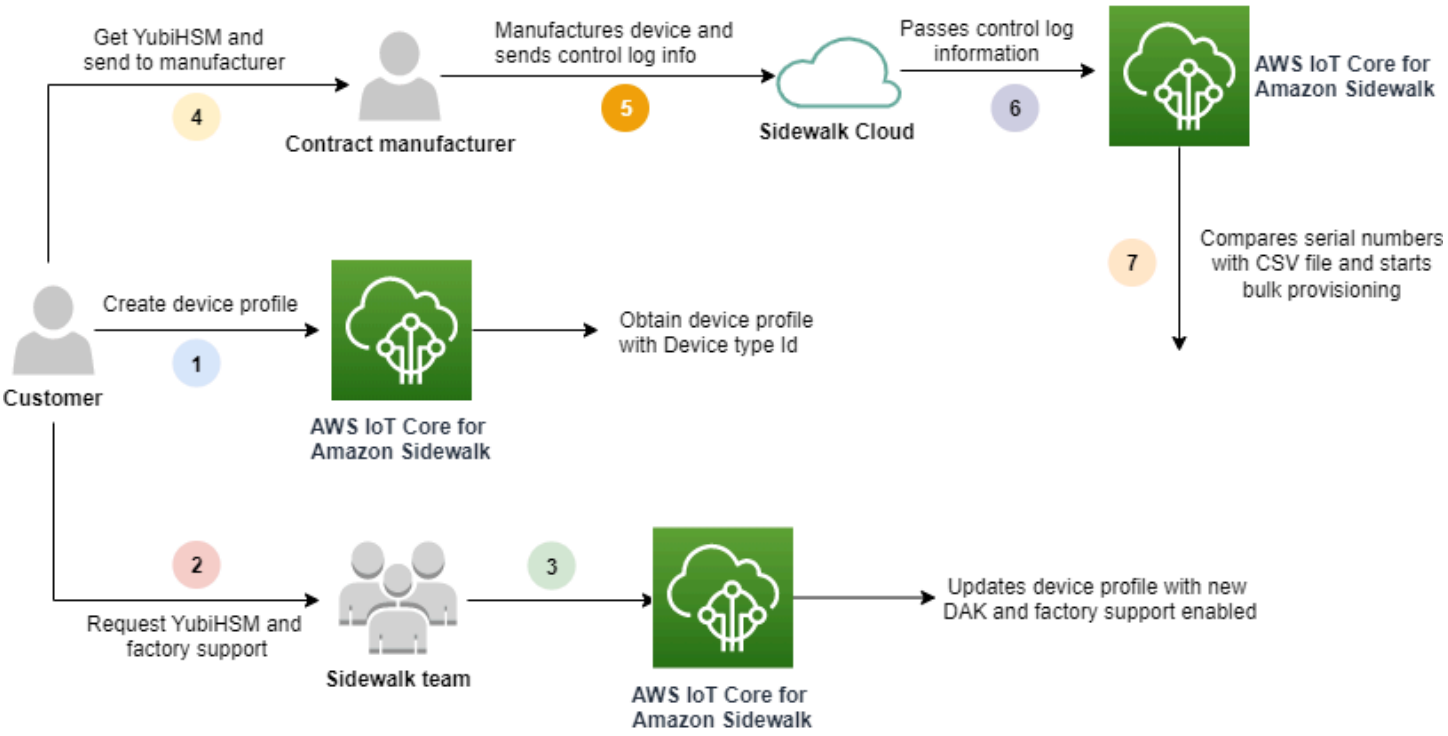
公告产品编号（**ApId**）

ApId 参数是一个用于标识公告产品的字母数字字符串。当您要为您批量预置的 Sidewalk 设备使用给定的设备配置文件时，必须指定此字段。然后适用于 Amazon Sidewalk 的 AWS IoT Core 生成 DAK，并通过 YubiHSM 密钥将其提供给您。相关的 DAK 信息将显示在设备配置文件中。

要获取 ApId，在检索有关您创建的设备配置文件的信息后，请联系 Amazon Sidewalk Support 团队。您可以从 AWS IoT 控制台或使用 GetDeviceProfile API 操作或 get-device-profile CLI 命令来获取设备配置文件信息。

批量预置的工作原理

此流程图显示了使用适用于 Amazon Sidewalk 的 AWS IoT Core 进行批量预置的工作原理。



以下过程说明了批量预置过程中的不同步骤。

1. 为 Sidewalk 设备创建设备配置文件

   在将终端设备送往工厂之前，首先创建设备配置文件。您可以使用此配置文件来预置各台设备，如添加设备配置文件和 Sidewalk 终端设备中所述。

2. 为配置文件请求工厂支持

   当您准备好将终端设备送到工厂时，请向 Amazon Sidewalk 团队索取 YubiHSM 密钥，并为设备配置文件请求工厂支持。

3. 获取 DAK 和工厂支持的配置文件

   然后，Amazon Sidewalk Support 团队将使用产品设备认证密钥（DAK）和工厂支持更新您的设备配置文件。您的设备配置文件将自动使用公告产品 ID（ApID）、新的 DAK 和证书信息（如证书 ID）进行更新。使用此配置文件的 Sidewalk 设备有资格用于批量预置。

4. 将 YubiHSM 密钥发送给制造商（CM）

   您的终端设备现已具有资格，因此您可以将 YubiHSM 密钥发送给合同制造商（CM）以开始制造过程。有关更多信息，请参阅《Amazon Sidewalk 文档》中的制造 Amazon Sidewalk 设备。

5. 制造设备并发送控制日志和序列号

   CM 制造设备并生成控制日志。CM 还为您提供一个 CSV 文件，其中包含待制造的设备列表及其 Sidewalk 制造序列号（SMSN）。以下代码显示了示例控制日志。它包含设备的序列号、APID 和公有设备证书。

```
{
    "controlLogs": [
    {
        "version": "4-0-1",
        "device":
        {
            "serialNumber": "device1",
            "productIdentifier": {
                "advertisedProductId": "abCD"
             },
             "sidewalkData": {
                "SidewalkED25519CertificateChain": "...",
                "SidewalkP256R1CertificateChain": "..."
             }
         }
      }
   ]
}
```

6. 将控制日志信息传递给适用于 Amazon Sidewalk 的 AWS IoT Core

   Amazon Sidewalk 云从制造商处检索控制日志信息，并将此信息传递给适用于 Amazon Sidewalk 的 AWS IoT Core。然后，可以创建设备及其序列号。

7. 检查序列号匹配并开始批量预置

   当您使用 AWS IoT 控制台或适用于 Amazon Sidewalk 的 AWS IoT Core API 操作 `StartWirelessDeviceImportTask` 时，适用于 Amazon Sidewalk 的 AWS IoT Core 将从 Amazon Sidewalk 获得的每台设备的 Sidewalk 制造序列号（SMSN）与 CSV 文件中的相应序列号进行比较。如果此信息匹配，它将启动批量预置过程并创建要导入适用于 Amazon Sidewalk 的 AWS IoT Core 的设备。

在工厂支持下创建设备配置文件

在批量预置您的 Amazon Sidewalk 设备之前，您必须创建设备配置文件，然后联系 Amazon Sidewalk Support 团队为其请求工厂支持。接着，Amazon Sidewalk 团队将使用新的设备认证密钥（DAK）更新您的设备配置文件，并为其添加工厂支持。然后，使用此配置文件的 Sidewalk 设备才有资格与适用于 Amazon Sidewalk 的 AWS IoT Core 配合使用，并且可以登记这些设备以进行批量预置。

下列步骤向您展示如何创建工厂支持的设备配置文件。

1. 创建设备配置文件

   首先创建设备配置文件。创建配置文件时，请将名称和可选标签指定为名称/值对。有关所需参数以及创建和使用配置文件的更多信息，请参阅如何创建和添加设备。

2. 为配置文件获取工厂支持

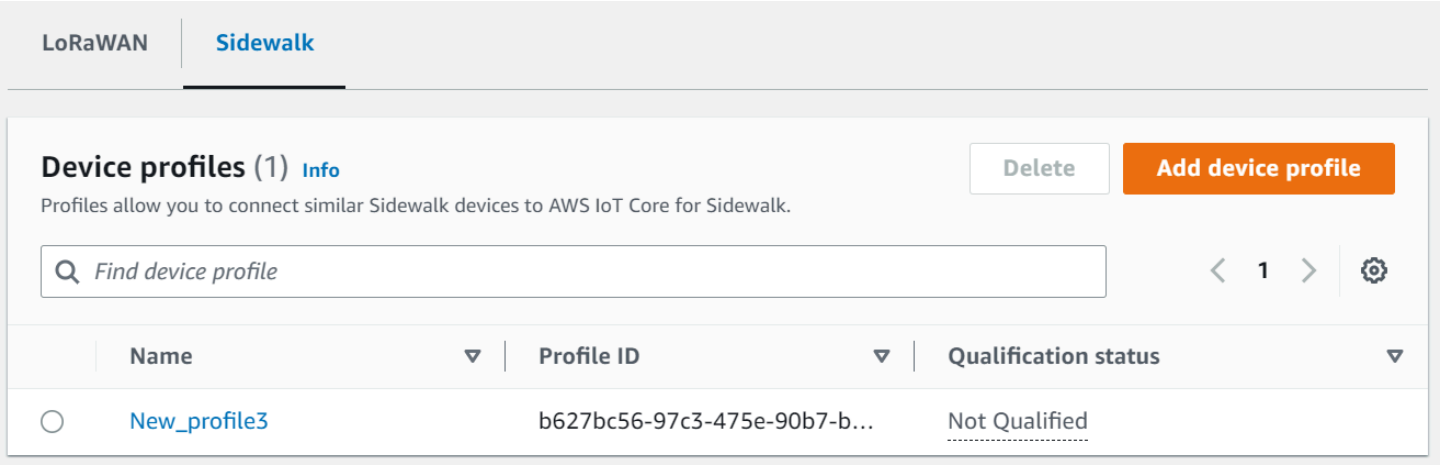
   然后，获取对设备配置文件的工厂支持，以便使用此配置文件的设备可以获得资格。要获得资格，请创建提交给 Amazon Sidewalk 团队的支持单。经该团队确认后，您将收到一个 ApId（公告产品 ID），并且您的配置文件将使用工厂签发的 DAK 进行更新。使用此配置文件的 Sidewalk 终端设备将获得资格。

您可以使用 AWS IoT 控制台、适用于 Amazon Sidewalk 的 AWS IoT Core API 操作或 AWS CLI 创建设备配置文件。

主题

创建配置文件（控制台）

要使用 AWS IoT 控制台创建设备配置文件，请转到配置文件中心的 Sidewalk 选项卡，然后选择创建配置文件。



要创建配置文件，请指定以下字段，然后选择提交。

- 名称

  输入配置文件的名称。

- 标签

  输入可选标签作为名称/值对，以帮助您更轻松地标识您的配置文件。标签还可以更轻松地跟踪账单费用。

查看配置文件信息并使配置文件获得资格

您将在配置文件中心看到您创建的配置文件。选择配置文件可查看其详细信息。您将看到有关以下内容的信息：

- 设备配置文件名称和唯一标识符，以及您指定为名称值对的任何可选标签。
- 配置文件的应用程序服务器公有密钥和设备类型 ID。
- 资格状态，表示您使用的设备配置文件不受工厂支持。要使您的设备配置文件获得资格以使其受工厂支持，请联系 Amazon Sidewalk Support。
- 设备认证密钥（DAK）信息。一旦您的设备配置文件获得资格，将颁发新的 DAK，并使用新的 DAK 信息自动更新您的配置文件。

创建配置文件（CLI）

要创建设备配置文件，请使用 `CreateDeviceProfile` API 操作或 `create-device-profile` CLI 命令。例如，以下命令可以为 Sidewalk 终端设备创建配置文件。

```
aws iotwireless create-device-profile \
    --name sidewalk_device_profile --sidewalk {}
```

运行此命令会返回配置文件详细信息，包括 Amazon 资源名称（ARN）和配置文件的 ID。

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

查看配置文件信息并使配置文件获得资格

使用 GetDeviceProfile API 操作或 get-device-profile CLI 命令，以获取有关您针对适用于 Amazon Sidewalk 的 AWS IoT Core 添加到您的账户中的设备配置文件的信息。要检索有关设备配置文件的信息，请指定配置文件 ID。然后，API 将返回有关与指定标识符匹配的设备配置文件的信息。

下面显示了 CLI 命令示例：

```
aws iotwireless get-device-profile \
    --id "12345678-234a-45bc-67de-e8901234f0a1" > device_profile.json
```

运行此命令会返回设备配置文件的参数、应用程序服务器公有密钥、DeviceTypeId、ApId、资格状态和 DAKCertificate 信息。

在此示例中，资格状态和 DAK 信息表明您的设备配置文件未获得资格。要使配置文件获得资格，请联系 Amazon Sidewalk Support，您的配置文件将获得一个没有设备限制的新 DAK。

```
{
    "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 Support 团队确认此信息后，您将收到 APID 和工厂支持的 DAK，如以下示例所示。

> **Note**
>
> `MaxAllowedSignature` 为 `-1` 表示 DAK 没有任何设备限制。有关 DAK 参数的信息，请参阅 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
    }
}
```

后续步骤

现在，您已经创建了具有工厂支持的 DAK 的设备配置文件，请将您从该团队获得的 YubiHSM 密钥提供给制造商。然后，您的设备将在工厂制造。这之后，控制日志信息将传递给 Amazon Sidewalk，其中包含设备的序列号（SMSN）。有关此工作流的更多信息，请参阅《Amazon Sidewalk 文档》中的制造 Amazon Sidewalk 设备。

然后，您可以通过向适用于 Amazon Sidewalk 的 AWS IoT Core 提供要登记的设备的序列号来批量预置 Sidewalk 设备。当适用于 Amazon Sidewalk 的 AWS IoT Core 收到控制日志时，它会将控制日志中的序列号与您提供的序列号进行比较。如果序列号匹配，则导入任务将开始将您的设备登记到适用于 Amazon Sidewalk 的 AWS IoT Core。有关更多信息，请参阅使用导入任务预置 Sidewalk 设备。

使用导入任务预置 Sidewalk 设备

本节介绍如何使用 AWS IoT 控制台、适用于 Amazon Sidewalk 的 AWS IoT Core API 操作或 AWS CLI 批量预置 Sidewalk 设备。下列各部分说明如何批量预置 Sidewalk 设备。

主题

Sidewalk 批量预置的工作原理

以下步骤说明了批量预置的工作原理。

1. 启动无线设备导入任务

   要批量预置 Sidewalk 设备，必须创建导入任务并提供要登记到适用于 Amazon Sidewalk 的 AWS IoT Core 的设备的 Sidewalk 制造序列号（SMSN）。在制造商将控制日志上传到 Amazon Sidewalk 后，您便通过电子邮件，以 CSV 文件的形式获得了设备的 Sidewalk 制造序列号（SMSN）。有关工作流以及如何获取控制日志的更多信息，请参阅《Amazon Sidewalk 文档》中的制造 Amazon Sidewalk 设备。

2. 在后台运行导入进程

   当适用于 Amazon Sidewalk 的 AWS IoT Core 收到导入任务请求时，它开始设置事物，并启动一个频繁轮询系统的后台进程。后台进程收到导入任务指令后，就会开始读取 CSV 文件。适用于 Amazon Sidewalk 的 AWS IoT Core 同时检查是否已从 Amazon Sidewalk 收到控制日志。

3. 创建无线设备记录

   从 Amazon Sidewalk 收到控制日志后，适用于 Amazon Sidewalk 的 AWS IoT Core 会检查控制日志中的序列号是否与 CSV 文件中的 SMSN 值匹配。如果序列号匹配，适用于 Amazon Sidewalk 的 AWS IoT Core 将开始为与这些序列号对应的 Sidewalk 设备创建无线设备记录。登记所有设备后，导入任务将标记为已完成。

Sidewalk 批量预置的关键注意事项

在将 Sidewalk 设备批量预置到适用于 Amazon Sidewalk 的 AWS IoT Core 时，请关注以下一些主要注意事项。

- 您必须在创建设备配置文件的同一 AWS 账户中使用 AWS IoT 控制台或适用于 Amazon Sidewalk 的 AWS IoT Core API 操作执行批量预置。
- 在批量预置 Sidewalk 设备之前，设备配置文件必须已经包含表明工厂支持的 DAK 信息。否则，使用 AWS IoT 控制台进行批量预置或批量预置 API 操作可能会失败。
- 开始导入任务后，处理 CSV 文件、导入无线设备并将其登记到适用于 Amazon Sidewalk 的 AWS IoT Core 可能至少需要 10 分钟或更长时间。
- 无线设备导入任务一旦启动，就将运行达 90 天。在此期间，它会检查是否已从 Amazon Sidewalk 收到控制日志。如果在 90 天届满前未从 Amazon Sidewalk 收到控制日志，则当您查看任务详细信息时，该任务将标记为已完成，并显示一条消息表明该任务已过期。导入任务中等待控制日志的设备的登记状态将标记为失败。
- 当您尝试更新已创建的导入任务时，只能向该任务添加其他设备。在创建导入任务之后和在已添加到导入任务的设备上启动任务之前，您可以随时添加新设备。如果更新文件包含原始导入任务中已存在的设备的序列号，则这些序列号将被忽略。
- 当您请求更新操作时，将代入与您在创建导入任务时使用的相同 IAM 角色来访问 Amazon S3 桶中的 CSV 文件。
- 只有当导入任务已成功完成或任务更新失败时，才能删除该任务。在提供错误的 IAM 角色或找不到 Amazon S3 桶文件等情况下，任务可能无法更新。如果导入任务处于 `PENDING` 状态，则无法更新或删除此任务。
- 您导入到任务中的 CSV 文件必须使用下一部分中描述的格式。

CSV 文件格式

Amazon S3 桶中所包含的您为导入任务指定的 CSV 文件必须使用以下格式：

- 第 1 行必须使用关键字 `smsn`，表示正在导入的 CSV 文件包含要导入的设备的 SMSN。
- 第 2 行及之后的行必须包含要登记的设备的 SMSN。设备 SMSN 必须采用 64 个十六进制字符格式。

此 JSON 文件显示了一个 CSV 文件格式示例。

```
smsn
1C1A10B0AC0A200C012BBAC2CBB1B21CB12C0CA2AC1C1BB22CAA01C1B0B01122
B122C2B1121BACA2221001AC1B22012AAC11112C11C2A100C1C2B012A1100C10
02B222C110B0A210B0A0C2C112CCCAC21C1C0B0AA1221AB1022A2CC11B1B1122
C2C021CA1C111CCAB1221C0021C1C2AAA0AA1A2A01ABC10CBAACCA2A0121022A
0CB22C01BBC2CA2C0B11001121ACB2ABB0BB0121C2BA101C012CC2B20C011AC0
```

如何使用 Sidewalk 批量预置

下列步骤向您展示如何使用 Amazon Sidewalk 批量预置。

1. 提供设备序列号

   要预置 Sidewalk 设备，您必须提供要登记的设备的序列号。您可以使用以下任一方法来预置设备。

   - 使用各台设备的 Sidewalk 制造序列号（SMSN）单独预置每台设备。如果您想测试工作流并更快地登记设备，而不必使用相应的 IAM 角色上传 CSV 文件，也不必等待设备准备好登记到此任务，则此方法很有用。
   - 通过提供 Amazon S3 桶 URL 来批量预置设备，该 URL 带有包含待预置设备的 SMSN 的 CSV 文件路径。当您有大量设备要登记时，此方法尤其有用。在这种情况下，单独登记每台设备可能工作量很大。相反，您只需提供指向已上传到 Amazon S3 桶的 CSV 文件的路径以及访问该文件的 IAM 角色即可。

2. 获取导入任务和设备登记状态

   对于您创建的每个导入任务，您可以检索有关任务登记状态和添加到任务中的设备的登记状态的信息。您还可以查看其他状态信息，例如任务或设备登记失败的原因。有关更多信息，请参阅

3. （可选）更新或删除导入任务

   您可以更新或删除已创建的导入任务。

   - 在已添加的设备上开始导入任务之前，您可以随时更新导入任务并向该任务添加其他设备。适用于 Amazon Sidewalk 的 AWS IoT Core 代入您在创建导入任务时使用的相同 IAM 角色。创建任务时，请指定新的 CSV 文件，其中包含要添加到任务中的设备的序列号。

     > **Note**
     >
     > 更新现有导入任务时，只能向任务中添加设备。适用于 Amazon Sidewalk 的 AWS IoT Core 在导入任务中已经存在的设备与您试图添加到任务中的设备之间执行并集操作。如果新文件包含导入任务中已存在的设备的序列号，则这些序列号将被忽略。

   - 您可以删除已成功完成的导入任务，也可以删除在 IAM 角色信息不正确或在创建或更新任务时 S3 桶文件不可用等情况下未能更新的导入任务。

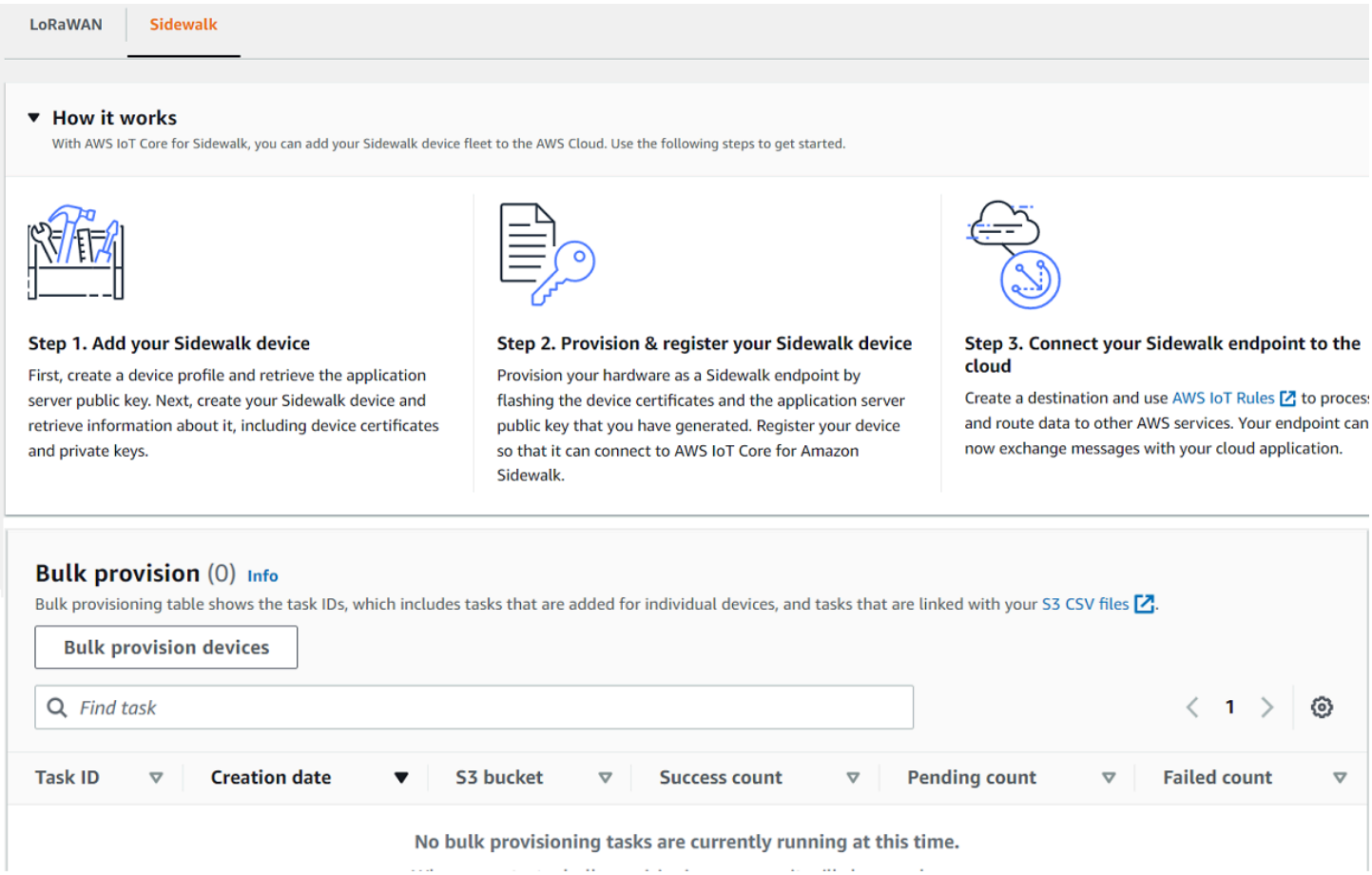
主题

批量预置 Sidewalk 设备

本节介绍如何使用 AWS IoT 控制台和 AWS CLI 将 Sidewalk 设备批量预置到适用于 Amazon Sidewalk 的 AWS IoT Core。

批量预置 Sidewalk 设备（控制台）

要使用 AWS IoT 控制台添加 Sidewalk 设备，请转到设备中心的 Sidewalk 选项卡，选择批量预调配设备，然后执行以下步骤。



1. 选择导入方法

   指定如何将要登记的设备批量导入适用于 Amazon Sidewalk 的 AWS IoT Core。

   - 要使用 SMSN 预置单个设备，请选择预置单个受工厂支持的设备。
   - 要通过提供包含设备列表及其 SMS 的 CSV 文件来批量预置设备，请选择使用 S3 桶。

2. 指定要登记的设备

   根据您选择登记设备的方法，添加设备信息及其序列号。

   a. 如果您选择预置单个受工厂支持的设备，请指定以下信息：

      i. 要登记的每个设备的名称。名称在您的 AWS 账户和 AWS 区域中必须是唯一的。

      ii. 设备的 Sidewalk 制造序列号（SMSN）位于输入 SMSN 字段中。

      iii. 一个目标，描述将消息从设备路由到其他 AWS 服务的 IoT 规则。

   b. 如果您选择使用 S3 桶：

i. 提供 S3 桶目标信息，其中包含 S3 URL 信息。要提供您的 CSV 文件，请选择浏览 S3，然后选择要使用的 CSV 文件。

适用于 Amazon Sidewalk 的 AWS IoT Core 会自动填充 S3 URL，这是指向 S3 桶中 CSV 文件的路径。`s3://bucket_name/file_name` 路径的格式为：要在 Amazon Simple Storage Service控制台中查看文件，选择 View（查看）。

ii. 提供 S3 预置角色，该角色允许适用于 Amazon Sidewalk 的 AWS IoT Core 代表您访问 S3 桶中的 CSV 文件。您可以创建新的服务角色或选择现有角色。

要创建新角色，您可以提供角色名称，也可以将其留空以自动生成随机名称。

iii. 提供一个目标，描述将消息从设备路由到其他 AWS 服务的 IoT 规则。

3. 开始导入任务

提供任何可选标签作为名称/值对，然后选择提交以开始无线设备导入任务。

批量预置 Sidewalk 设备（CLI）

要针对适用于 Amazon Sidewalk 的 AWS IoT Core 将 Sidewalk 设备登记到您的账户，请使用以下任一 API 操作，具体取决于您是要单独添加设备，还是通过提供 S3 桶中包含的 CSV 文件来添加设备。

- 使用 S3 CSV 文件批量上传设备

要通过提供 S3 桶中的 CSV 文件来批量上传设备，请使用 `StartWirelessDeviceImportTask` API 操作或 `start-wireless-device-import-task` AWS CLI 命令。创建任务时，请指定指向 Amazon S3 桶中 CSV 文件的路径，以及授予适用于 Amazon Sidewalk 的 AWS IoT Core 访问 CSV 文件的权限的 IAM 角色。

任务开始运行后，适用于 Amazon Sidewalk 的 AWS IoT Core 将开始读取 CSV 文件，并将文件中的序列号（SMSN）与从 Amazon Sidewalk 收到的控制日志中的相应信息进行比较。当序列号匹配时，它将开始创建与这些序列号对应的无线设备记录。

以下命令显示了创建导入任务的示例：

```
aws iotwireless start-wireless-device-import-task \
    --cli-input-json "file://task.json"
```

下面显示的是 `task.json` 文件的内容。

task.json 的内容

```
{
    "DestinationName": "Sidewalk_Destination",
    "Sidewalk": {
        "DeviceCreationFile": "s3://import_task_bucket/import_file1",
        "Role": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI"
    }
}
```

运行此命令会返回导入任务的 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 SMSN 单独预置设备

  要使用 SMSN 单独预置设备，请使用 StartSingleWirelessDeviceImportTask API 操作或 start-single-wireless-device-import-task AWS CLI 命令。创建任务时，请指定 Sidewalk 目标和要登记的设备的序列号。

  当序列号与从 Amazon Sidewalk 收到的控制日志中的相应信息匹配时，任务将运行并创建无线设备记录。

  以下命令显示了创建导入任务的示例：

```
aws iotwireless start-single-wireless-device-import-task \
    --destination-name sidewalk_destination \
    --sidewalk
 '{"SidewalkManufacturingSn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A
```

  运行此命令会返回导入任务的 ID 和 ARN。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
    "Id": "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

```
}
```

更新或删除导入任务

如果要向导入任务中添加其他设备，可以更新该任务。如果您不再需要该任务或任务失败，也可以删除该任务。有关何时更新或删除任务的信息，请参阅[如何使用 Sidewalk 批量预置]。

> **Warning**
>
> 删除是永久性操作，无法撤消。删除已成功完成的导入任务将不会移除已使用该任务登记的终端设备。

要更新或删除导入任务，请执行以下操作：

- 使用 AWS IoT控制台

  下列步骤说明如何使用 AWS IoT 控制台更新或删除导入任务。

  要更新导入任务，请执行下列操作：

  1. 转至 AWS IoT 控制台的 Sidewalk 设备中心。
  2. 选择要更新的导入任务，然后选择编辑。
  3. 提供另一个 S3 文件，其中包含要添加到任务中的设备的序列号，然后选择提交。

  要删除导入任务，请执行以下操作：

  1. 转至 AWS IoT 控制台的 Sidewalk 设备中心。
  2. 选择要删除的任务，然后选择删除。

- 使用 AWS IoT Wireless API 或 AWS CLI

  使用以下 AWS IoT Wireless API 操作或 CLI 命令更新或删除您的导入任务。

  - **`UpdateWirelessDeviceImportTask`** API 或 **`update-wireless-device-import-task`** CLI

    此 API 操作将 Amazon S3 CSV 文件的内容附加到现有的导入任务。您只能添加以前未包含在任务中的设备的序列号。

- **DeleteWirelessDeviceImportTask** API 或 **delete-wireless-device-import-task** CLI

  此 API 操作使用导入任务 ID 删除标记为待删除的导入任务。

查看导入任务和设备登记状态

您的无线设备导入任务和添加到该任务中的 Sidewalk 设备可能具有以下状态消息之一。您将看到这些消息显示在 AWS IoT 控制台中，或者，当您使用任何 AWS IoT Wireless API 操作或 AWS CLI 命令来检索有关这些任务及其设备的信息时，也会看到这些信息。

查看导入任务状态信息

创建导入任务后，您可以查看您创建的导入任务以及添加到该任务中的设备的登记状态。登记状态指示待登记的设备数量、已成功登记的设备数量以及未能登记的设备数量。

刚创建导入任务后，待处理计数将显示一个与已添加的设备数量相对应的值。任务开始并读取 CSV 文件以创建无线设备记录后，随着设备成功登记，待处理计数将减少，而成功计数将增加。如果任何设备登记失败，则失败计数将增加。

要查看导入任务和设备登记状态，请执行以下操作：

- 使用 AWS IoT控制台

  在 AWS IoT 控制台的 Sidewalk 设备中心中，您可以看到您创建的导入任务以及设备登记状态信息摘要的计数。如果您查看您创建的任何导入任务的详细信息，则可以看到有关设备登记状态的其他信息。
- 使用 AWS IoT Wireless API 或 AWS CLI

  要查看设备登记状态，请使用以下任一 AWS IoT Wireless API 操作或相应的 AWS CLI 命令。

  - **ListWirelessDeviceImportTasks** API 或 **list-wireless-device-import-tasks** CLI

    此 API 操作返回有关已针对 AWS IoT Wireless 添加到您的账户中的所有导入任务及其状态的信息。它还会返回这些任务中 Sidewalk 设备登记状态摘要的计数。
  - **ListDevicesForWirelessDeviceImportTask** API 或 **list-devices-for-wireless-device-import-task** CLI

    此 API 操作返回有关指定的导入任务及其状态的信息，以及有关已添加到导入任务中的所有 Sidewalk 设备及其登记状态信息。

- **GetWirelessDeviceImportTask** API 或 **get-wireless-device-import-task** CLI

  此 API 操作返回有关指定的导入任务及其状态的信息，以及该任务中 Sidewalk 设备的登记状态摘要的计数。

导入任务状态

在您的 AWS 账户中创建的导入任务可能具有以下状态消息之一。状态表示您的导入任务是已开始处理、已完成还是失败。还可以使用 AWS IoT 控制台或任何 AWS IoT Wireless API 操作的 `StatusReason` 参数来检索其他状态详细信息。

- INITIALIZING

  适用于 Amazon Sidewalk 的 AWS IoT Core 已收到无线设备导入任务请求并且正在设置任务。

- INITIALIZED

  适用于 Amazon Sidewalk 的 AWS IoT Core 已完成导入任务设置，正在等待控制日志到达，以便它可以使用设备的序列号（SMSN）导入设备并继续处理任务。

- PENDING

  导入任务正在队列中等待处理。适用于 Amazon Sidewalk 的 AWS IoT Core 正在评估处理队列中的其他任务。

- COMPLETE

  导入任务已处理并已完成。

- FAILED

  导入任务或设备任务失败。您可以使用 `StatusReason` 参数来确定导入任务失败的原因，例如验证异常。

- DELETING

  导入任务已标记为待删除，正在删除中。

设备登记状态

您添加到导入任务中的 Sidewalk 设备可能具有以下状态消息之一。状态表明您的设备是已准备好供登记、已登记还是无法登记。还可以使用 AWS IoT 控制台或 AWS IoT Wireless API 操作

ListDevicesForWirelessDeviceImportTask 的 OnboardingStatusReason 参数来检索其他状态详细信息。

- INITIALIZED

  适用于 Amazon Sidewalk 的 AWS IoT Core 已完成导入任务设置，正在等待控制日志到达，以便它可以使用设备的序列号（SMSN）导入设备并继续处理任务。

- PENDING

  导入任务正在队列中等待处理并开始将设备登记到该任务。适用于 Amazon Sidewalk 的 AWS IoT Core 正在评估处理队列中的其他任务。

- ONBOARDED

  Sidewalk 设备已成功登记到导入任务。

- FAILED

  导入任务或设备任务失败，Sidewalk 设备未能登记到任务。您可以使用 OnboardingStatusReason 参数来检索有关设备登记失败原因的更多详细信息。

AWS IoT Wireless 中的安全性

AWS 十分重视云安全性。作为 AWS 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是 AWS 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云的安全性 - AWS 负责保护在 AWS 云中运行 AWS 服务的基础设施。AWS 还向您提供可安全使用的服务。第三方审核员定期测试和验证我们的安全性的有效性，作为 AWS 合规性计划的一部分。要了解适用于 AWS IoT Wireless 的合规性计划，请参阅按合规性计划提供的范围内 AWS 服务。
- 云中的安全性：您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档将帮助您了解如何在使用 AWS IoT Wireless 时应用责任共担模式。它说明了如何配置 AWS IoT Wireless 以实现您的安全性与合规性目标。您还将了解如何使用其他 AWS 服务来帮助您监控和保护 AWS IoT Wireless 资源。

内容

AWS IoT Wireless 中的数据保护

AWS 责任共担模式适用于 AWS IoT Wireless 中的数据保护。如该模式中所述，AWS 负责保护运行所有 AWS Cloud 的全球基础设施。您负责维护对托管在此基础架构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题解答。有关欧洲数据保护的信息，请参阅 AWS 安全性博客上的 AWS 责任共担模式和 GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management（IAM）设置单个用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 使用 SSL/TLS 与AWS资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用AWS CloudTrail设置 API 和用户活动日志记录。
- 使用 AWS 加密解决方案以及 AWS 服务中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果在通过命令行界面或 API 访问AWS时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括通过控制台、API、AWS CLI 或 AWS SDK 使用 AWS IoT Wireless 或其他 AWS 服务时。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

AWS IoT Wireless 中的数据加密

默认情况下，所有动态和静态 AWS IoT Wireless 数据均经过加密。AWS IoT Wireless 不支持来自 AWS KMS key 的客户托管 AWS KMS 密钥。要加密数据，AWS IoT Wireless 仅使用 AWS 拥有的密钥。

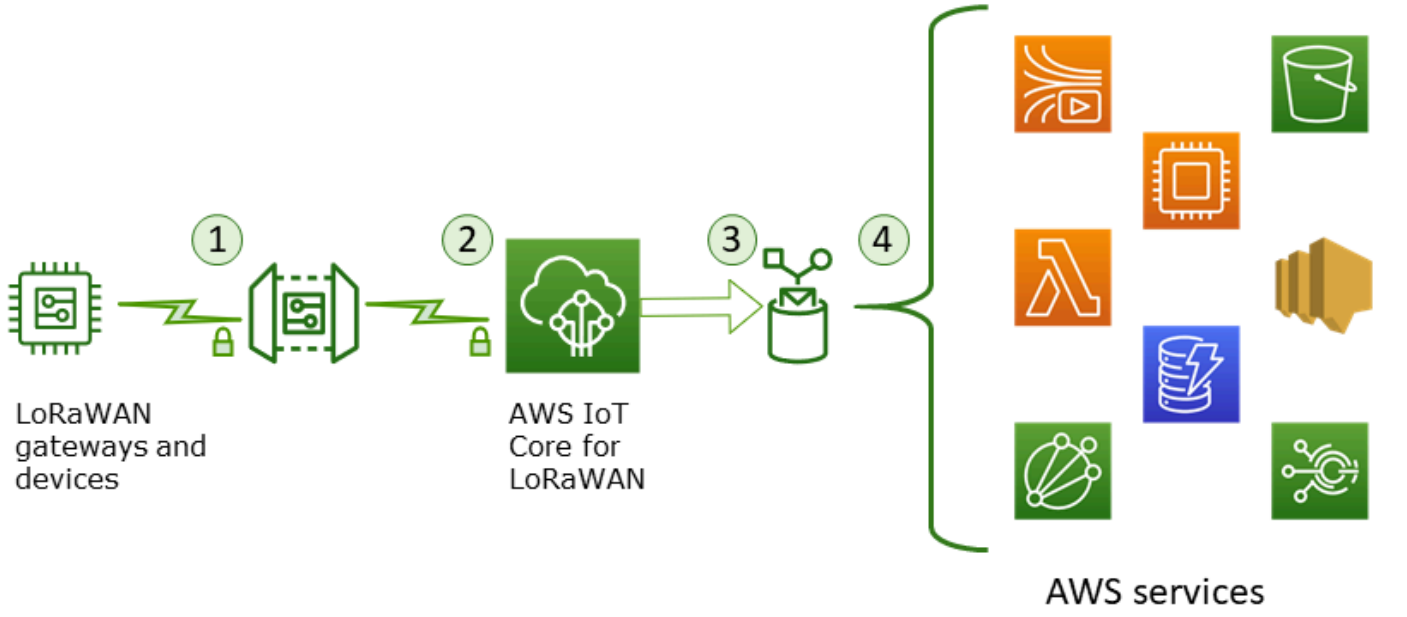
使用 适用于 LoRaWAN 的 AWS IoT Core 确保数据和传输安全

适用于 LoRaWAN 的 AWS IoT Core 使用以下方法保护 LoRaWAN 设备、网关和 适用于 LoRaWAN 的 AWS IoT Core 之间的数据和通信：

- 设备在与 LoRaWAN 网关通信时应遵循的最佳安全实践，如白皮书 LoRaWAN 安全性中所述。
- AWS IoT Core 用于将网关连接到 适用于 LoRaWAN 的 AWS IoT Core 并将数据发送到其他 AWS 服务的安全解决方案。有关更多信息，请参阅 AWS IoT Core 中的数据保护。

如何在整个系统中保护数据

此图表标识了 LoRaWAN 系统中的关键元素，这些元素连接到 适用于 LoRaWAN 的 AWS IoT Core 以确定在整个过程中保护数据的方式。



1. LoRaWAN 无线设备在传输二进制消息之前会使用 AES128 CTR 模式对其进行加密。
2. 网关与 适用于 LoRaWAN 的 AWS IoT Core 的连接通过 TLS 得到保护，如AWS IoT 中的传输安全性中所述。适用于 LoRaWAN 的 AWS IoT Core 对二进制消息进行解密，并将解密的二进制消息负载编码为 base64 字符串。
3. 生成的 base64 编码消息将作为消息负载发送到分配给设备的目标中所述的 AWS IoT 规则中。AWS 内的数据使用 AWS 拥有的密钥进行加密。
4. AWS IoT 规则将消息数据定向到规则配置中描述的服务。AWS 内的数据使用 AWS 拥有的密钥进行加密。

LoRaWAN 设备和网关传输安全性

LoRaWAN 设备和 适用于 LoRaWAN 的 AWS IoT Core 存储预共享的根密钥。会话密钥由 LoRaWAN 设备和 适用于 LoRaWAN 的 AWS IoT Core 根据这些协议衍生得出。对称会话密钥用于标准 AES-128 CTR 模式下的加密和解密。还使用 4 字节的消息完整性代码 (MIC) 来检查遵循标准 AES-128 CMAC 算法的数据完整性。可以使用加入/重新加入进程更新会话密钥。

LoRa 网关的安全实践在 LoRaWAN 规范中有所说明。LoRa 网关使用 `Basics Station` 通过 Web 套接字连接到 适用于 LoRaWAN 的 AWS IoT Core。适用于 LoRaWAN 的 AWS IoT Core 仅支持 `Basics Station` 2.0.4 及更高版本。

在建立 Web 套接字连接之前，适用于 LoRaWAN 的 AWS IoT Core 使用 TLS 服务器和客户端身份验证模式对网关进行身份验证。为了确保 LoRaWAN 协议的机密性，使用了 TLS 版本 1.2。TLS 支持适

用于许多编程语言和操作系统。AWS内的数据是由特定的AWS服务加密的。有关其他 AWS 服务上数据加密的更多信息，请参阅该服务的安全文档。

适用于 LoRaWAN 的 AWS IoT Core 还维护一个配置和更新服务器（CUPS），该服务器配置和更新用于 TLS 身份验证的证书和密钥。

Identity and Access Management for AWS IoT Wireless

AWS Identity and Access Management（IAM）是一项 AWS 服务，可以帮助管理员安全地控制对 AWS 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和获得授权（拥有权限），可以使用 AWS IoT Wireless 资源。IAM 是一项无需额外费用即可使用的 AWS 服务。

主题

受众

使用 AWS Identity and Access Management（IAM）的方式因您在 AWS IoT Wireless 中完成的工作而异。

服务用户 - 如果您使用 AWS IoT Wireless 服务来完成任务，则您的管理员会为您提供所需的凭证和权限。当您使用更多 AWS IoT Wireless 功能来完成工作时，您可能需要额外权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS IoT Wireless 中的功能，请参阅 排查 AWS IoT Wireless 身份和访问问题。

服务管理员 - 如果您在公司负责管理 AWS IoT Wireless 资源，则可能拥有对 AWS IoT Wireless 的完全访问权限。您有责任确定您的服务用户应访问哪些 AWS IoT Wireless 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 用于 AWS IoT Wireless 的更多信息，请参阅 如何将 AWS IoT Wireless 与 IAM 结合使用。

IAM 管理员 - 如果您是 IAM 管理员，您可能希望详细了解如何编写策略以管理对 AWS IoT Wireless 的访问。要查看您可以在 IAM 中使用的 AWS IoT Wireless 基于身份的策略示例，请参阅 基于身份的 AWS IoT Wireless 策略示例。

使用身份进行身份验证

身份验证是使用身份凭证登录AWS的方法。您必须作为AWS 账户根用户、IAM 用户或通过代入 IAM 角色进行身份验证（登录到AWS）。

您可以使用通过身份源提供的凭证以联合身份登录到AWS。AWS IAM Identity Center（IAM Identity Center）用户、您公司的单点登录身份验证以及您的 Google 或 Facebook 凭证都是联合身份的示例。当以联合身份登录时，管理员以前使用 IAM 角色设置了身份联合验证。当使用联合身份验证访问AWS时，就是在间接代入角色。

根据用户类型，可以登录AWS Management Console或AWS访问门户。有关登录到 AWS 的更多信息，请参阅 AWS 登录 用户指南 中的 如何登录到您的 AWS 账户。

如果以编程方式访问AWS，则AWS将提供软件开发工具包（SDK）和命令行界面 (CLI)，以便使用凭证以加密方式签署请求。如果不使用AWS工具，则必须自行对请求签名。有关使用推荐的方法自行签署请求的更多信息，请参阅《IAM 用户指南》中的签署 AWS API 请求。

无论使用何种身份验证方法，您都可能需要提供其他安全信息。例如，AWS 建议您使用多重身份验证（MFA）来提高账户的安全性。要了解更多信息，请参阅 AWS IAM Identity Center 用户指南 中的 多重身份验证 和 IAM 用户指南 中的 在 AWS 中使用多重身份验证（MFA）。

AWS 账户 根用户

创建AWS 账户时，最初使用的是一个对账户中所有AWS 服务和资源拥有完全访问权限的登录身份。此身份称为AWS 账户根用户，使用创建账户时所用的电子邮件地址和密码登录，即可获得该身份。强烈建议不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关需要您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南 中的 需要根用户凭证的任务。

IAM 用户和组

IAM 用户 是 AWS 账户内对某个人员或应用程序具有特定权限的一个身份。在可能的情况下，建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果有一些特定的使用场景需要长期凭证以及 IAM 用户，我们建议轮换访问密钥。有关更多信息，请参阅 IAM 用户指南 中的 对于需要长期凭证的使用场景定期轮换访问密钥。

IAM 组 是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。可以使用群组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，可能具有一个名为 IAMAdmins 的群组，并为该群组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人担任。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的何时创建 IAM 用户（而不是角色）。

IAM 角色

> **Note**
>
> AWS IoT Wireless 不支持服务角色和服务相关角色。

IAM 角色是 AWS 账户 中具有特定权限的身份。它类似于 IAM 用户，但与特定人员不关联。可以通过切换角色，在AWS Management Console中暂时代入 IAM 角色。您可以调用 AWS CLI 或 AWS API 操作或使用自定义网址以担任角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 - 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅 IAM 用户指南 中的 为第三方身份提供商创建角色。如果使用 IAM Identity Center，则需要配置权限集。为控制身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅 AWS IAM Identity Center 用户指南 中的 权限集。
- 临时 IAM 用户权限 - IAM 用户或角色可担任 IAM 角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 - 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户存取权限的主要方式。但是，对于某些AWS 服务，可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅 IAM 用户指南 中的 IAM 角色与基于资源的策略有何不同。
- 跨服务访问 – 某些 AWS 服务使用其他 AWS 服务中的特征。例如，在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service（Amazon S3）中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话：当您使用 IAM 用户或角色在 AWS 中执行操作时，您将被视为主体。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用主体调用

AWS 服务 的权限，结合请求的 AWS 服务，向下游服务发出请求。只有在服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

- 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅 IAM 用户指南 中的 创建向 AWS 服务委派权限的角色。
- 服务相关角色 – 服务相关角色是与AWS 服务关联的一种服务角色。服务可以担任代表您执行操作的角色。服务相关角色显示在AWS 账户中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

- 在 Amazon EC2 上运行的应用程序 – 可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时凭证。这优先于在 EC2 实例中存储访问密钥。要将AWS角色分配给 EC2 实例并使其对该实例的所有应用程序可用，可以创建一个附加到实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅 IAM 用户指南 中的 使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅《IAM 用户指南》中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

将创建策略并将其附加到AWS身份或资源，以控制AWS中的访问。策略是AWS中的对象；在与身份或资源相关联时，策略定义它们的权限。在主体（用户、根用户或角色会话）发出请求时，AWS将评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略在AWS中存储为 JSON 文档。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概览。

管理员可以使用AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件 下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。然后，管理员可以向角色添加 IAM 策略，并且用户可以代入角色。

IAM 策略定义操作的权限，无关乎使用哪种方法执行操作。例如，假设有一个允许 `iam:GetRole` 操作的策略。具有该策略的用户可以从AWS Management Console、AWS CLI或AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管式策略是可以附加到AWS 账户中的多个用户、组和角色的独立策略。托管式策略包括AWS托管式策略和客户托管式策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管式策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合用户或AWS 服务。

基于资源的策略是位于该服务中的内联策略。不能在基于资源的策略中使用来自 IAM 的AWS托管式策略。

访问控制列表（ACL）

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3、AWS WAF和 Amazon VPC 是支持 ACL 的服务示例。要了解有关 ACL 的更多信息，请参阅 Amazon Simple Storage Service 开发人员指南 中的访问控制列表（ACL）概览。

其他策略类型

AWS支持额外的、不太常用的策略类型。这些策略类型可以设置更常用的策略类型所授予的最大权限。

- 权限边界 – 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 字段中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的 IAM 实体的权限边界。

- 服务控制策略（SCP）– SCP 是 JSON 策略，指定了组织或组织单位（OU）在 AWS Organizations 中的最大权限。AWS Organizations 服务可以分组和集中管理您的企业拥有的多个 AWS 账户 账户。如果在组织内启用了所有特征，则可对任意或全部账户应用服务控制策略（SCP）。SCP 限制成员账户中实体（包括每个AWS 账户根用户）的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的 SCP 的工作原理。
- 会话策略 - 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解 AWS 如何确定在涉及多种策略类型时是否允许请求，请参阅《IAM 用户指南》中的策略评估逻辑。

如何将 AWS IoT Wireless 与 IAM 结合使用

在使用 IAM 管理对 AWS IoT Wireless 的访问之前，您应了解哪些 IAM 功能可与 AWS IoT Wireless 结合使用。要大致了解 AWS IoT Wireless 和其他 AWS 服务如何与 IAM 一起使用，请参阅《IAM 用户指南》中的使用 IAM 的 AWS 服务。

可以与 AWS IoT Wireless 搭配使用的 IAM 功能

IAM 功能	AWS IoT Wireless 支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键	是
ACL	否
ABAC（策略中的标签）	是
临时凭证	是

IAM 功能	AWS IoT Wireless 支持
主体权限	是
服务角色	否
服务相关角色	否

主题

- AWS IoT Wireless 基于身份的策略
- AWS IoT Wireless 内基于资源的策略
- 策略操作
- 策略资源
- 条件键
- 访问控制列表（ACL）
- 带有 AWS IoT Wireless 的 ABAC
- 将临时凭证用于 AWS IoT Wireless
- AWS IoT Wireless 的跨服务主体权限
- 服务角色
- AWS IoT Wireless 的服务相关角色

AWS IoT Wireless 基于身份的策略

支持基于身份的策略	是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

示例

要查看 AWS IoT Wireless 基于身份的策略示例，请参阅 基于身份的 AWS IoT Wireless 策略示例。

AWS IoT Wireless 内基于资源的策略

支持基于资源的策略	否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合用户或AWS 服务。

要启用跨账户存取，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当主体和资源处于不同的 AWS 账户 中时，则信任账户中的 IAM 管理员还必须授予主体实体（用户或角色）对资源的访问权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅IAM 用户指南中的 IAM 角色与基于资源的策略有何不同。

策略操作

支持策略操作	是

管理员可以使用AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件 下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

AWS IoT Wireless 中的策略操作在操作前使用以下前缀：`iotwireless:`例如，要授予某人权限以使用 `ListWirelessDevices` API 列出其 AWS 账户中已注册的所有无线设备，请将

iotwireless:ListWirelessDevices 操作纳入其策略中。策略语句必须包含 Action 或 NotAction 元素。AWS IoT Wireless 定义了一组自己的操作，以描述您可以使用该服务执行的任务。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

```
"Action": [
      "iotwireless:ListMulticastGroups",
      "iotwireless:ListFuotaTasks"
   ]
```

您也可以使用通配符（*）指定多个操作。例如，要指定以单词 Get 开头的所有操作，包括以下操作：

```
"Action": "iotwireless:Get*"
```

要查看 AWS IoT Wireless 操作列表，请参阅《IAM 用户指南》中的 AWS IoT Wireless 定义的操作。

策略资源

支持策略资源	是

管理员可以使用AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件 下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实操，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

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

该 AWS IoT Wireless 服务拥有以下 ARN：

```
arn:${Partition}:iotwireless:${Region}:${Account}:${Resource}/${Resource-id}
```

有关 ARN 格式的更多信息，请参阅 Amazon 资源名称（ARN）和 AWS 服务命名空间。

例如，要在语句中指定网络分析器配置 NAConfig1，请使用以下 ARN：

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:NetworkAnalyzerConfiguration/
NAConfig1"
```

要指定属于特定账户的所有 FUOTA 任务，请使用通配符 (*)：

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:FuotaTask/*"
```

无法对特定资源执行某些 AWS IoT Wireless 操作，例如，用于列出资源的操作。在这些情况下，您必须使用通配符（*）。

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

许多 AWS IoT Wireless API 操作涉及多种资源。例如，AssociateWirelessDeviceWithThing 将无线设备与 AWS IoT 事物关联，因此 IAM 用户必须有权使用该设备和物联网事物。要在单个语句中指定多个资源，请使用逗号分隔 ARN。

```
"Resource": [
      "WirelessDevice",
      "thing"
```

要查看 AWS IoT Wireless 资源类型及其 ARN 的列表，请参阅《IAM 用户指南》中的 AWS IoT Wireless 定义的资源。要了解您可以在哪些操作中指定每个资源的 ARN，请参阅 AWS IoT Wireless 定义的操作。

条件键

支持特定于服务的策略条件键	是

管理员可以使用AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件 下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您要为单个条件键指定多个值，则 AWS 使用逻辑 OR 运算来评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM 策略元素：变量和标签。

AWS 支持全局条件键和特定于服务的条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南 中的 AWS 全局条件上下文键。

AWS IoT Wireless 定义了自己的一组条件键，还支持使用一些全局条件键。要查看所有 AWS 全局条件键，请参阅《IAM 用户指南》中的 AWS 全局条件上下文键。要查看 AWS IoT Wireless 条件键的列表，请参阅《IAM 用户指南》中的 AWS IoT Wireless 条件键。要了解您可以对哪些操作和资源使用条件键，请参阅 AWS IoT Wireless 定义的操作。

访问控制列表（ACL）

支持 ACL	否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

带有 AWS IoT Wireless 的 ABAC

支持 ABAC（策略中的标签）	是

基于属性的访问权限控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在 AWS 中，这些属性称为标签。您可以将标签附加到 IAM 实体（用户或角色）以及 AWS 资源。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，需要使用 aws:ResourceTag/*key-name*、aws:RequestTag/*key-name* 或 aws:TagKeys 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的什么是 ABAC？。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

您可以将标签附加到 AWS IoT Wireless 资源或将请求中的标签传递到 AWS IoT Wireless。要基于标签控制访问，需要使用 `YOUR-SERVICE-PREFIX:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。有关为 AWS IoT Wireless 资源添加标签的更多信息，请参阅 标记您的 AWS IoT Wireless 资源。

将临时凭证用于 AWS IoT Wireless

支持临时凭证	是

某些 AWS 服务 在使用临时凭证登录时无法正常工作。有关更多信息，包括 AWS 服务与临时凭证配合使用，请参阅《IAM 用户指南》中的使用 IAM 的 AWS 服务。

如果您不使用用户名和密码而用其他方法登录到AWS Management Console，则使用临时凭证。例如，当您使用贵公司的单点登录 (SSO) 链接访问AWS时，该过程将自动创建临时凭证。当您以用户身份登录控制台，然后切换角色时，还会自动创建临时凭证。有关切换角色的更多信息，请参阅 IAM 用户指南 中的 切换到角色（控制台）。

您可以使用 AWS CLI 或者 AWS API 创建临时凭证。之后，您可以使用这些临时凭证访问 AWS。AWS 建议您动态生成临时凭证，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

AWS IoT Wireless 的跨服务主体权限

支持转发访问会话 (FAS)	是

当您使用 IAM 用户或角色在 AWS 中执行操作时，您将被视为主体。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用主体调用 AWS 服务 的权限，结合请求的 AWS 服务，向下游服务发出请求。只有在服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

服务角色

支持服务角色	否

服务角色是由一项服务代入、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅 IAM 用户指南 中的 创建向 AWS 服务 委派权限的角色。

AWS IoT Wireless 的服务相关角色

支持服务相关角色	否

服务相关角色是一种与 AWS 服务相关的服务角色。服务可以担任代表您执行操作的角色。服务相关角色显示在AWS 账户中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

基于身份的 AWS IoT Wireless 策略示例

默认情况下，IAM 用户和角色没有创建或修改 AWS IoT Wireless 资源的权限。它们还无法使用 AWS Management Console、AWS CLI 或 AWS API 执行任务。IAM 管理员必须创建 IAM 策略，以便为用户和角色授予权限以对所需的指定资源执行特定的 API 操作。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户或组。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅《IAM 用户指南》中的 在 JSON 选项卡上创建策略。

主题

- 策略最佳实践
- 使用 AWS IoT Wireless 控制台
- 允许用户查看他们自己的权限
- 执行 AWS IoT Wireless 无线设备操作所需的权限

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 AWS IoT Wireless 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- AWS 托管式策略及转向最低权限许可入门 - 要开始向用户和工作负载授予权限，请使用 AWS 托管式策略来为许多常见使用场景授予权限。可以在AWS 账户中找到这些策略。我们建议通过定义特定于您的使用场景的AWS客户托管式策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略或工作职能的 AWS 托管式策略。
- 应用最低权限 - 在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限 - 您可以向策略添加条件来限制对操作和资源的访问。例如，可以编写策略条件来指定必须使用 SSL 发送所有请求。如果通过特定AWS 服务（例如AWS CloudFormation）使用服务操作，还可以使用条件来授予对服务操作的访问权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM Access Analyzer 策略验证。
- 需要多重身份验证 (MFA) - 如果您所处的场景要求您的 AWS 账户 中有 IAM 用户或根用户，请启用 MFA 来提高安全性。要在调用 API 操作时需要 MFA，请将 MFA 条件添加到策略中。有关更多信息，请参阅《IAM 用户指南》中的配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实践的更多信息，请参阅 IAM 用户指南中的 IAM 中的安全最佳实践。

使用 AWS IoT Wireless 控制台

要访问 AWS IoT Wireless 控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您的 AWS 账户中 AWS IoT Wireless 资源的详细信息。如果您创建的基于身份的策略比所需的最低权限更严格，则无法为具有该策略的实体（IAM 用户或角色）正常运行控制台。

要确保这些实体仍可使用 AWS IoT Wireless 控制台，也可向实体附加以下 AWS 托管策略。有关更多信息，请参阅 IAM 用户指南中的为用户添加权限：

```
AWSIoTWirelessFullAccess
```

对于只需要调用 AWS CLI 或 AWS API 的用户，无需为其提供最低控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联策略和托管式策略。此策略包括在控制台上完成此操作或者以编程方式使用 AWS CLI 或 AWS API 所需的权限。

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

执行 AWS IoT Wireless 无线设备操作所需的权限

您可以在基于身份的策略中使用条件，以控制对 AWS IoT Wireless 操作的访问。此示例演示如何创建允许创建和管理设备的策略。不过，只有在事物标签 `Owner` 具有该用户的用户名值时，才会授予权限。此策略还授予在控制台上完成此操作的必要权限。

```
{
 "Version": "2012-10-17",
 "Statement": [{
    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
              "iotwireless:CreateWirelessDevice",
              "iotwireless:GetWirelessDevice",
              "iotwireless:ListWirelessDevices",
              "iotwireless:UpdateWirelessDevice",
              "iotwireless:DeleteWirelessDevice"
          ],
    "Resource": "*"
    }
 ]
}
```

该策略有一条语句授予执行 `CreateWirelessDevice`、`GetWirelessDevice`、`ListWirelessDevices`、`UpdateWirelessDevic` 和 `DeleteWirelessDevice` 操作的权限。AWS IoT Wireless 调用这些方法来创建和管理您的无线设备。

该策略不指定主体元素，因为在基于身份的策略中，您未指定获取权限的主体。将策略附加到用户时，该用户是隐式主体。向 IAM 角色附加权限策略后，该角色的信任策略中标识的主体将获取权限。

适用于 AWS IoT Wireless 的 AWS 托管式策略

要向用户、组和角色添加权限，与自己编写策略相比，使用 AWS 托管策略更简单。创建仅为团队提供所需权限的 IAM 客户管理型策略需要时间和专业知识。要快速入门，您可以使用我们的 AWS 托管策略。这些策略涵盖常见使用案例，可在您的 AWS 账户 中使用。有关 AWS 托管策略的更多信息，请参阅《IAM 用户指南》中的AWS 托管策略。

AWS 服务负责维护和更新 AWS 托管式策略。您无法更改 AWS 托管式策略中的权限。服务偶尔会向 AWS 托管策略添加额外权限以支持新特征。此类更新会影响附加策略的所有身份（用户、组和角色）。当启动新特征或新操作可用时，服务最有可能会更新 AWS 托管策略。服务不会从 AWS 托管策略中删除权限，因此策略更新不会破坏您的现有权限。

此外，AWS 还支持跨多种服务的工作职能的托管策略。例如，ReadOnlyAccess AWS 托管式策略提供对所有 AWS 服务和资源的只读访问权限。当服务启动新特征时，AWS 会为新操作和资源添加只读权限。有关工作职能策略的列表和说明，请参阅《IAM 用户指南》中的适用于工作职能的 AWS 托管策略。

AWS 托管策略：AWSIoTWirelessDataAccess

您可以将 `AWSIoTWirelessDataAccess` 策略附加到 IAM 身份。

此策略向相关身份授予权限，以允许使用 `SendDataToWirelessDevice` API 向 LoRaWAN 和 Sidewalk 设备发送数据。要在 AWS Management Console中查看此策略，请参阅 AWSIoTWirelessDataAccess。

权限详细信息

该策略包含以下权限。

- `iotwireless` – 检索 AWS IoT Wireless 数据。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iotwireless:SendDataToWirelessDevice"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

AWS 托管策略：AWSIoTWirelessFullAccess

您可以将 AWSIoTWirelessFullAccess 策略附加到 IAM 身份。

此策略向相关身份授予权限，以允许访问所有 AWS IoT Wireless 操作。要在 AWS Management Console中查看此策略，请参阅 AWSIoTWirelessFullAccess。

权限详细信息

该策略包含以下权限。

- iotwireless – 检索 AWS IoT Wireless 数据并执行所有 AWS IoT Wireless 操作。

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

AWS 托管策略：AWSIoTWirelessFullPublishAccess

您可以将 AWSIoTWirelessFullPublishAccess 策略附加到 IAM 身份。

此策略向相关身份授予权限，以允许进行受限访问，代表您发布 AWS IoT 规则。要在 AWS Management Console中查看该策略，请参阅 AWSIoTWirelessFullPublishAccess。

权限详细信息

该策略包含以下权限。

- iot - 执行获取端点 URL 并发布到 AWS IoT 规则引擎的操作。

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

AWS 托管策略：AWSIoTWirelessLogging

您可以将 AWSIoTWirelessLogging 策略附加到 IAM 身份。

此策略向相关身份授予权限，以允许创建 Amazon CloudWatch Logs 日志组并将日志流式传输到这些组。本策略附加到您的 CloudWatch 日志记录角色。要在 AWS Management Console中查看该策略，请参阅 AWSIoTWirelessLogging。

权限详细信息

该策略包含以下权限。

- `logs` 检索 CloudWatch 日志。此外，允许创建 CloudWatch Logs 组和将日志流式传输到这些组。

```
{
    "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 托管策略：AWSIoTWirelessReadOnlyAccess

您可以将 `AWSIoTLogging` 策略附加到 IAM 身份。

此策略向相关身份授予权限，以允许对所有 AWS IoT Wireless 操作进行只读访问。要在 AWS Management Console中查看该策略，请参阅 `AWSIoTWirelessReadOnlyAccess`。

权限详细信息

该策略包含以下权限。

- `logs` - 执行 AWS IoT Wireless `List` 和 `Get` API 操作。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iotwireless:List*",
                "iotwireless:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 托管策略：AWSIoTWirelessGatewayCertManager

您可以将 AWSIoTWirelessGatewayCertManager 策略附加到 IAM 身份。

此策略向相关身份授予权限，以允许创建、列出和描述 AWS IoT 证书。要在 AWS Management Console中查看该策略，请参阅 AWSIoTWirelessGatewayCertManager。

权限详细信息

该策略包含以下权限。

- iot - 执行创建、描述和列出证书的操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IoTWirelessGatewayCertManager",
            "Effect": "Allow",
            "Action": [
                "iot:CreateKeysAndCertificate",
                "iot:DescribeCertificate",
                "iot:ListCertificates"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

AWS IoT Wireless；对 AWS 托管式策略的更新

查看有关 AWS IoT Wireless 的 AWS 托管式策略更新的详细信息（从该服务开始跟踪这些更改开始）。有关此页面更改的自动提示，请订阅AWS IoT Wireless 文档历史记录页面上的 RSS 信息源。

更改	描述	日期
AWS IoT Wireless 开启了跟踪更改	AWS IoT Wireless 为其 AWS 托管式策略开启了跟踪更改。	2022 年 5 月 18 日

排查 AWS IoT Wireless 身份和访问问题

使用以下信息可帮助您诊断和修复在使用 AWS IoT Wireless 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 AWS IoT Wireless 中执行操作
- 我想要查看我的访问密钥
- 我是管理员并希望允许其他人访问 AWS IoT Wireless
- 我想要允许我的 AWS 账户之外的用户访问我的 AWS IoT Wireless 资源

我无权在 AWS IoT Wireless 中执行操作

如果AWS Management Console告诉您，无权执行某个操作，则必须联系管理员寻求帮助。管理员是指提供用户名和密码的人员。

当 `mateojackson` IAM 用户尝试使用控制台查看有关 *`WirelessDevice`* 的详细信息，但不具有 `YOUR-SERVICE-PREFIX:`*`GetWirelessDevice`* 权限时，会出现以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: YOUR-SERVICE-PREFIX:GetWirelessDevice on resource: my-LoRaWAN-device
```

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他使用 YOUR-SERVICE-PREFIX:*GetWirelessDevice* 操作访问 *my-LoRaWAN-device* 资源。

我想要查看我的访问密钥

在创建 IAM 用户访问密钥后，您可以随时查看您的访问密钥 ID。但是，您无法再查看您的秘密访问密钥。如果您丢失了私有密钥，则必须创建一个新的访问密钥对。

访问密钥包含两部分：访问密钥 ID（例如 AKIAIOSFODNN7EXAMPLE）和秘密访问密钥（例如 wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY）。与用户名和密码一样，您必须同时使用访问密钥 ID 和秘密访问密钥对请求执行身份验证。像对用户名和密码一样，安全地管理访问密钥。

> **Important**
>
> 请不要向第三方提供访问密钥，即便是为了帮助找到您的规范用户 ID 也不行。如果您这样做，可能会向某人提供对您的 AWS 账户 账户的永久访问权限。

当您创建访问密钥对时，系统会提示您将访问密钥 ID 和秘密访问密钥保存在一个安全位置。秘密访问密钥仅在您创建它时可用。如果丢失了您的秘密访问密钥，您必须为 IAM 用户添加新的访问密钥。您最多可拥有两个访问密钥。如果您已有两个密钥，则必须删除一个密钥对，然后再创建新的密钥。要查看说明，请参阅 IAM 用户指南中的管理访问密钥。

我是管理员并希望允许其他人访问 AWS IoT Wireless

要允许其他人访问 AWS IoT Wireless，您必须为需要访问权限的人员或应用程序创建一个 IAM 实体（用户或角色）。它们将使用该实体的凭证访问 AWS。然后，您必须将策略附加到实体，以便在 AWS IoT Wireless 中向其授予正确的权限。

要立即开始使用，请参阅《IAM 用户指南》中的创建您的第一个 IAM 委派用户和组。

我想要允许我的 AWS 账户之外的用户访问我的 AWS IoT Wireless 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 AWS IoT Wireless 是否支持这些功能，请参阅 如何将 AWS IoT Wireless 与 IAM 结合使用。

- 要了解如何为您拥有的 AWS 账户 中的资源提供访问权限，请参阅 IAM 用户指南 中的为您拥有的另一个 AWS 账户 中的 IAM 用户提供访问权限。
- 要了解如何为第三方 AWS 账户提供您资源的访问权限，请参阅《IAM 用户指南》中的为第三方拥有的 AWS 账户提供访问权限。
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

AWS IoT Wireless 合规性验证

作为多个 AWS 合规性计划的一部分，第三方审计员将评测 AWS IoT Wireless 的安全性与合规性。其中包括 SOC、PCI、FedRAMP、HIPAA 及其他。

有关特定合规性计划范围内的 AWS 服务列表，请参阅合规性计划范围内的 AWS 服务。有关常规信息，请参阅 AWS 合规性计划。

您可以使用 AWS Artifact 下载第三方审核报告。有关更多信息，请参阅在 AWS Artifact 中下载报告。

您使用 AWS IoT Wireless 时的合规性责任取决于您数据的敏感度、您公司的合规性目标以及适用的法律法规。AWS 提供以下资源来帮助实现合规性：

- 安全性与合规性 Quick Start 指南 - 这些部署指南讨论了架构注意事项，并提供了在 AWS 上部署基于安全性和合规性的基准环境的步骤。
- 《设计符合 HIPAA 安全性和合规性要求的架构》白皮书 – 此白皮书介绍公司如何使用AWS创建符合 HIPAA 标准的应用程序。
- AWS合规性资源 – 此业务手册和指南集合可能适用于您的行业和位置。
- 《AWS Config 开发人员指南》中的使用规则评估资源 – AWS Config；评测您的资源配置对内部实践、行业指南和法规的遵循情况。
- AWS Security Hub – 此AWS服务提供了AWS中安全状态的全面视图，可帮助您检查是否符合安全行业标准和最佳实操。

AWS IoT Wireless 中的恢复能力

AWS 全球基础设施围绕 AWS 区域和可用区构建。区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区

之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础架构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅 AWS全球基础设施。

AWS IoT Wireless 中的基础设施安全性

作为一项托管服务，AWS IoT Wireless 由 Amazon Web Services：安全流程概述白皮书中所述的 AWS 全球网络安全程序提供保护。

您可以使用 AWS 发布的 API 调用通过网络访问 AWS IoT Wireless。客户端必须支持传输层安全性 (TLS) 1.0 或更高版本。建议使用 TLS 1.2 或更高版本。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman (ECDHE)。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

使用 Amazon CloudWatch Logs 监控 AWS IoT Wireless 资源

监控是保持 AWS IoT Wireless 和您的其他 AWS 解决方案的可靠性、可用性和性能的重要环节。您可以对 LoRaWAN 和 Sidewalk 设备使用监控功能，并在它们登记到 AWS IoT Wireless 时从中获取有用信息和错误。

我们强烈建议您从 AWS 解决方案的各个部分收集监控数据，以便更轻松地调试出现的多点故障。首先创建一个监控计划来回答以下问题。如果您不确定如何回答这些问题，您仍然可以继续启用日志记录并建立性能基准。

- 监控目的是什么？
- 您将监控哪些资源？
- 监控这些资源的频率如何？
- 您将使用哪些监控工具？
- 谁负责执行监控任务？
- 出现错误时应通知谁？

下一步是启用日志记录，并通过在不同时间和不同负载条件下测量性能，在您的环境中建立常规 AWS IoT Wireless 性能基准。监控 AWS IoT Wireless 时，请保留历史监控数据，以便将其与当前性能数据进行比较。这将帮助您确定一般的性能模式和性能异常，并设计解决问题的方法。

监控工具

您可以使用以下监控工具来监控 AWS IoT Wireless、在出现错误时进行报告并在适当的时候采取自动措施：

- Amazon CloudWatch 可实时监控您的 AWS 资源以及您在 AWS 上实时运行的应用程序。您可以收集和跟踪指标，创建自定义的控制平面，以及设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。例如，您可以使用 CloudWatch 跟踪 Amazon EC2 实例的 CPU 使用率或其他指标并且在需要时自动启动新实例。有关更多信息，请参阅《Amazon CloudWatch 用户指南》。
- 您可以使用网络分析器监控 LoRaWAN 资源（包括 LoRaWAN 设备和网关），从而缩短建立连接来开始接收跟踪消息所需的时间，为您提供即时日志信息。有关更多信息，请参阅使用网络分析器实时监控无线资源机群。

如何使用 Amazon CloudWatch 监控资源

您可以使用 CloudWatch 监控 AWS IoT Wireless。CloudWatch 会收集原始数据并将其处理为易读且近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅《Amazon CloudWatch 用户指南》。

要记录和监控 AWS IoT Wireless 资源，请执行以下步骤：

1. 创建日志记录角色来记录 AWS IoT Wireless 资源，如 为 AWS IoT Wireless 创建日志记录角色和策略 中所述。
2. CloudWatch Logs 控制台中日志消息的默认日志级别为 ERROR，这些内容较为简略，仅包含错误信息。如果您要查看更详细的消息，我们建议您先使用 CLI 配置日志记录，如 为 AWS IoT Wireless 资源配置日志记录 中所述。
3. 接下来，您可以通过查看 CloudWatch Logs 控制台中的日志条目来监控资源。有关更多信息，请参阅查看 CloudWatch AWS IoT Wireless 日志条目。
4. 您可以使用 Logs groups（日志组）来创建筛选条件表达式，但我们建议您首先创建简单的筛选条件，并查看日志组中的日志条目，然后转到 CloudWatch Insights 创建查询，以根据您所监控的资源或事件筛选日志条目。有关更多信息，请参阅使用 CloudWatch Insights 为 AWS IoT Wireless 筛选日志。

为 AWS IoT Wireless 配置日志记录

在可以监控和日志记录 AWS IoT 活动前，首先使用 CLI 或 API 启用 AWS IoT Wireless 资源的日志记录。

在考虑如何配置 AWS IoT Wireless 日志记录时，除非另有指定，否则原定设置日志记录配置将确定如何记录 AWS IoT 活动。首先，您可能要获取默认日志级别为 INFO 的详细日志。

查看初始日志后，您可以将默认日志级别更改为较低的详细程度级别 ERROR，并对可能需要更多关注的资源设置更详细的资源特定日志级别。日志级别可随时更改。

以下主题介绍了如何为 AWS IoT Wireless 资源配置日志记录。

主题

- 为 AWS IoT Wireless 创建日志记录角色和策略
- 为 AWS IoT Wireless 资源配置日志记录

为 AWS IoT Wireless 创建日志记录角色和策略

以下演示了如何仅为 AWS IoT Wireless 资源创建日志记录角色。如果您还想为 AWS IoT Core 创建日志记录角色，请参阅 https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html。

为 AWS IoT Wireless 创建日志记录角色

在可以启用日志记录之前，您必须创建一个 IAM 角色和一个策略，用于向 AWS 授予代表您监控 AWS IoT Wireless 活动的权限。

创建 IAM 角色以进行日志记录

要为 AWS IoT Wireless 创建日志记录角色，请打开 Roles hub of the IAM console（IAM 控制台的角色中心）并选择 Create role（创建角色）。

1. 在选择受信任实体的类型下，选择其他 AWS 账户。
2. 在 Account ID（账户 ID）中，输入您的AWS账户 ID，然后选择 Next: Permissions（下一步：权限）。
3. 在搜索框中，输入 **`AWSIoTWirelessLogging`**。
4. 选中策略旁名为 AWSIoTWirelessLogging 的复选框，然后选择 Next: Tags（下一步：标签）。
5. 选择 下一步: 审核。
6. 对于 Role name（角色名称），请输入 **`IoTWirelessLogsRole`**，然后选择 Create role（创建角色）。

编辑 IAM 角色的信任关系

在运行上一步后显示的确认信息中，选择您创建的角色的名称，IoTWirelessLogsRole。接下来，您将编辑角色以添加以下信任关系。

1. 在角色 IoTWirelessLogsRole 的 Summary（摘要）部分，选择 Trust relationships（信任关系）选项卡，然后选择 Edit trust relationships（编辑信任关系）。
2. 在 Policy Document（策略文档）中，更改 `Principal` 属性以使其类似于此示例。

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

在您更改 Principal 属性后，完整的策略文档应该如此示例所示。

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

3. 要保存更改并退出，请选择 Update Trust Policy（更新信任策略）。

AWS IoT Wireless 的日志记录策略

以下策略文档提供了角色策略和信任策略，让 AWS IoT Wireless 可代表您向 CloudWatch 提交日志条目。

> **Note**
>
> 该由AWS托管的策略文档是在您创建日志记录角色 IoTWirelessLogsRole 时自动创建的。

角色策略

以下显示了角色策略文档。

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

信任策略仅记录 AWS IoT Wireless 活动：

下面显示了仅记录 AWS IoT Wireless 活动的信任策略。

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

如果您创建了 IAM 角色以同时录入 AWS IoT Core 活动，则策略文档将允许您记录这两个活动。有关为 AWS IoT Core 创建的日志记录角色的信息，请参阅 https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html。

后续步骤

您已了解如何创建记录角色来录入您的 AWS IoT Wireless 资源。预设情况下，日志的日志级别为 ERROR，因此如果您只想查看错误信息，请转到 查看 CloudWatch AWS IoT Wireless 日志条目 通过查看日志条目来监控您的无线资源。

如果需要在日志条目中获得更多信息，可以为资源或不同事件类型配置默认日志级别，例如将日志级别设置为 INFO。有关为您的资源配置日志记录的信息，请参阅 为 AWS IoT Wireless 资源配置日志记录。

为 AWS IoT Wireless 资源配置日志记录

要为 AWS IoT Wireless 资源配置日志记录，您可以使用 API 或 CLI。开始监控 AWS IoT Wireless 资源时，您可以使用原定设置配置。若要执行该操作，您可以跳过此主题，然后继续执行 使用 CloudWatch Logs 监控 AWS IoT Wireless 来监控您的日志。

开始监控日志后，您可以使用 CLI 将日志级别更改为更详细的选项，例如提供 INFO 和 ERROR 信息并为更多资源启用日志记录。

AWS IoT Wireless 资源和日志级别

在使用 API 或 CLI 之前，请使用下表了解可以为其配置日志记录的不同日志级别和资源。该表显示了您在监控资源时能在 CloudWatch Logs 中看到的参数。如何为资源配置日志记录将决定您会在控制台中看到的日志。

有关 CloudWatch Logs 示例以及如何使用这些参数记录有关 AWS IoT Wireless 资源有关有用信息的内容，请参阅 查看 CloudWatch AWS IoT Wireless 日志条目。

日志级别和资源

名称	可能的值	描述
logLevel	INFO、ERROR 或 DISABLED	• ERROR：显示导致操作失败的任何错误。日志中仅包含 ERROR 信息。 • INFO：提供有关事物流的高级别信息。日志包括 INFO 和 ERROR 信息。 • DISABLED：禁用所有日志记录。
resource	WirelessGateway 或 WirelessDevice	资源类型，可以是 WirelessGateway 或 WirelessDevice 。
wirelessGatewayType	LoRaWAN	无线网关的类型，当 resource 为 WirelessGateway 时，始终为 LoRaWAN。
wirelessDeviceType	LoRaWAN 或 Sidewalk	无线设备的类型，当 resource 为 WirelessDevice 时，可以是 LoRaWAN 或者 Sidewalk。
wirelessGatewayId	-	无线网关的标识符，当 resource 为 WirelessGateway 时。

名称	可能的值	描述
wirelessDeviceId	-	无线设备的标识符，当 resource 为 WirelessDevice 时。
event	Join、Rejoin、Registration 、Uplink_data 、Downlink_data 、CUPS_Request 和 Certificate	记录的事件类型，取决于您记录的资源是无线设备还是无线网关。有关更多信息，请参阅查看 CloudWatch AWS IoT Wireless 日志条目。

AWS IoT Wireless 日志记录 API

您可以通过以下 API 操作来配置资源日志记录。该表还显示了您必须为使用 API 操作创建的示例 IAM 策略。以下部分介绍如何使用 API 配置资源的日志级别。

对 API 操作进行日志记录

API 名称	描述	IAM 策略示例
GetLogLevelsByResourceTypes	返回当前默认日志级别，或按资源类型返回日志级别，其中可以包括无线设备或无线网关的日志选项。	{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:GetLogLevelsByResourceTypes"], "Resource": ["*"

API 名称	描述	IAM 策略示例
		] }] }

GetResourceLogLevel	返回给定资源标识符和资源类型的日志级别覆盖。资源可以是无线设备或无线网关。	{
		"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 名称	描述	IAM 策略示例

PutResourceLogLevel	为给定资源标识符和资源类型设置日志级别覆盖。资源可以是无线网关或无线设备。
	
	Note
	此 API 每个账户限制 200 个日志级别覆盖。
		{
		"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 名称	描述	IAM 策略示例

ResetAllResourceLogLevels	删除所有资源（包括无线网关和无线设备）的日志级别覆盖。
	
	Note
	此 API 不会影响使用 UpdateLogLevelsByResourceTypes API 设置的日志级别。
		<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 名称	描述	IAM 策略示例
ResetResourceLogLevel	删除给定资源标识符和资源类型的日志级别覆盖。资源可以是无线网关或无线设备。	{ "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 名称	描述	IAM 策略示例
UpdateLogLevelsByResourceTypes	设置默认日志级别，或按资源类型设置日志级别。您可以将此 API 用于无线设备或无线网关的日志选项，并控制将在 CloudWatch 中显示的日志消息。 **Note** 事件是可选的，事件类型与资源类型相关联。有关更多信息，请参阅事件和资源类型。	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:UpdateLogLevelsByResourceTypes" ], "Resource": [ "*" ] } ] }`

使用 CLI 配置资源的日志级别

本节介绍了如何使用 API 或 AWS CLI 配置 AWS IoT Wireless 资源的日志级别。

使用 CLI 之前：

- 请确保为要为其运行 CLI 命令的 API 创建了 IAM 策略，如前所述。
- 您需要提供要使用角色的 Amazon Resource Name (ARN)。如果需要创建用于日志记录的角色，请参阅 为 AWS IoT Wireless 创建日志记录角色和策略。

为什么要使用 AWS CLI

预设情况下，如果您如 为 AWS IoT Wireless 创建日志记录角色和策略 中所述创建 IAM 角色，IoTWirelessLogsRole，您将在默认日志级别为 ERROR 的 AWS Management Console 中看到 CloudWatch Logs。要更改所有资源或特定资源的原定设置日志级别，请使用 AWS IoT Wireless 无线日志记录 API 或 CLI。

如何使用 AWS CLI

API 操作可分为以下类型，具体取决于您是要为所有资源配置日志级别还是为特定资源配置日志级别：

- API 操作GetLogLevelsByResourceTypes 和 UpdateLogLevelsByResourceTypes 可以为您账户中特定类型的所有资源检索和更新日志级别，例如无线网关或 LoRaWAN 或 Sidewalk 设备。
- API 操作 GetResourceLogLevel、PutResourceLogLevel 和 ResetResourceLogLevel 可以检索、更新和重置您使用资源标识符指定的各个资源的日志级别。
- 对于您使用 PutResourceLogLevel API 指定日志级别覆盖操作的所有资源， API 操作 ResetAllResourceLogLevels 会将日志级别覆盖重置为 null。

使用 CLI 为 AWS IoT 配置资源特定的日志记录

> **Note**
>
> 您也可以使用 API 执行此流程，即使用 AWS API 中与此处显示的 CLI 命令对应的方法。

1. 预设情况下，所有资源的日志级别都设置为 ERROR。要为账户中的所有资源设置默认日志级别或按资源类型设置日志级别，请使用 update-log-levels-by-resource-types 命令。以下示例显示如何创建 JSON 文件 Input.json，并将其作为 CLI 命令的输入提供。您可以使用此命令有选择地禁用特定类型的资源和时间的日志记录或覆盖默认日志级别。

```
{
    "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

无线设备的日志选项列表。每个日志选项都包括无线设备类型（Sidewalk 或 LoRaWAN），以及无线设备事件日志选项列表。每个无线设备事件日志选项都可以选择是否包括事件类型及其日志级别。

WirelessGatewayLogOptions

无线网关的日志选项列表。每个日志选项都包括无线网关类型 (LoRaWAN) 和无线网关事件日志选项列表。每个无线网关事件日志选项都可以选择是否包括事件类型及其日志级别。

DefaultLogLevel

用于所有资源的日志级别。有效值包括 ERROR、INFO 和 DISABLED。默认值为 INFO。

LogLevel

要用于单个资源类型和事件的日志级别。这些日志级别会覆盖默认日志级别，例如适用于 LoRaWAN 的 INFO 网关日志级别，以及适用于两种事件类型的日志级别 DISABLED 和 ERROR。

运行以下命令以向命令提供 Input.json 文件作为输入。此命令不会生成任何输出。

```
aws iotwireless update-log-levels-by-resource-types \
    --cli-input-json Input.json
```

如果要删除无线设备和无线网关的日志选项，请运行以下命令。

```
{
    "DefaultLogLevel":"DISABLED",
    "WirelessDeviceLogOptions": [],
    "WireslessGatewayLogOptions":[]
}
```

2. update-log-levels-by-resource-types 命令不返回任何输出。使用 get-log-levels-by-resource-types 命令检索资源特定的日志记录信息。该命令返回默认日志级别以及无线设备和无线网关日志选项。

> **Note**
>
> get-log-levels-by-resource-types 命令无法在 CloudWatch 控制台中直接检索日志级别。您可以使用 get-log-levels-by-resource-types 命令获取您使用 update-log-levels-by-resource-types 命令为资源指定的最新日志级别信息。

```
aws iotwireless get-log-levels-by-resource-types
```

当您运行以下命令时，它会返回您使用 update-log-levels-by-resource-types 指定的最新日志记录信息。例如，如果删除无线设备日志选项，则运行 get-log-levels-by-resource-types 将返回此值，为 `null`。

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions": null,
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
            }
          ]
        }
      ]
}
```

3. 要控制单个无线网关或无线设备资源的日志级别，请使用以下 CLI 命令：

   - put-resource-log-level
   - get-resource-log-level

- reset-resource-log-level

有关何时使用这些 CLI 的示例，假设您的账户中有大量正在记录的无线设备或网关。如果您只想排除某些无线设备的故障，可以通过将 `DefaultLogLevel` 设置为 `DISABLED` 禁用所有无线设备的日志记录，然后针对您账户中的设备，使用 put-resource-log-level 将 `LogLevel` 设置为 `ERROR`。

```
aws iotwireless put-resource-log-level \
    --resource-identifier
    --resource-type WirelessDevice
    --log-level ERROR
```

在本例中，命令仅将特定无线设备资源的日志级别设置为 `ERROR`，而所有其他资源的日志均已禁用。此命令不会生成任何输出。要检索此信息并验证是否已设置日志级别，请使用 get-resource-log-level 命令。

4. 在上一步中，调试问题并解决错误后，您可以运行 reset-resource-log-level 命令将该资源的日志级别重置为 `null`。如果您使用 `put-resource-log-level` 命令为多个无线设备或网关资源设置日志级别覆盖（例如，为多个设备排除故障），您可以使用 reset-all-resource-log-levels 命令为所有这些资源将日志级别覆盖重置为 `null`。

```
aws iotwireless reset-all-resource-log-levels
```

此命令不会生成任何输出。要检索资源的日志记录信息，请运行 get-resource-log-level 命令。

后续步骤

您已了解如何创建日志记录角色并使用 AWS IoT Wireless API 用于配置 适用于 LoRaWAN 的 AWS IoT Core 资源的日志记录。接下来，要了解有关监控日志条目的信息，请转到 使用 CloudWatch Logs 监控 AWS IoT Wireless。

使用 CloudWatch Logs 监控 AWS IoT Wireless

适用于 LoRaWAN 的 AWS IoT Core 具有超过 50 个原定设置启用的 CloudWatch 日志条目。每个日志条目都描述了事件类型、日志级别和资源类型。有关更多信息，请参阅AWS IoT Wireless 资源和日志级别。

如何监控 AWS IoT Wireless 资源

如果为 AWS IoT Wireless 启用日志记录，当消息在您的设备与 AWS IoT 之间来回传递时， AWS IoT Wireless 会发送关于每条消息的进度事件。预设情况下， AWS IoT Wireless 日志条目具有原定设置的错误日志级别。如果如 为 AWS IoT Wireless 创建日志记录角色和策略 中所述启用日志记录，您将在 CloudWatch 控制台中看到具有默认日志级别 ERROR 的消息。使用此日志级别，消息将仅显示您正在使用的所有无线设备和网关资源的错误信息。

如果希望日志显示其他信息，例如日志级别为 INFO 的信息，或者想要禁用某些设备的日志并仅显示某些设备的日志消息，则可以使用 AWS IoT Wireless 日志记录 API。有关更多信息，请参阅使用 CLI 配置资源的日志级别。

您还可以创建筛选条件表达式以仅显示所需消息。

在您可以在控制台中查看 AWS IoT Wireless 日志前

要使 /aws/iotwireless 日志组显示在 CloudWatch 控制台中，则必须执行以下操作。

- AWS IoT Wireless 中已启用日志记录。有关如何启用在 AWS IoT Wireless 中启用日志记录的更多信息，请参阅 为 AWS IoT Wireless 配置日志记录。
- 通过执行 AWS IoT Wireless 操作，写入部分日志条目。

要更有效地创建和使用筛选条件表达式，我们建议您按照下列主题中所述尝试使用 CloudWatch Insights。我们还建议您按照主题在此处显示的顺序进行操作。这将首先帮助您使用 CloudWatch Log groups（日志组），以了解可用于在控制台中查看日志条目的不同类型的资源、其事件类型和日志级别。然后，您可以了解如何通过使用 CloudWatch Insights 来创建筛选条件表达式，从资源中获取更多有用的信息。

主题

- 查看 CloudWatch AWS IoT Wireless 日志条目
- 使用 CloudWatch Insights 为 AWS IoT Wireless 筛选日志

查看 CloudWatch AWS IoT Wireless 日志条目

在根据 为 AWS IoT Wireless 创建日志记录角色和策略 中所述为 AWS IoT Wireless 配置日志记录并写入部分日志条目时，您可以执行以下步骤以在 CloudWatch 控制台中查看日志条目。

在 CloudWatch Logs 组控制台中查看 AWS IoT 日志

在 CloudWatch 控制台中，CloudWatch Logs 显示在名为 /aws/iotwireless 的日志组中。有关 CloudWatch Logs 的更多信息，请参阅 CloudWatch Logs。

要在 CloudWatch 控制台中查看您的 AWS IoT 日志

导航到 CloudWatch 控制台，然后选择导航窗格中的 Log groups（日志组）。

1. 在 Filter（筛选条件）文本框中，输入 **/aws/iotwireless**，然后选择 /aws/iotwireless 日志组。
2. 要查看 适用于 LoRaWAN 的 AWS IoT Core 为您的账户生成的完整列表，请选择 Search all（搜索全部）。要查看单个日志流，请选择展开图标。
3. 要筛选日志流，您也可以在 Filter events（筛选事件）文本框中输入一条查询。以下是可以尝试输入的查询：

   - `{ $.logLevel = "ERROR" }`

     使用此筛选条件可以查找所有日志级别为 ERROR 的日志，您可以展开各个错误流以读取错误消息，这将帮助您解决这些错误。

   - `{ $.resource = "WirelessGateway" }`

     查找适用于 WirelessGateway 资源的所有日志，无论日志级别为何。

   - `{ $.event = "CUPS_Request" && $.logLevel = "ERROR" }`

     查找所有事件状态为 CUPS_Request 且日志级别为 ERROR 的日志。

事件和资源类型

下表显示了不同类型的事件，而您将看到其日志条目。事件类型还取决于资源类型是无线设备还是无线网关。您可以对资源和事件类型使用默认日志级别，也可以通过为每个资源和事件类型指定日志级别来覆盖默认日志级别。

基于使用资源的事件类型

资源	资源类型	事件类型	
无线网关	LoRaWAN	• CUPS_Request • 证书	
无线设备	LoRaWAN	• 联接 • 重新加入 • Uplink_Data	

资源	资源类型	事件类型	
		• Downlink_Data	
无线设备	Sidewalk	• 注册 • Uplink_Data • Downlink_Data	

以下主题包含有关这些事件类型以及无线网关和无线设备日志条目的详细信息。

主题

无线网关和无线设备资源的日志条目

启用日志记录后，您可以查看无线网关和无线设备的日志条目。以下部分介绍基于您的资源和事件类型的各种日志条目。

无线网关日志条目

此部分显示了无线网关资源的一些示例日志条目，您可在 CloudWatch 控制台中看到这些资源。这些日志消息的事件类型可以为 CUPS_Request 或者 Certificate，并且可以配置为在资源级别或事件级别显示 INFO、ERROR 或者 DISABLED 的日志级别。如果您只想查看错误信息，请将日志级别设置为 ERROR。ERROR 日志条目中的消息将包含有关失败原因的信息。

无线网关资源的日志条目可根据以下事件类型进行分类：

- CUPS_Request

  网关上运行的 LoRa Base Station 会定期向 Configuration and Updates Server (CUPS) 发送更新请求。对于此事件类型，如果在为您的无线网关资源配置 CLI 时将日志级别设置为 INFO，则在日志中：

  - 如果事件成功，您将看到 logLevel 为 INFO 的日志消息。这些消息将包括有关发送到您的网关的 CUPS 响应的详细信息和网关详细信息。以下是此日志条目的一个示例。有关 logLevel 和日志条目中其他字段的更多信息，请参阅 AWS IoT Wireless 资源和日志级别。

    ```
    {
        "timestamp": "2021-05-13T16:56:08.853Z",
    ```

```
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "fefff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "INFO",
    "message": "Sending CUPS response of total length 3213 to GatewayEui:
 fefff00000000e2 with  TC Credentials,"
}
```

- 如果出现错误，您将看到 logLevel 为 ERROR 的日志条目，且消息将包含有关错误的详细信息。CUPS_Request 事件可能出现的错误包括：缺少 CUPS CRC、网关的 TC Uri 与 适用于 LoRaWAN 的 AWS IoT Core 不匹配、缺少 IoTWirelessGatewayCertManagerRole，或者无法获取无线网关记录。以下示例显示了一个丢失的 CRC 日志条目。要解决此错误，请检查您的网关设置以验证您输入的 CUPS CRC 是否正确。

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "fefff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "ERROR",
    "message": "The CUPS CRC is missing from the request. Check your gateway setup
 and enter the CUPS CRC,"
}
```

- 证书

这些日志条目将帮助您检查您的无线网关是否提供了正确的证书，用于对与 AWS IoT 的连接进行身份验证。对于此事件类型，如果在为您的无线网关资源配置 CLI 时将日志级别设置为 INFO，则在日志中：

- 如果事件成功，您将看到 logLevel 为 INFO 的日志消息。这些消息将包括有关证书 ID 和无线网关标识符的详细信息。以下是此日志条目的一个示例。有关 logLevel 和日志条目中其他字段的更多信息，请参阅 AWS IoT Wireless 资源和日志级别。

```
{
    "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)"
}
```

- 如果出现错误，您将看到 `logLevel` 为 `ERROR` 的日志条目，且消息将包含有关错误的详细信息。`Certificate` 事件可能出现错误的示例包括无效的证书 ID、无线网关标识符，或无线网关标识符与证书 ID 之间不匹配。以下示例显示了由于无线网关标识符无效导致的 `ERROR`。要纠正该错误，请检查网关标识符。

```
{
    "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)"
}
```

无线设备日志条目

此部分显示了无线设备资源的一些示例日志条目，您将在 CloudWatch 控制台 中控制这些资源。这些日志消息的事件类型取决于您使用的是 LoRaWAN 还是 Sidewalk 设备。每种无线设备资源或事件类型都可以配置为显示 `INFO`、`ERROR` 或者 `DISABLED` 日志级别。

> **Note**
>
> 您的请求不得同时包含 LoRaWAN 和 Sidewalk 无线元数据。要避免 `ERROR` 日志条目出现此状况，请指定 LoRaWAN 或 Sidewalk 无线数据。

LoRaWAN 设备日志条目

LoRaWAN 无线设备的日志条目可根据以下事件类型进行分类：

- **`Join` 和 `Rejoin`**

  当您添加 LoRaWAN 设备并将其连接到 适用于 LoRaWAN 的 AWS IoT Core 时，在设备可以发送上行链路数据之前，您必须完成 `activation` 或者 `join procedure` 流程。有关更多信息，请参阅将您的无线设备添加到 适用于 LoRaWAN 的 AWS IoT Core。

  对于此事件类型，如果在为您的无线网关资源配置 CLI 时将日志级别设置为 `INFO`，则在日志中：

  - 如果事件成功，您将看到 `logLevel` 为 `INFO` 的日志消息。这些消息将包含有关您的加入或重新加入请求状态的详细信息。以下是此日志条目的一个示例。有关 `logLevel` 和日志条目中其他字段的更多信息，请参阅 AWS IoT Wireless 资源和日志级别。

    ```
    {
        "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"
    }
    ```

  - 如果出现错误，您将看到 `logLevel` 为 `ERROR` 的日志条目，且消息将包含有关错误的详细信息。`Join` 和 `Rejoin` 事件可能出现错误的示例包括无效的 LoRaWAN 区域设置或无效的 Message Integrity Code (MIC) 检查。以下示例显示了由于 MIC 检查引起的加入错误。要解决此错误，请检查您是否输入了正确的根密钥。

    ```
    {
        "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 and Downlink_Data

当有效负载从 LoRaWAN 或 Sidewalk 设备发送到 AWS IoT 时，事件类型 Uplink_Data 可用于由 AWS IoT Wireless 生成的消息。事件类型 Downlink_Data 用于与下行链接消息相关的消息，这些消息是从 AWS IoT 发送到无线设备的。

对于此事件类型，如果在为您的无线设备配置 CLI 时将日志级别设置为 INFO，那么在日志中，您将看到：

- 如果事件成功，您将看到 logLevel 为 INFO 的日志消息。这些消息将包含有关已发送的上行链路或下行链路消息的状态以及无线设备标识符的详细信息。下面显示了 Sidewalk 设备的此日志条目示例。有关 logLevel 和日志条目中其他字段的更多信息，请参阅 AWS IoT Wireless 资源和日志级别。

```
{
    "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\"}"
}
```

- 如果出现错误，您将看到 logLevel 为 ERROR，并且消息将包含有关错误的详细信息，这将帮助您解决错误。Registration 事件可能会出现错误的示例包括：身份验证问题、请求无效或过多、无法加密或解密负载，或者无法使用指定 ID 找到无线设备。以下示例显示了处理消息时遇到的权限错误。

```
{
    "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 设备日志条目

您的 Sidewalk 设备的日志条目可以根据以下事件类型进行分类：

- **`Registration`**

  这些日志条目将帮助您监控您使用 AWS IoT Wireless 注册的任何 Sidewalk 设备的状态。对于此事件类型，如果在为您的无线设备资源配置 CLI 时将日志级别设置为 `INFO`，那么在日志中，您将看到 `logLevel` 为 `INFO` 和 `ERROR` 的日志消息。这些消息将包括有关整个注册进度的详细信息。`ERROR` 日志消息将包含有关如何解决注册设备问题的信息。

  下面显示了日志消息的示例，其日志级别为 `INFO`。有关 `logLevel` 和日志条目中其他字段的更多信息，请参阅 AWS IoT Wireless 资源和日志级别。

  ```
  {
      "resource": "WirelessDevice",
      "wirelessDeviceId": "8d0b2775-e19b-4b2a-a351-cb8a2734a504",
      "wirelessDeviceType": "Sidewalk",
      "event": "Registration",
      "logLevel": "INFO",
      "message": "Successfully completed device registration. Amazon SidewalkId =
   2000000002"
  }
  ```

- Uplink_Data and Downlink_Data

  Sidewalk 设备的事件类型 `Uplink_Data` 和 `Downlink_Data` 类似于 LoRaWAN 设备的相应事件类型。有关更多信息，请参阅之前 LoRaWAN 设备日志条目中所述的 Uplink_Data and Downlink_Data 部分。

后续步骤

您已经了解了如何查看资源的日志条目以及在启用 AWS IoT Wireless 日志记录后可以在 CloudWatch 控制台中查看的不同日志条目。虽然您可以使用 Log groups（日志组）创建筛选条件流，我们建议您使用 CloudWatch Insights 创建和使用筛选条件流。有关更多信息，请参阅使用 CloudWatch Insights 为 AWS IoT Wireless 筛选日志。

使用 CloudWatch Insights 为 AWS IoT Wireless 筛选日志

虽然您可以使用 CloudWatch Logs 创建筛选条件表达式，但我们建议您使用 CloudWatch Insights 更有效地根据您的应用程序创建和使用筛选条件表达式。

建议您首先使用 CloudWatch Logs groups（日志组），了解可用于在控制台中查看日志条目的不同类型的资源、事件类型和日志级别。然后，您可以使用本页上的某些筛选条件表达式示例作为参考，为您的 AWS IoT Wireless 资源创建自己的筛选条件。

在 CloudWatch Logs 洞察控制台中查看 AWS IoT 日志

在 CloudWatch 控制台中，CloudWatch Logs 显示在名为 /aws/iotwireless 的日志组中。有关 CloudWatch Logs 的更多信息，请参阅 CloudWatch Logs。

要在 CloudWatch 控制台中查看您的 AWS IoT 日志

导航到 CloudWatch 控制台，然后在导航窗格中选择 Logs Insights。

1. 在 Filter（筛选条件）文本框中，输入 **`/aws/iotwireless`**，然后选择 `/aws/iotwireless` Logs Insights。
2. 要查看日志组的完整列表，请选择 Select log group(s)（选择日志组）。要查看 AWS IoT Wireless 日志组，请选择 `/aws/iotwireless`。

现在，您可以开始输入查询来筛选日志组了。以下部分包含一些有用的查询，可帮助您获得有关资源指标的洞察。

创建有用的查询以筛选和获取对 AWS IoT Wireless 的洞察

您可以使用筛选条件表达式通过 CloudWatch Insights 显示其他有用的日志信息。以下是一些示例查询：

仅显示特定资源类型的日志

您可以创建一个查询，以帮助您仅显示特定资源类型的日志，例如 LoRaWAN 网关或 Sidewalk 设备。例如，要筛选日志以仅显示 Sidewalk 设备的消息，您可以输入以下查询并选择 Run query（运行查询）。要保存此查询，请选择 Save（保存）。

```
fields @message
| filter @message like /Sidewalk/
```

在运行查询后，您将在 Logs（日志）选项卡中看到结果，该选项卡显示与您账户中的 Sidewalk 设备相关的日志时间戳。您还将看到一个条形图，该条形图将显示事件发生的时间（如果以前发生了与您的 Sidewalk 设备相关的事件）。如果您在 Logs（日志）选项卡中展开其中一个结果，将显示如下所示的内容。或者，如果要排除与 Sidewalk 设备相关的故障，可以添加另一个筛选条件，将日志级别设置为 ERROR 并仅显示错误信息。

```
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": "fefff000000011a",
                    "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            fefff000000011a
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
```

显示特定消息或事件

您可以创建一个查询，以帮助您显示特定消息并观察事件发生时间。例如，如果要查看从 LoRaWAN 无线设备发送下行链路消息的时间，可以输入以下查询并选择 Run query（运行查询）。要保存此查询，请选择 Save（保存）。

```
filter @message like /Downlink message sent/
```

在查询运行后，您将在 Logs（日志）选项卡下看到结果，该选项卡显示下行链路消息成功发送到您的无线设备时的时间戳。您还将看到一个条形图，该条形图将显示下行链路消息的发送时间（如果以前有下行链接消息已发送到您的无线设备）。如果您在 Logs（日志）选项卡中展开其中一个结果，将显示如下所示的内容。或者，如果没有发送下行链接消息，则可以修改查询，从而仅显示消息未发送时的结果，以便您可以调试问题。

```
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
```

后续步骤

您已经学会了如何使用 CloudWatch Insights 通过创建查询来筛选日志消息以获取更多有用的信息。您可以组合前面描述的一些筛选条件，并根据所监控的资源设计自己的过滤条件。有关使用 CloudWatch Insights 的更多信息，请参阅使用 CloudWatch Insights 分析日志数据。

使用 CloudWatch Insights 创建查询后，如果已保存查询，则可以根据需要加载并运行已保存的查询。或者，如果单击 CloudWatch Logs Insights 控制台中的 History（历史记录）按钮，您可以查看先前运行的查询并根据需要重新运行它们，或者通过创建其他查询进一步修改这些查询。

AWS IoT Wireless 的事件通知

AWS IoT Wireless 可以发布消息，以通知您登记到 AWS IoT Core 的 LoRaWAN 和 Sidewalk 设备的事件。例如，您可以收到有关事件的通知，例如您账户中的 Sidewalk 设备已配置或注册的时间。

如何通知资源有关事件

发生特定事件时，会发布事件通知。例如，在预调配 Sidewalk 设备时生成事件。每个事件将导致发送单个事件通知。事件通知通过 MQTT 随 JSON 负载一起发布。负载的内容取决于事件的类型。

> **Note**
>
> 至少发布一次事件通知。事件消息可能会多次发布。事件通知的顺序无法保证。

事件和资源类型

下表显示您将收到通知的不同类型的事件。事件类型取决于资源类型是无线设备、无线网关还是 Sidewalk 账户。您还可以在资源级别为资源启用事件，这适用于特定类型的所有资源或选定的资源，如下一节所述。有关不同事件类型的更多信息，请参阅 LoRaWAN 资源的事件通知 和 Sidewalk 资源的事件通知。

基于资源的事件类型

资源	资源类型	事件类型	
无线设备	LoRaWAN	联接	
	Sidewalk	• 设备注册状态 • 近距离	
无线网关	LoRaWAN	连接状态	
Sidewalk 账户	Sidewalk	• 设备注册状态 • 近距离	

用于接收无线事件通知的策略

要接收事件通知，您的设备必须使用合适的策略，该策略允许设备连接到 AWS IoT 设备网关并订阅 MQTT 事件主题。您还必须订阅合适的主题筛选条件。

以下是接收各种无线事件的通知时需要的策略示例。

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

无线事件的 MQTT 主题的格式

为向您发送无线资源的事件通知，AWS IoT 使用以美元 ($) 符号开头的 MQTT 保留主题。您可以订阅并发布这些预留主题 但是，您不能创建以美元符号开头的新主题。

> **Note**
>
> MQTT 主题是特定于您的 AWS 账户 并使用格式 `arn:aws:iotwireless:aws-region:AWS-account-ID:topic/Topic`。有关更多信息，请参阅《AWS IoT Developer Guide》中的 MMQTT topics。

无线设备的保留 MQTT 主题使用以下格式：

- 资源级主题

这些主题适用于您的 AWS 账户 中已登记到 AWS IoT Wireless 的特定类型的所有资源。

```
$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/resources
```

- 标识符级主题

  这些主题适用于您的 AWS 账户 中已登记到 AWS IoT Wireless 的特定类型的选定资源（由资源标识符指定）。

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/
  {resourceIdentifierType}/{resourceID}/{id}
  ```

有关资源级和标识符级的主题的更多信息，请参阅 事件配置。

下表显示了各种事件的 MQTT 主题示例：

事件和 MQTT 主题

事件	MQTT 主题	备注

Sidewalk 设备注册状态	• 资源级主题
	`$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/wireless_devices`
	• 标识符级主题
	`$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}` 对于 `sidewalk_accounts` 为 `amazon_id` ，对于 `wireless_devices` 为 `wireless_device_id`

事件	MQTT 主题	备注
	{resourceID}/{id}	
Sidewalk 近似	• 资源级主题 $aws/iotwireless/events/proximity/{eventType}/sidewalk/wireless_devices • 标识符级主题 $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} 对于 sidewalk_accounts 为 amazon_id ，对于 wireless_devices 为 wireless_device_id

事件	MQTT 主题	备注

LoRaWAN 加入	• 资源级主题
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices`
	• 标识符级主题
	`$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 网关连接状态	• 资源级主题
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways`
	• 标识符级主题
	`$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`

有关不同事件的更多信息，请参阅 LoRaWAN 资源的事件通知 和 Sidewalk 资源的事件通知。

如果您订阅了这些主题，当消息发布到其中一个事件通知主题时，您将收到通知。有关更多信息，请参阅《AWS IoT Developer Guide》中的 MQTT reserved topics。

无线事件的定价

有关订阅事件和接收通知的定价的信息，请参阅 AWS IoT Core 定价。

启用无线资源的事件

在保留主题的订阅者能够接收消息之前，您必须启用事件通知。要做到这一点，您可以使用 AWS Management Console，或者 AWS IoT Wireless API 或 AWS CLI。

事件配置

您可以将事件配置为向属于特定类型的所有资源或针对单独的无线资源发送通知。资源类型可以是无线网关、Sidewalk 合作伙伴账户或无线设备（可以是 LoRaWAN 或 Sidewalk 设备）。有关可以为无线设备启用的事件类型的信息，请参阅 LoRaWAN 资源的事件类型 和 Sidewalk 资源的事件类型。

所有资源

您可以启用事件，以便您的 AWS 账户 中属于特定资源类型的所有资源收到通知。例如，您可以启用一个事件，以向您通知已登记到 适用于 LoRaWAN 的 AWS IoT Core 的所有 LoRaWAN 网关的连接状态所发生的更改。监控这些事件将帮助您在如下情况下获得通知：资源实例集中的某些 LoRaWAN 网关断开连接或者您 AWS 账户 中的一些 Sidewalk 设备的信标丢失。

单独的资源

您还可以将单独的 LoRaWAN 和 Sidewalk 资源添加到您的事件配置中，并为它们启用通知。这将帮助您监控特定类型的单独资源。例如，您可以将选定的 LoRaWAN 和 Sidewalk 设备添加到配置中，并接收这些资源的加入或设备注册状态事件的通知。

先决条件

您的 LoRaWAN 或 Sidewalk 资源必须具有允许其接收事件通知的适当策略。有关更多信息，请参阅用于接收无线事件通知的策略。

使用 AWS Management Console 启用通知

要从控制台启用事件消息，请转到 AWS IoT 控制台的 Settings（设置）选项卡，然后转至 LoRaWAN and Sidewalk event notification（LoRaWAN 和 Sidewalk 事件通知）部分。

您可以为您的 AWS 账户 中属于特定资源类型的所有资源启用通知，并对其进行监视。

针对所有资源启用通知

1. 在 LoRaWAN and Sidewalk event notification（LoRaWAN 和 Sidewalk 事件通知）部分，转到 All resources（所有资源）选项卡，选择 Action（操作），然后选择 Manage events（管理事件）。
2. 启用要监控的事件，然后选择 Update events（更新事件）。如果您不再要监视某些事件，请依次选择 Action（操作）和 Manage events（管理事件），然后禁用这些事件。

您还可以为您的 AWS 账户 中属于特定资源类型的单独资源启用通知，并对其进行监视。

针对单独的资源启用通知

1. 在 LoRaWAN and Sidewalk event notification（LoRaWAN 和 Sidewalk 事件通知）部分，选择 Action（操作），然后选择 Add resources（添加事件）。
2. 选择要接收通知的资源和事件：

   a. 选择是否要监视 LoRaWAN resources（LoRaWAN 资源）或 Sidewalk resources（Sidewalk 资源）的事件。

   b. 根据资源类型，您可以选择要为资源启用的事件。然后，您可以订阅这些事件并接收通知。如果您选择...

      - LoRaWAN resources（LoRaWAN 资源）：您可以为您的 LoRaWAN 设备启用 join（加入）事件，或为您的 LoRaWAN 网关启用 connection status（连接状态）事件。
      - Sidewalk 资源：您可以为您的 Sidewalk 合作伙伴账户和 Sidewalk 设备启用 device registration state（设备注册状态）和/或 proximity（邻近）事件。
3. 根据您选择的资源类型和事件，选择要监控的无线设备或网关。对于所有组合的资源，您可以选择最多 250 个资源。
4. 选择 Submit（提交）以添加您的资源。

您添加的资源将与它们的 MQTT 主题一起出现在控制台的 LoRaWAN and Sidewalk event notification（LoRaWAN 和 Sidewalk 事件通知）部分中对应于资源类型的选项卡中。

- LoRaWAN join（LoRaWAN 加入）事件和 Sidewalk 设备的事件将出现在控制台的 Wireless devices（无线设备）部分。
- LoRaWAN 网关的 Connection status（连接状态）事件将显示在 Wireless gateways（无线网关）部分。

- Sidewalk 账户的 Device registration state（设备注册状态）和 proximity（邻近）事件将出现在 Sidewalk accounts（Sidewalk 账户）选项卡中。

使用 MQTT 客户端订阅主题

根据您是针对所有资源还是针对单独资源类型启用了事件，启用的事件将显示在控制台中，其 MQTT 主题将显示在 All resources（所有资源）选项卡或指定资源类型的选项卡中。

- 如果您选择其中一个 MQTT 主题，您可以前往 MQTT 客户端订阅这些主题并接收消息。
- 如果您添加了多个事件，则可以订阅多个事件主题并接收有关它们的通知。要订阅多个主题，请选择主题，然后依次选择 Action（操作）和 Subscribe（订阅）。

使用 AWS CLI 启用通知

您可以使用 AWS IoT Wireless API 或 AWS CLI 配置事件并将资源添加到配置中。

针对所有资源启用通知

您可以使用 UpdateEventConfigurationByResourceTypes API 或 update-event-configuration-by-resource-types CLI 命令，为您 AWS 账户中属于特定资源类型的所有资源启用通知并对其进行监视。例如：

```
aws iotwireless update-event-configuration-by-resource-types \
   --cli-input-json input.json
```

input.json 的内容

```
{
   "DeviceRegistrationState": {
      "Sidewalk": {
         "AmazonIdEventTopic": "Enabled"
      }
   },
   "ConnectionStatus": {
      "LoRaWAN": {
         "WirelessGatewayEventTopic": "Enabled"
      }
   }
}
```

> **Note**
>
> 使用反斜杠 (\) 转义所有双引号 (")。

您可以通过调用 GetEventConfigurationByResourceTypes API 或通过使用 `get-event-configuration-by-resource-types` CLI 命令获取当前事件配置。例如：

```
aws iotwireless get-event-configuration-by-resource-types
```

针对单独的资源启用通知

要将单独的资源添加到事件配置中并控制使用 API 或 CLI 发布的事件，请调用 UpdateEventConfigurations 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 的内容

```
{
   "Join": {
      "LoRaWAN": {
         "DevEuiEventTopic": "Disabled"
      },
      "WirelessDeviceIdEventTopic": "Enabled"
   }
}
```

> **Note**
>
> 使用反斜杠 (\) 转义所有双引号 (")。

您可以通过调用 GetResourceEventConfiguration API 或通过使用 `get-resource-event-configuration` CLI 命令获取当前事件配置。例如：

```
aws iotwireless get-resource-event-configuration \
    --identifier-type WirelessDeviceId \
    --identifier 1ffd32c8-8130-4194-96df-622f072a315f
```

列出事件配置

您也可以使用 AWS IoT Wireless API 或 AWS CLI 列出至少启用了一个事件主题的事件配置。要列出配置，请使用 ListEventConfigurations API 操作或通过使用 `list-event-configurations` CLI 命令。例如：

```
aws iotwireless list-event-configurations --resource-type WirelessDevice
```

LoRaWAN 资源的事件通知

您可以使用 AWS Management Console 或 AWS IoT Wireless API 操作向您通知 LoRaWAN 设备和网关的事件。有关事件通知以及如何启用它们的信息，请参阅 AWS IoT Wireless 的事件通知和 启用无线资源的事件。

LoRaWAN 资源的事件类型

您可以为 LoRaWAN 资源启用的事件包括：

- 向您通知 LoRaWAN 设备加入事件的加入事件。当设备加入 适用于 LoRaWAN 的 AWS IoT Core 时，或者收到类型为 0 或 2 的重新加入请求时，您将收到通知。
- 当 LoRaWAN 网关的连接状态更改为已连接或断开连接时向您发出通知的连接状态事件。

以下部分包含有关 LoRaWAN 资源的事件的更多信息：

主题

- LoRaWAN 加入事件
- 连接状态事件

LoRaWAN 加入事件

适用于 LoRaWAN 的 AWS IoT Core 可以发布消息，向您通知登记到 AWS IoT 的 LoRaWAN 设备的加入事件。当收到类型为 0 或 2 的加入或重新加入请求并且设备已加入 适用于 LoRaWAN 的 AWS IoT Core 时，加入事件会通知您。

加入事件的工作原理

当您将 LoRaWAN 设备登记到 适用于 LoRaWAN 的 AWS IoT Core 时，适用于 LoRaWAN 的 AWS IoT Core 会执行将您的设备加入 到 适用于 LoRaWAN 的 AWS IoT Core 的过程。然后，您的设备将被激活以供使用，并可以发送上行消息以指示其可用。设备加入后，可以在设备与 适用于 LoRaWAN 的 AWS IoT Core 之间交换上行链路和下行链路消息。有关登记您的设备的更多消息，请参阅 将您的设备登记到 适用于 LoRaWAN 的 AWS IoT Core。

您可以启用事件以在设备加入 适用于 LoRaWAN 的 AWS IoT Core 时通知您。如果加入事件失败，收到类型为 0 或 2 的重新加入请求以及加入事件被接受时，您也会收到通知。

启用 LoRaWAN 加入事件

在 LoRaWAN 加入保留主题的订阅方可以接收消息之前，必须从 AWS Management Console 或使用 API 或 CLI 为订阅方启用事件通知。您可以为 AWS 账户 中的所有 LoRaWAN 资源或所选资源启用这些事件。有关如何启用这些事件的更多信息，请参阅 启用无线资源的事件。

LoRaWAN 事件的 MQTT 主题的格式

LoRaWAN 设备的保留 MQTT 主题使用以下格式。如果您已订阅了这些主题，那么注册到您的 AWS 账户 的所有 LoRaWAN 设备都可以接收通知：

- 资源级主题

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices
  ```

- 标识符主题

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices/
  {resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必须是 `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，您可以订阅以下主题，以便在 适用于 LoRaWAN 的 AWS IoT Core 已接受来自设备的加入请求时接收事件通知。

```
$aws/iotwireless/events/join/join_accepted/lorawan/wireless_devices/
wireless_device_id/{id}
```

您还可以使用 + 通配符同时订阅多个主题。+ 通配符匹配级别中包含字符的任何字符串，例如以下主题：

```
$aws/iotwireless/events/join/join_req_received/lorawan/wireless_devices/
wireless_device_id/+
```

> **Note**
>
> 您不能使用通配符 # 订阅保留主题。

有关订阅主题时使用 + 通配符的更多信息，请参阅《AWS IoT Developer Guide》中的 MQTT topic filters。

LoRaWAN 加入事件的消息有效负载

下面显示了 LoRaWAN 加入事件的消息有效负载。

```
{
 // 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",
    }
}
```

负载包含以下属性：

eventId

适用于 LoRaWAN 的 AWS IoT Core 生成的唯一事件 ID（字符串）。

eventType

发生的事件类型。可以是以下任一值：

- `join_req_received`：此字段将显示 EUI 参数 `JoinEui` 或 `AppEui`
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`：此字段将显示 `NetId` 和 `DevAddr`。

wirelessDeviceId

LoRaWAN 设备的 ID。

timestamp

事件发生时的 Unix 时间戳。

DevEui

在设备标注或设备文档中找到的设备的唯一标识符。

DevAddr 和 EUI（可选）

这些字段是可选的设备地址和 EUI 参数 `JoinEUI` 或 `AppEUI`。

连接状态事件

适用于 LoRaWAN 的 AWS IoT Core 可以发布消息，向您通知登记到 AWS IoT 的 LoRaWAN 网关的连接状态事件。当 LoRaWAN 网关的连接状态更改为已连接或断开连接时，连接状态事件会向您发出通知。

连接状态事件的工作原理

在将网关登记到 适用于 LoRaWAN 的 AWS IoT Core 之后，您可以将网关连接到 适用于 LoRaWAN 的 AWS IoT Core 并验证其连接状态。当网关连接状态更改为已连接或断开连接时，此事件会通知您。有关将您的网关登记并连接到 适用于 LoRaWAN 的 AWS IoT Core 的更多信息，请参阅将您的网关登记到 适用于 LoRaWAN 的 AWS IoT Core和连接您的 LoRaWAN 网关并验证其连接状态。

LoRaWAN 网关的 MQTT 主题的格式

LoRaWAN 网关的保留 MQTT 主题使用以下格式。如果您已经订阅了这些主题，那么注册到 AWS 账户 的所有 LoRaWAN 网关都可以接收通知：

- 对于资源级主题：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways
  ```

- 对于标识符主题：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/
  wireless_gateways/{resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必须是 `connection_status`。

{eventType}

{eventType} 可以是 `connected` 或 `disconnected`。

{resourceID}

{resourceID} 可以是 `gateway_eui` 或 `wireless_gateway_id`。

例如，您可以订阅以下主题以便在所有网关连接到 适用于 LoRaWAN 的 AWS IoT Core 时接收事件通知：

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/{id}
```

您还可以使用 + 通配符同时订阅多个主题。+ 通配符匹配级别中包含字符的任何字符串，例如以下主题：

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/+
```

> **Note**
>
> 您不能使用通配符 # 订阅保留主题。

有关订阅主题时使用 + 通配符的更多信息，请参阅《AWS IoT Developer Guide》中的 MQTT topic filters。

连接状态事件的消息有效负载

下面显示了连接状态事件的消息有效负载。

```
{
 // General fields
    "eventId": "string",
    "eventType": "connected|disconnected",
    "WirelessGatewayId": "string",
    "timestamp": "timestamp",

 // Event-specific fields
    "LoRaWAN": {
        "GatewayEui": "string"
    }
}
```

负载包含以下属性：

eventId

适用于 LoRaWAN 的 AWS IoT Core 生成的唯一事件 ID（字符串）。

eventType

发生的事件类型。可以是 `connected` 或 `disconnected`。

wirelessGatewayId

LoRaWAN 网关的 ID。

timestamp

事件发生时的 Unix 时间戳。

GatewayEui

网关标签或网关文档中找到的网关的唯一标识符。

Sidewalk 资源的事件通知

您可以使用 AWS Management Console 或 AWS IoT Wireless API 操作就 Sidewalk 设备和合作伙伴账户的事件向您发出通知。有关事件通知以及如何启用它们的信息，请参阅 AWS IoT Wireless 的事件通知 和 启用无线资源的事件。

Sidewalk 资源的事件类型

您可以为 Sidewalk 资源启用的事件包括：

- 通知您 Sidewalk 设备状态更改的设备事件，例如设备已注册并准备使用的时间。
- 接近事件，当 AWS IoT Wireless 收到来自 Amazon Sidewalk 的已收到或丢失信标的通知时通知您。

以下各部分包含有关 Sidewalk 资源的事件的更多信息：

主题

- 设备注册状态事件
- 近似事件

设备注册状态事件

设备注册状态事件在设备注册状态发生更改时发布事件通知，例如在 Sidewalk 设备已设置或注册时。这些事件为您提供有关设备从预调配到注册的不同状态的信息。

设备注册状态事件的工作原理

当您使用 Amazon Sidewalk 和 AWS IoT Wireless 登记 Sidewalk 设备时，AWS IoT Wireless 将执行一项 `create` 操作并将 Sidewalk 设备添加到您的 AWS 账户中。然后，您的设备进入预调配状态，然

后 eventType变成 provisioned。有关登记您的设备的更多消息，请参阅 开始使用适用于 Amazon Sidewalk 的 AWS IoT Core。

对设备进行 provisioned 后，Amazon Sidewalk 将执行 register 操作，向 AWS IoT Wireless 注册您的 Sidewalk 设备。注册过程开始，使用 AWS IoT 设置加密和会话密钥。注册设备后，eventType 变成 registered，并且您的设备已能够使用。

对设备进行 registered 后，Sidewalk 可以发送请求以 deregister 您的设备。然后，AWS IoT Wireless 执行请求并将设备状态更改回到 provisioned。有关设备状态的更多信息，请参阅 Device Tester。

启用设备注册状态事件的通知

在设备注册状态保留主题的订阅方可以接收消息之前，必须从 AWS Management Console 或使用 API 或 CLI 为其启用事件通知。您可以为 AWS 账户 中的所有 Sidewalk 资源或所选资源启用这些事件。有关如何启用这些事件的更多信息，请参阅 启用无线资源的事件。

设备注册状态事件的 MQTT 主题格式

要通知设备注册状态事件，您可以订阅以美元 ($) 符号开头的 MQTT 保留主题。有关更多信息，请参阅《AWS IoT Developer Guide》中的 MMQTT topics。

Sidewalk 设备注册状态事件的保留 MQTT 主题使用以下格式：

- 对于资源级主题：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 对于标识符主题：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必须是 device_registation_state。

{eventType}

{eventType} 可以是 provisioned 或 registered。

{resourceType}

{resourceType} 可以是 `sidewalk_accounts` 或 `wireless_devices`。

{resourceID}

{resourceID} 对于 {resourceType} `sidewalk_accounts` 为 `amazon_id`，对于 {resourceType} `wireless_devices` 为 `wireless_device_id`。

您还可以使用 + 通配符同时订阅多个主题。+ 通配符匹配级别中包含字符的任何字符串。例如，如果您希望收到所有可能的事件类型（`provisioned` 和 `registered`）的通知以及注册到特定 Amazon ID 的所有设备通知，可以使用以下主题筛选条件：

```
$aws/iotwireless/events/device_registration_state/+/sidewalk/
sidewalk_accounts/amazon_id/+
```

> **Note**
>
> 您不能使用通配符 # 订阅保留主题。有关主题筛选器的更多信息，请参阅《AWS IoT Developer Guide》中的 MQTT topic filters。

设备注册状态事件的消息负载

启用设备注册状态事件的通知后，事件通知将通过 MQTT 随 JSON 负载一起发布。事件包含以下负载示例：

```
{
    "eventId": "string",
    "eventType": "provisioned|registered",
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",

    // Event-specific fields
    "operation": "create|deregister|register",
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

负载包含以下属性：

eventId

唯一事件 ID (字符串)。

eventType

发生的事件类型。可以是 `provisioned` 或 `registered`。

wirelessDeviceId

无线设备的标识符。

timestamp

事件发生时的 Unix 时间戳。

操作

触发事件的操作。有效值包括 `create`、`register` 和 `deregister`。

sidewalk

您要接收其事件通知的 Sidewalk Amazon ID 或 `SidewalkManufacturingSn`。

近似事件

AWS IoT 从 Sidewalk 设备接收信标时近似事件发布事件通知。当您的 Sidewalk 设备接近 Amazon Sidewalk 时，Amazon Sidewalk 会按定期间隔筛选从您的设备发送的信标，并通过 AWS IoT Wireless 接收信标。然后，在收到信标时，AWS IoT Wireless 通知您这些事件。

近似事件的工作原理

近似事件会在 AWS IoT 收到信标时通知您，您的 Sidewalk 设备可以随时发射信标。当您的设备靠近 Amazon Sidewalk 时，Sidewalk 将接收信标，并按定期时间间隔将信标转发给 AWS IoT Wireless。Amazon Sidewalk 已将此时间间隔配置为 10 分钟。当 AWS IoT Wireless 从 Sidewalk 接收到信标时，您将收到事件通知。

近似事件将在发现信标或信标丢失时通知您。您可以配置近似事件通知的时间间隔。

启用近似事件的通知

在 Sidewalk 近似保留主题的订阅方可以接收消息之前，必须从 AWS Management Console 或使用 API 或 CLI 为订阅方启用事件通知。您可以为 AWS 账户 中的所有 Sidewalk 资源或所选资源启用这些事件。有关如何启用这些事件的更多信息，请参阅 启用无线资源的事件。

近似事件的 MQTT 主题的格式

要通知近似事件，您可以订阅以美元 ($) 符号开头的 MQTT 保留主题。有关更多信息，请参阅《AWS IoT Developer Guide》中的 MMQTT topics。

Sidewalk 近似事件的保留 MQTT 主题使用以下格式：

- 对于资源级主题：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 对于标识符主题：

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

其中：

{eventName}

{eventName} 必须是 `proximity`。

{eventType}

{eventType} 可以是 `beacon_discovered` 或 `beacon_lost`。

{resourceType}

{resourceType} 可以是 `sidewalk_accounts` 或 `wireless_devices`。

{resourceID}

{resourceID} 对于 {resourceType} `sidewalk_accounts` 为 `amazon_id`，对于 {resourceType} `wireless_devices` 为 `wireless_device_id`。

您还可以使用 + 通配符同时订阅多个主题。+ 通配符匹配级别中包含字符的任何字符串。例如，如果您希望收到所有可能的事件类型（`beacon_discovered` 和 `beacon_lost`）的通知以及注册到特定 Amazon ID 的所有设备通知，可以使用以下主题筛选条件：

`$aws/iotwireless/events/proximity/+/sidewalk/sidewalk_accounts/amazon_id/+`

> **Note**
>
> 您不能使用通配符 # 订阅保留主题。有关主题筛选器的更多信息，请参阅《AWS IoT Developer Guide》中的 MQTT topic filters。

近似事件的消息负载

启用近似事件通知后，将通过 MQTT 随 JSON 负载一起发布事件消息。事件包含以下负载示例：

```
{
    "eventId": "string",
    "eventType": "beacon_discovered|beacon_lost",
    "WirelessDeviceId": "string",
    "timestamp": "1234567890123",

    // Event-specific fields
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

负载包含以下属性：

eventId

唯一的事件 ID，字符串。

eventType

发生的事件类型。可以是 `beacon_discovered` 或 `beacon_lost`。

WirelessDeviceId

无线设备的标识符。

timestamp

事件发生时的 Unix 时间戳。

sidewalk

您要接收其事件通知的 Sidewalk Amazon ID 或 `SidewalkManufacturingSn`。

AWS IoT Wireless API 操作

在登记 LoRaWAN 或 Sidewalk 终端设备或创建用于批量预置 Sidewalk 终端设备的导入任务时，您可以执行以下额外的 API 操作。

以下各节包含有关这些 API 操作的更多信息。

主题

可对设备配置文件执行的 AWS IoT Wireless API 操作

您可以对 LoRaWAN 和 Sidewalk 设备配置文件执行以下 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

以下各节将向您展示如何列出和删除配置文件。有关创建和检索设备配置文件的信息，请参阅：

- 添加设备配置文件
- 步骤 1：创建设备配置文件

列出您的 AWS 账户中的设备配置文件

您可以使用 `ListDeviceProfiles` API 操作列出您的 AWS 账户中添加到 AWS IoT Wireless 的设备配置文件。您可以使用此信息来识别要将此配置文件关联到的设备。

要筛选列表以便仅显示 LoRaWAN 或 Sidewalk 设备配置文件，请在运行 API 时设置 Type。下面显示了 CLI 命令示例：

```
aws iotwireless list-device-profiles --wireless-device-type "Sidewalk"
```

运行此命令会返回您添加的设备配置文件列表，包括其配置文件标识符和 Amazon 资源名称（ARN）。要检索有关特定配置文件的更多详细信息，请使用 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"
        }
    ]
}
```

从您的 AWS 账户中删除设备配置文件

您可以使用 DeleteDeviceProfile API 操作删除您的设备配置文件。下面显示了 CLI 命令示例：

> **Warning**
>
> 删除操作无法撤消。将从您的 AWS 账户中永久删除设备配置文件。

```
aws iotwireless delete-device-profile --name "SidewalkProfile"
```

此命令不会生成任何输出。您可以使用 GetDeviceProfile API 或 ListDeviceProfiles API 操作来验证配置文件是否已从您的账户中删除。

可对 LoRaWAN 和 Sidewalk 设备执行的 AWS IoT Wireless API 操作

您可以对 LoRaWAN 和 Sidewalk 设备执行以下 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

以下各节将向您展示如何列出和删除设备。有关创建无线设备和检索设备信息的信息，请参阅：

- 将您的无线设备添加到 适用于 LoRaWAN 的 AWS IoT Core
- 步骤 2：添加 Sidewalk 设备

将您的 AWS 账户中的无线设备与 IoT 事物相关联

要将您的 LoRaWAN 和 Sidewalk 设备与 AWS IoT 事物关联，请使用 AssociateWirelessDeviceWithThing API 操作。

AWS IoT 中的事物可让您更轻松地搜索和管理设备。将事物与设备关联将允许设备访问其他 AWS IoT Core 功能。相关使用此 API 的更多信息，请参阅 AssociateWirelessDeviceWithThing。

下面显示了运行此命令的示例。运行此命令不会生成任何输出。

```
aws iotwireless associate-wireless-device-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MySidewalkThing"
```

要取消无线设备与 AWS IoT 事物的关联，请使用 DisassociateWirelessDeviceFromThing API 操作，如以下示例所示。

```
aws iotwireless disassociate-wireless-device-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

列出您的 AWS 账户中的无线设备

要列出您的 AWS 账户中添加到 AWS IoT Wireless 的无线设备，请使用 ListWirelessDevices API 操作。要筛选列表以便仅返回 LoRaWAN 或 Sidewalk 设备，请设置 WirelessDeviceType。

下面显示了运行此命令的示例：

```
aws iotwireless list-wireless-devices --wireless-device-type Sidewalk
```

运行此命令会返回您添加的设备列表，包括其配置文件标识符和 Amazon 资源名称（ARN）。要检索有关特定设备的更多详细信息，请使用 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"
        }
    ]
}
```

从您的 AWS 账户中删除无线设备

要删除无线设备，请将要删除的设备的 WirelessDeviceID 传递给 DeleteWirelessDevice API 操作。

下面显示了命令示例：

```
aws iotwireless delete-wireless-device --id "23456789-abcd-0123-bcde-fabc012345678"
```

此命令不会生成任何输出。您可以使用 GetWirelessDevice API 或 ListWirelessDevices API 操作来验证设备是否已从您的账户中删除。

可对无线设备目标执行的 AWS IoT Wireless API 操作

您可以对 LoRaWAN 和 Sidewalk 设备的目标执行以下 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

以下各节将向您介绍如何获取、列出、更新和删除目标。有关如何创建目标的信息，请参阅为 Sidewalk 终端设备添加目标。

获取有关您的目标的信息

您可以使用 GetDestination API 操作来获取有关您针对 AWS IoT Wireless 添加到您的账户中的目标的信息。提供目标名称作为 API 的输入。API 将返回有关与指定的标识符匹配的目标的信息。

下面显示了 CLI 命令示例：

```
aws iotwireless get-destination --name SidewalkDestination
```

运行此命令会返回目标的参数。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "SidewalkDestination",
    "Expression": "IoTWirelessRule",
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

更新目标的属性

使用 UpdateDestination API 操作更新您针对 AWS IoT Wireless 添加到您的账户中的目标的属性。下面显示了更新描述属性的示例 CLI 命令：

```
aws iotwireless update-destination --name SidewalkDestination \
    --description "Destination for messages processed using IoTWirelessRule"
```

列出您的 AWS 账户中的目标

使用 ListDestinations API 操作列出您的 AWS 账户中添加到 AWS IoT Wireless 的目标。要筛选列表以便仅返回 LoRaWAN 和 Sidewalk 终端设备的目标，请使用 `WirelessDeviceType` 参数。

下面显示了 CLI 命令示例：

```
aws iotwireless list-destinations --wireless-device-type "Sidewalk"
```

运行此命令将返回您添加的目标列表，包括其 Amazon 资源名称（ARN）。要检索有关特定目标的更多详细信息，请使用 `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"
        }
    ]
}
```

从您的 AWS 账户中删除目标

要删除您的目标，请将要删除的目标的名称作为输入传递给 DeleteDestination API 操作。下面显示了 CLI 命令示例：

> **Warning**
>
> 删除操作无法撤消。将从您的 AWS 账户中永久删除目标。

```
aws iotwireless delete-destination --name "SidewalkDestination"
```

此命令不会生成任何输出。您可以使用 GetDestination API 或 ListDestinations API 操作来验证目标是否已从您的账户中删除。

用于批量预置的 适用于 Amazon Sidewalk 的 AWS IoT Core API 操作

您可以执行以下 API 操作来批量预置 Sidewalk 终端设备：

- 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

以下各节将向您介绍如何获取、列出、更新和删除导入任务。有关创建导入任务的信息，请参阅用于批量预置的 适用于 Amazon Sidewalk 的 AWS IoT Core API 操作。

获取有关导入任务的信息

您可以使用 ListDevicesForWirelessDeviceImportTask API 操作来检索有关特定导入任务的信息以及该任务中设备的登记状态。作为 API 操作的输入，请指定您从 StartWirelessDeviceImportTask 或 StartSingleWirelessDeviceImportTask API 操作中获得的导入任务 ID。然后，API 将返回有关与指定的标识符匹配的导入任务的信息。

下面显示了 CLI 命令示例：

```
aws iotwireless list-devices-for-wireless-device-import-task --id e2a5995e-743b-41f2-a1e4-3ca6a5c5249f
```

运行此命令会返回您的导入任务信息和设备登记状态。

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

获取导入任务设备摘要

要获取您添加到特定导入任务的设备的登记状态的摘要信息计数，请使用 GetWirelessDeviceImportTask API 操作。下面显示了 CLI 命令示例。

```
aws iotwireless get-wireless-device-import-task --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

以下代码显示了来自命令的示例响应。

```
{
   "NumberOfFailedImportedDevices": 2,
   "NumberOfOnboardedImportedDevices": 4,
   "NumberOfPendingImportedDevices": 1
}
```

将设备添加到导入任务

使用 UpdateWirelessDeviceImportTask API 操作将设备添加到您添加的现有导入任务中。您可以使用此 API 操作添加以前未包含在您使用 StartWirelessDeviceImportTask API 操作创建的任务中的设备的序列号（SMSN）。

要将设备附加到导入任务，作为 API 请求的一部分，请在 Amazon S3 桶中指定一个新的 CSV 文件，其中包含待添加设备的序列号。只有在当前处于导入任务中的设备的登记过程尚未开始时，才会接受该请求。如果登记过程已经开始，则 UpdateWirelessDeviceImportTask API 请求将失败。

如果您仍想将设备附加到导入任务，则可以再次执行 UpdateWirelessDeviceImportTask API 操作。在执行此 API 操作之前，第一个 UpdateWirelessDeviceImportTask API 请求必须已完成处理 S3 桶中的 CSV 文件。

> **Note**
>
> 当您执行 ListImportedWirelessDeviceTasks API 请求时，当前不会返回使用 UpdateWirelessDeviceImportTask API 操作指定的新 CSV 文件的 S3 URL。相反，API 操作会返回最初使用 StartWirelessDeviceImportTask API 请求发送的请求的 S3 URL。

下面显示了 CLI 命令示例。

```
aws iotwireless update-wireless-device-import task \
    --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f" \
    --sidewalk '{"FileForCreateDevices": "s3://import_task_bucket/import_file3"}'
```

列出您的 AWS 账户中的导入任务

使用 ListWirelessDeviceImportTasks API 或 list-imported-wireless-device-tasks CLI 命令列出您的 AWS 账户中的导入任务。下面显示了 CLI 命令示例。

```
aws iotwireless list-wireless-device-import-tasks
```

运行此命令会返回您创建的导入任务列表。该列表包括这些任务的 Amazon S3 CSV 文件和指定的 IAM 角色、导入任务 ID 以及设备登记状态的摘要信息。

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

从您的 AWS 账户中删除导入任务

要删除导入任务，请将导入任务 ID 传递给 `DeleteWirelessDeviceImportTask` API 操作或 `delete-wireless-device-import-task` CLI 命令。

> **Warning**
>
> 删除操作无法撤消。将从您的 AWS 账户中永久删除导入任务。

当您执行 `DeleteWirelessDeviceImportTask` API 请求时，后台进程会开始删除导入任务。当请求正在进行时，导入任务中设备的序列号（SMSN）正处于删除过程中。只有在删除完成后，您才能使

用 ListImportedWirelessDeviceTasks 或 GetImportedWirelessDeviceTasks API 操作查看此信息。

如果导入任务中仍包含等待登记的设备，则只有在导入任务中的所有设备都已登记或登记失败之后，才会处理 DeleteWirelessDeviceImportTask API 请求。导入任务将在 90 天后过期，一旦任务过期，就可以将其从您的账户中删除。但是，不会删除使用导入任务成功登记的设备。

> **Note**
>
> 如果您尝试使用 DeleteWirelessDeviceImportTask API 请求创建另一个包含待删除设备的序列号的导入任务，那么 StartWirelessDeviceImportTask API 操作将返回错误。

下面显示了 CLI 命令示例：

```
aws iotwireless delete-import-task --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

此命令不会生成任何输出。删除任务后，要验证导入任务是否已从您的账户中删除，可以使用 GetWirelessDeviceImportTask API 操作或 ListWirelessDeviceImportTasks API 操作。

使用 AWS CloudFormation 创建 AWS IoT Wireless 资源

AWS IoT Wireless 与 AWS CloudFormation 集成，后者是一项服务，可帮助您对 AWS 资源进行建模和设置，这样您只需花较少的时间来创建和管理资源与基础设施。您可以创建一个模板来描述所需的所有 AWS 资源，然后 AWS CloudFormation 可为您调配和配置这些资源。

在您使用 AWS CloudFormation 时，可重复使用模板来不断地重复设置 AWS IoT Wireless 资源。描述您的资源一次，然后在多个 AWS 账户 和区域中反复预调配相同的资源。

AWS IoT Wireless 和 AWS CloudFormation 模板

要为 AWS IoT Wireless 和相关服务设置和配置资源，您必须了解 AWS CloudFormation 模板。模板是 JSON 或 YAML 格式的文本文件。这些模板可描述您要在 AWS CloudFormation 堆栈中调配的资源。如果您不熟悉 JSON 或 YAML，可以在 AWS CloudFormation Designer 的帮助下开始使用 AWS CloudFormation 模板。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的什么是 AWS CloudFormation Designer？。

AWS IoT Wireless 支持在 AWS CloudFormation 中创建无线资源。有关更多信息（包括有关 AWS IoT Wireless 资源的 JSON 和 YAML 模板示例），请参阅《AWS CloudFormation User Guide》中的 AWS IoT Wireless resource type reference。

了解有关 AWS CloudFormation 的更多信息

要了解有关 AWS CloudFormation 的更多信息，请参阅以下资源：

- AWS CloudFormation
- AWS CloudFormation 用户指南
- AWS CloudFormation 命令行界面用户指南

AWS IoT Wireless 配额

对于每个 AWS 服务，您的 AWS 账户都具有默认配额（以前称为“限额”）。除非另有说明，否则，每个配额都特定于区域。您可以请求增加某些配额，但其他一些配额无法增加。

要查看 AWS IoT Wireless 配额，请打开 Service Quotas 控制台。在导航窗格中，选择 AWS 服务并选择 AWS IoT Wireless。

要请求提高配额，请参阅《Service Quotas 用户指南》中的请求提高配额。如果配额在服务限额中尚不可用，请使用提高限制表格。

AWS IoT Wireless 具有以下配额：

- 适用于在设备之间传输的设备数据的 适用于 LoRaWAN 的 AWS IoT Core 配额
- 适用于 LoRaWAN 和 SideWalk 设备的 AWS IoT Wireless API 操作。

有关更多信息，请参阅《AWS General Reference》中的 适用于 LoRaWAN 的 AWS IoT Core quotas。

标记您的 AWS IoT Wireless 资源

为了帮助您管理和组织设备、网关、目标和配置文件，您可以选择将自己的元数据以标签的形式分配给其中每个资源。本节介绍标签并说明如何创建标签。AWS IoT Wireless 没有账单组，使用的是与 AWS IoT Core 相同的账单组。有关更多信息，请参阅 AWS IoT Core 文档 中的 Billing groups。

标签基本知识

如果您具有相同类型的多种 AWS IoT Wireless 资源，则可以使用标签按照不同方式（例如按用途、所有者或环境）对这些资源进行分类。这将帮助您根据分配给资源的标签快速识别该资源。

每个标签都包含您定义的一个键和一个可选值。例如，您可以针对要更新设备固件的一组 LoRaWAN 设备定义一组标签。为了更轻松地管理您的资源，建议您创建一组一致的标签键，以满足您对每种资源的需求。

您可以根据添加或应用的标签搜索和筛选资源。您还可以使用标签通过 IAM 策略来控制对资源的访问，以及使用账单组标签对费用进行分类和跟踪。

创建和管理标签

您可以在 AWS Management Console、AWS IoT Wireless、或 AWS CLI 中使用标签编辑器创建和管理标签。

使用控制台

为便于使用，AWS Management Console中的标签编辑器提供了一种用于创建和管理标签的集中而统一的方法。有关更多信息，请参阅使用 AWS Management Console中的使用标签编辑器。

使用 API 或 CLI

您还可以使用 API 或 CLI，在创建标签时使用以下命令中的 `Tags` 字段将标签与无线设备、网关、配置文件和目标相关联：

- AssociateAwsAccountWithPartnerAccount
- CreateDestination
- CreateDeviceProfile

- CreateFuotaTask
- CreateMulticastGroup
- CreateServiceProfile
- CreateWirelessGateway
- CreateWirelessGatewayTaskDefinition
- CreateWirelessDevice
- API_StartBulkAssociateWirelessDeviceWithMulticastGroup

更新或列出资源标签

您可以使用以下命令为支持标记的现有资源添加、修改或删除标签：

- TagResource
- ListTagsForResource
- UntagResource

您可以修改标签的密钥和值，还可以随时删除资源的标签。您可以将标签的值设为空的字符串，但是不能将其设为空值。如果添加的标签的键与该资源上现有标签的键相同，新值就会覆盖旧值。如果删除资源，则所有与资源相关的标签都将被删除。

标签限制

下面是适用于标签的基本限制：

- 每个资源的最大标签数 - 50
- 最大键长度 - 127 个 Unicode 字符（采用 UTF-8 格式）
- 最大值长度 - 255 个 Unicode 字符（采用 UTF-8 格式）
- 标签键和值区分大小写。
- 请勿在标签名称或值中使用 `aws:` 前缀。它保留供 AWS 使用。您无法编辑或删除带此前缀的标签名称或值。具有此前缀的标签不计入每个资源的标签数限制。
- 如果在多个服务和资源中使用您的标记方案，请记住，其他服务可能对允许使用的字符有限制。允许使用的字符包括：可用 UTF-8 格式表示的字母、空格和数字以及以下特殊字符：+ - = . _ : / @。

在 IAM 策略中使用标签

要指定用户能够创建、修改或使用的资源，您可以在用于执行 AWS IoT Wireless API 操作的 IAM 策略中应用基于标签的资源级权限。要根据资源标签控制用户访问（权限），请在 IAM 策略中将 Condition 元素（也称为 Condition 数据块）与以下条件上下文键和值结合使用。

- 使用 `aws:ResourceTag/tag-key: tag-value` 可允许或拒绝带特定标签的资源上的用户操作。
- 使用 `aws:RequestTag/tag-key: tag-value` 可要求在发出创建或修改允许标签的资源的 API 请求时使用（或不使用）特定标签。
- 使用 `aws:TagKeys: [tag-key, ...]` 可要求在发出创建或修改允许标签的资源的 API 请求时使用（或不使用）一组特定标签键。

> **Note**
>
> IAM 策略中的条件上下文键和值仅适用于能够标记的资源的标识符是必需参数的那些 AWS IoT 操作。例如，不会根据条件上下文键/值允许/拒绝使用 DescribeEndpoint，因为在该请求中没有引用任何可标记的资源。

有关使用标签的更多信息，请参阅《AWS Identity and Access Management 用户指南》中的使用标签控制访问权限。该指南的 IAM JSON 策略参考一部分包含 IAM 中的 JSON 策略的元素、变量和评估逻辑的详细语法、描述和示例。

以下策略示例应用两个基于标签的限制。受此策略限制的 IAM 用户：

- 无法为资源提供标签“env=prod”（在此示例中，请参阅 `"aws:RequestTag/env" : "prod"` 行
- 无法修改或访问具有现有标签“env=prod”的资源（在此示例中，请参阅 `"aws:ResourceTag/env" : "prod"` 行）。

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

您还可以为给定标签键指定多个标签值，方法是将它们括在列表中，如下所示：

```
"StringEquals" : {
              "aws:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> 如果您基于标签允许或拒绝用户访问资源，则必须考虑显式拒绝用户对相同资源添加或删除这些标签的能力。否则，用户可能通过修改资源标签来绕过您的限制并获得资源访问权限。

AWS IoT Wireless 用户指南的文档历史记录

下表介绍了 AWS IoT Wireless 的文档版本。

变更	说明	日期
初始版本	首次发布 AWS IoT Wireless 用户指南	2020 年 12 月 31 日