aws

開発者ガイド

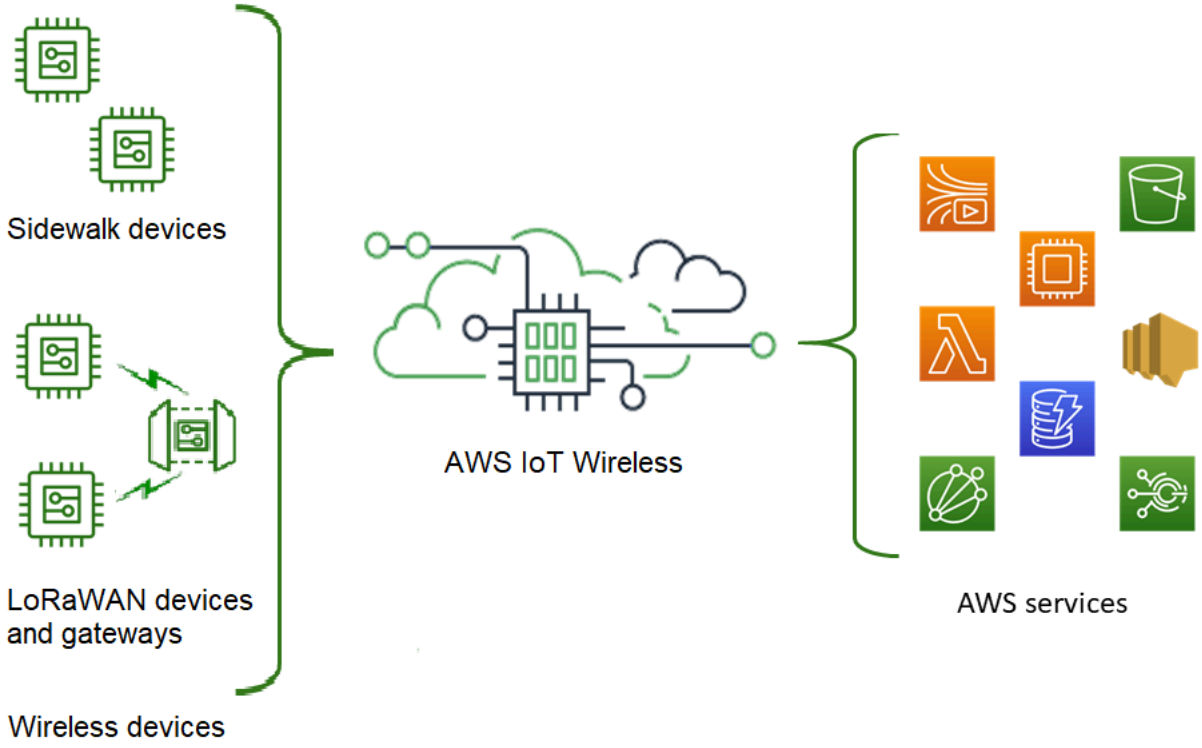
AWS IoT Wireless

AWS IoT Wireless: 開発者ガイド

Table of Contents

AWS IoT Wireless とは

AWS IoT Wireless は、ワイヤレスデバイスを他のデバイスや AWS クラウド サービスに接続するクラウドサービスを提供します。デバイスを AWS IoT Wireless に接続することで、デバイスを AWS IoT ベースのソリューションに統合できます。AWS IoT Wireless を使用して、LoRaWAN と Sidewalk デバイスを両方オンボードできます。これらのワイヤレスデバイスは、低電力広域ネットワーク (LPWAN) 通信プロトコルを使用して AWS IoT と通信します。



AWS IoT Wireless の機能

AWS IoT Wireless には次の機能があります。

LoRaWAN デバイスと Sidewalk デバイスをオンボードする

LoRaWAN と Sidewalk デバイスを両方 AWS IoT Wireless にオンボードできます。

- AWS IoT Core for LoRaWAN

  AWS IoT Wireless に LoRaWAN デバイスとゲートウェイをオンボードするには、AWS IoT Core for LoRaWAN を使用します。これは、プライベート LNS をセットアップし、運用する必要のないフルマネージド型の LoRaWAN Network Server (LNS) です。AWS IoT Core for LoRaWAN

は、Configuration and Update Server (CUPS) および Firmware Updates Over-The-Air (FUOTA) 機能を使用してゲートウェイ管理を提供します。詳細については、「AWS IoT Core for LoRaWAN とは」を参照してください。

- AWS IoT Core for Amazon Sidewalk

  Sidewalk デバイスを AWS IoT Wireless にオンボードするには、Amazon Sidewalk の AWS IoT Core が提供する機能を使用できます。Amazon Sidewalk は、Amazon Echo、Ring セキュリティカメラ、イメージングライトなどのデバイスを接続する共有ネットワークであり、コミュニティ内の他の Sidewalk デバイスをサポートできます。詳細については、「AWS IoT Core for Amazon Sidewalk の概要」を参照してください。

AWS IoT Core との統合

AWS IoT Wireless と AWS IoT Core の統合によって提供される次の機能を使用できます。

- デバイスを AWS IoT のモノに関連付ける

  ワイヤレスデバイスとゲートウェイを AWS IoT の モノ と関連付けると、デバイスの表現をクラウドに簡単に保存できます。AWS IoT でモノを使用すると、デバイスの検索や管理、その他の AWS IoT Core 機能へのアクセスがより簡単になります。詳細については、「AWS IoT Core デベロッパーガイド 」の「AWS IoT でデバイスを管理する」を参照してください。

- AWS IoT ルールを使用してメッセージをルーティングする

  AWS IoT のルール機能を使用して、他の AWS のサービス やアプリケーションとやり取りできます。デバイスからクラウドに送信されるアップリンクメッセージは、これらのサービスやその他のアプリケーションにルーティングできます。詳細については、AWS IoT Core デベロッパーガイドの「AWS IoT のルール」を参照してください。

AWS IoT Wireless を初めて使用する場合

AWS IoT Wireless を初めて使用する方には、以下のセクションを初めに読むことをお勧めします。

- AWS IoT Core for LoRaWAN とは

  このセクションでは、LoRaWAN テクノロジーの概要と AWS IoT Core for LoRaWAN の仕組みについて説明します。また、詳細を理解するのに役立つリソースも提供します。

- AWS IoT Core for Amazon Sidewalk の概要

このセクションでは、Amazon Sidewalk テクノロジーの概要と AWS IoT Core for Amazon Sidewalk の仕組みについて説明します。また、詳細を理解するのに役立つリソースも提供します。

- AWS IoT Core for Amazon Sidewalk の開始方法

  AWS IoT Core for Amazon Sidewalk で を使用する方法と Amazon Sidewalk デバイスをオンボードする方法については、このセクションをお読みください。

- AWS IoT Core for LoRaWAN へのゲートウェイとデバイスの接続

  次に、 コンソールと API を使用して LoRaWAN デバイスをオンボードする方法の詳細について説明します。

関連サービス

- 「Amazon CloudWatch」

  LoRaWAN または Sidewalk デバイスを AWS IoT Wireless にオンボードした後、Amazon CloudWatch を使用してワイヤレスデバイスとゲートウェイをリアルタイムでログに記録し、モニタリングできます。LoRaWAN デバイスとゲートウェイをモニタリングするには、ネットワークアナライザーを使用することもできます。これにより、接続のセットアップとトレースメッセージの受信開始にかかる時間が短縮されます。

- AWS IoT Core

  AWS IoT Core 統合を使用して、ルールエンジンからアクセスできる AWS のサービス に接続することもできます。詳細については、「ルールエンジンで使用される AWS のサービス」を参照してください。

AWS IoT Wireless へのアクセス

コンソール、API、または CLI を使用して、LoRaWAN デバイスと Sidewalk デバイスの両方をオンボードできます。

- AWS IoT コンソールを使用する場合

  ワイヤレスデバイスをオンボードするには、AWS Management Console の AWS IoT Wireless ページを使用します。

- AWS IoT Wireless API を使用する場合

  AWS IoT Wireless を使用して、Sidewalk デバイスと LoRaWAN デバイスの両方をオンボードできます。AWS IoT Core が構築されている AWS IoT Wireless API は、AWS SDK でサポートされています。詳細については、AWS SDK およびツールキットを参照してください。

- AWS CLI の使用

  AWS CLI を使用して、LoRaWAN と Amazon Sidewalk デバイスをオンボードおよび管理するためのコマンドを実行します。詳細については、「AWS IoT Wireless CLI リファレンス」を参照してください。

AWS IoT Wireless の開始方法

AWS アカウント にサインアップし、手順に従って IAM ユーザーを作成することで、AWS IoT Wireless の使用を開始できます。サインアップしたら、AWS Management Console、AWS IoT Wireless API、または AWS CLI を使用して Sidewalk および LoRaWAN デバイスとゲートウェイをオンボーディングできます。デバイスをオンボーディングするときは、リソースをより簡単に識別できるように、リソースを記述してタグ付けする方法を検討してください。

以下のトピックでは、AWS IoT Wireless の使用を開始する方法について説明します。

トピック

- AWS IoT Wireless のセットアップ
- AWS IoT Wireless リソースについて説明する

AWS IoT Wireless のセットアップ

AWS にサインアップすると、AWS アカウント は、AWS IoT Wireless を含めた AWS 内のすべてのサービスに自動的にサインアップされます。料金は、使用するサービスの料金のみが請求されます。

AWS IoT Wireless をセットアップするには、次のセクションのステップを実行します。

トピック

- AWS アカウントのセットアップ
- Python および AWS CLI のインストール

AWS アカウントのセットアップ

Amazon Sidewalk に AWS IoT Core for LoRaWAN または AWS IoT Core を初めて使用する場合は、AWS アカウント をセットアップするのに以下のタスクを実行してください。

トピック

- AWS アカウントへのサインアップ
- IAM ユーザーの作成
- IAM ユーザーとしてのサインイン

AWS アカウントへのサインアップ

AWS アカウント をお持ちでない場合は、以下の手順を実行してアカウントを作成してください。

AWS アカウント にサインアップするには

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一環として、通話呼び出しを受け取り、電話のキーパッドを使用して検証コードを入力するように求められます。

   AWS アカウントにサインアップすると、AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービスとリソースへのアクセス権があります。セキュリティのベストプラクティスとして、管理ユーザーに管理アクセスを割り当て、ルートユーザーアクセスが必要なタスクを実行する場合にのみ、ルートユーザーを使用してください。

IAM ユーザーの作成

管理者ユーザーを作成するには、以下のいずれかのオプションを選択します。

管理者を管理する方法を 1 つ選択します	目的	方法	以下の操作も可能
IAM Identity Center 内 (推奨)	短期の認証情報を使用して AWS にアクセスします。 これはセキュリティのベストプラクティスと一致しています。ベストプラクティスの詳細については、IAM	「AWS IAM Identity Center ユーザーガイド」の「開始方法」の手順に従います。	「AWS Command Line Interface ユーザーガイド」の「AWS IAM Identity Center を使用するための AWS CLI の設定」に従って、プログラムによるアクセスを設定します。

管理者を管理する方法を 1 つ選択します	目的	方法	以下の操作も可能
	ユーザーガイドの「IAM でのセキュリティのベストプラクティス」を参照してください。		
IAM 内 (非推奨)	長期認証情報を使用して AWS にアクセスする。	「IAM ユーザーガイド」の「最初の IAM 管理者のユーザーおよびグループの作成」の手順に従います。	IAM ユーザーガイドの「IAM ユーザーのアクセスキーの管理」に従って、プログラムによるアクセスを設定します。

IAM ユーザーとしてのサインイン

IAM ユーザーを作成したら、IAM ユーザー名とパスワードを使用して AWS にサインインできます。

IAM ユーザーとしてサインインする前に、IAM コンソールで IAM ユーザーのサインインのリンクを確認できます。IAM ダッシュボードの IAM ユーザーのサインインのリンクの下に、AWS アカウント のサインインのリンクが表示されます。サインインリンクの URL には、ダッシュ (-) を含まない AWS アカウント ID が含まれています。

サインインリンクの URL に AWS アカウント ID を含めない場合は、アカウントのエイリアスを作成します。詳細については、IAM ユーザーガイドの AWS アカウント のエイリアスの作成、削除および一覧表示を参照してください。

IAM ユーザーとしてサインインするには

1. からサインアウトします AWS Management Console
2. AWS アカウント ID (ダッシュを除く) または AWS アカウント エイリアスを含めたサインインリンクを入力します。

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

3. 作成した IAM ユーザー名とパスワードを入力します。

サインインすると、ナビゲーションバーに「*your_user_name @ your_aws_account_id*」が表示されます。

Python および AWS CLI のインストール

LoRaWAN または Sidewalk エンドデバイスを接続する前に、インストール Python をセットアップし、AWS CLI を設定する必要があります。

> **Important**
>
> Sidewalk エンドデバイスのプロビジョニングと登録に関するオンボーディングワークフロー全体を実行するには、Sidewalk ゲートウェイと HDK をセットアップする必要もあります。手順については、「Amazon Sidewalk ドキュメント」の「Hardware Development Kit (HDK) のセットアップ」と「Sidewalk ゲートウェイのセットアップ」を参照してください。

トピック

- Python および Python3-pip のインストール
- AWS CLI のセットアップ

Python および Python3-pip のインストール

次のセクションで説明するように、AWS CLI と boto3 を使用するには、Python バージョン 3.6 以降が必要です。AWS IoT コンソールを使用してエンドデバイスをオンボードする場合は、このセクションを飛ばして AWS アカウント のセットアップを続行できます。Python と Python3-pip が既にインストールされているかどうかを確認するには、次のコマンドを実行します。これらのコマンドを実行してバージョンが返される場合は、Python と Python3-pip が正しくインストールされています。

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

このコマンドでエラーが返される場合は、Python がインストールされていないか、オペレーティングシステムが Python v3.x 実行可能ファイルを Python3 として呼び出していることが原因であ

る可能性があります。その場合は、コマンドを実行するときに python のすべてのインスタンスを python3 に置き換えてください。それでもエラーが発生する場合は、Python インストーラをダウンロードして実行するか、以下で説明するように、ご使用のオペレーティングシステムに応じて Python をインストールしてください。

Windows

Windows マシンで、Python のウェブサイトから Python をダウンロードし、インストーラーを実行して Python をマシンにインストールします。

Linux

Ubuntu マシンで、以下の sudo コマンドを実行して Python をインストールします。

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

macOS

Mac マシンで、Homebrew を使用して Python をインストールします。Homebrew では pip もインストールされます。その後、pip はインストールされた Python3 バージョンを指定します。

```
$ brew install python
```

AWS CLI のセットアップ

以下の手順は、AWS CLI および boto3 (AWS SDK for Python) を設定する方法を示しています。これらのステップを実施する前に、AWS アカウント にサインアップし、管理ユーザーを作成しておく必要があります。手順については、AWS IoT Wireless のセットアップ を参照してください。

1. AWS CLI のインストールと設定

   AWS CLI を使用すると、Sidewalk エンドデバイスを AWS IoT Core for Amazon Sidewalk にプログラム的にオンボードできます。AWS IoT コンソールを使用してデバイスをオンボードする場合は、このセクションを省略できます。AWS IoT Core コンソールを開いて次のセクションに進み、デバイスの AWS IoT Core for Amazon Sidewalk への接続を開始してください。AWS CLI の設定方法については、「AWS CLI のインストールと設定」を参照してください。

2. boto3 (AWS SDK for Python) のインストール

   次のコマンドは、boto3 (AWS SDK for Python) と AWS CLI のインストール方法を示しています。boto3 を実行するために必要な botocore もインストールする必要があります。詳細な手順については、「Boto3 ドキュメントガイド」の「Boto3 のインストール」を参照してください。

   > **Note**
   >
   > awscli バージョン 1.26.6 には、PyYAML バージョンの 3.10 以降、5.5 以前が必要です。

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

3. 認証情報およびデフォルトのリージョンの設定

   ~/.aws/credentials および ~/.aws/config ファイル内で、認証情報およびデフォルトのリージョンを設定します。boto3 ライブラリは、これらの認証情報を使用して AWS アカウントを識別し、API コールを承認します。設定手順については、以下を参照してください。

   - 「設定」(Boto3 ドキュメントガイド)
   - 「設定ファイルと認証情報ファイルの設定」(AWS CLI ドキュメントガイド)

AWS IoT Wireless リソースについて説明する

LoRaWAN または Sidewalk デバイスのオンボーディングを開始する前に、デバイス、ゲートウェイ、および送信先の命名規則を考慮してください。AWS IoT Wireless では、作成するリソースを識別するためのいくつかのオプションが提供されます。AWS IoT Wireless リソースには作成時に一意の ID が与えられますが、この ID は説明的なものではなく、リソースの作成後に変更することもできません。リソースの選択、識別、および管理をより便利にするため、ほとんどの AWS IoT Wireless リソースに名前を割り当てたり、説明を追加したり、タグとタグ値をアタッチできます。

- リソース名と説明

  デバイス、ゲートウェイ、およびプロファイルの場合、リソース名は、リソースの作成後に変更できるオプションのフィールドです。名前が、リソースハブページに表示されるリストに表示されます。

送信先については、AWS アカウントと AWS リージョン 一意の名前を指定します。送信先リソースの作成後は、送信先名を変更できません。

名前には最大 256 文字を使用できますが、リソースハブの表示スペースには制限があります。可能な場合は、名前の識別部分が最初の 20～30 文字で表示されることを確認してください。

- リソースタグ

  タグは、AWS リソースにアタッチできるメタデータのキーと値のペアです。タグキーと対応する値の両方を選択します。

  ゲートウェイ、送信先、およびプロファイルには、最大 50 個のタグをアタッチできます。デバイスがタグをサポートしていません。

リソース名と説明

AWS IoT Wireless リソースの名前のサポート

リソース	名前フィールドのサポート
送信先	Name はリソースの一意の ID であり、変更できません。
ワイヤレスデバイス	Name はオプションのリソースの記述子であり、変更できます。
LoRaWAN ゲートウェイ	Name はオプションのリソースの記述子であり、変更できます。
プロフィール	Name はオプションのリソースの記述子であり、変更できます。

名前フィールドは、リソースのリソースハブリストに表示されます。ただし、スペースは限られているため、名前の最初の 15～30 文字しか表示されない場合があります。リソースの名前を選択すると

きは、それらがどのようにリソースを識別するようにしたいか、コンソールでどのように表示されるかを考慮します。

説明

送信先、デバイス、およびゲートウェイの各リソースは、最大 2,048 文字を入力できる説明フィールドもサポートします。説明フィールドは、個々のリソースの詳細ページにのみ表示されます。説明フィールドはリソースの詳細ページにのみ表示されるため、多くの情報を保持できますが、複数のリソースのコンテキストでスキャンするには不便です。

リソースタグ

AWS IoT Wireless リソースの AWS タグのサポート

リソース	AWS タグのサポート
送信先	最大 50 個の AWS タグをリソースに追加できます。
ワイヤレスデバイス	このリソースは AWS タグをサポートしていません。
LoRaWAN ゲートウェイ	最大 50 個の AWS タグをリソースに追加できます。
プロフィール	最大 50 個の AWS タグをリソースに追加できます。

タグは、AWS リソースを特定し、整理するのに使用できる単語または語句であり、メタデータとして機能します。タグキーは情報のカテゴリと考えることができ、タグ値はそのカテゴリの特定の値と考えることができます。例えば、色のタグ値があり、一部のリソースにそのタグ用に青の値を与え、他のリソースに赤の値を与えることができます。これで、AWS コンソールのタグエディタを使用して、青の色タグ値を持つリソースを見つけることができます。

AWS IoT Wireless でのタグ付けの詳細については、「AWS IoT Wireless リソースのタグ付け」を参照してください。

タグ付けおよびタグ付け戦略の詳細については、タグエディタを参照してください。

AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWAN は、Configuration and Update Server (CUPS) および Firmware Updates Over-The-Air (FUOTA) 機能を使用してゲートウェイ管理を提供する、フルマネージド型の LoRaWAN Network Server (LNS) です。プライベート LNS を AWS IoT Core for LoRaWAN に置き換えて、Long Range Wide Area Network (LoRaWAN) デバイスおよびゲートウェイを AWS IoT Core に接続できます。そうすることで、メンテナンス、運用コスト、セットアップ時間、およびオーバーヘッドコストを削減できます。

> **Note**
>
> AWS IoT Core for LoRaWAN は、IPv4 アドレス形式のみをサポートしています。IPv6 またはデュアルスタック設定 (IPv4 と IPv6) のサポートはありません。詳細については、「IPv6 をサポートする AWS のサービス」を参照してください。

序章

LoRaWAN デバイスは、LoRaWAN プロトコルを使用してライセンスフリーの無線スペクトルで動作する、長距離、低電力、バッテリー駆動のデバイスです。LoRaWAN は LoRa 上に構築された Low Power Wide Area Network (LPWAN) 通信プロトコルです。LoRa は、デバイス間の低消費電力、広域通信を可能にする物理層プロトコルです。

LoRaWAN デバイスを AWS IoT に接続するには、LoRaWAN ゲートウェイを使用する必要があります。ゲートウェイは、デバイスを AWS IoT Core for LoRaWAN に接続してメッセージを交換するためのブリッジとして機能します。AWS IoT Core for LoRaWAN は AWS IoT ルールエンジンを使用して LoRaWAN デバイスから他の AWS IoT のサービスにメッセージをルーティングします。

開発労力を削減し、デバイスを AWS IoT Core for LoRaWAN に迅速にオンボードするために、LoRaWAN 認定エンドデバイスを使用することをお勧めします。詳細については、「AWS IoT Core for LoRaWAN 製品ページ」を参照してください。デバイスの LoRaWAN 認証取得については、「Certifying LoRaWAN products」を参照してください。

AWS IoT Core for LoRaWAN へのアクセス

コンソールまたは AWS IoT Wireless API を使用すると、LoRaWAN デバイスやゲートウェイを素早く AWS IoT Core for LoRaWAN にオンボードできます。

コンソールを使用する場合

AWS Management Console を使用して LoRaWAN デバイスおよびゲートウェイをオンボードするには、AWS Management Console にサインインして AWS IoT コンソールの [AWS IoT Core for LoRaWAN] ページに移動します。次に、[概要] セクションを使用して、ゲートウェイとデバイスを AWS IoT Core for LoRaWAN に追加します。詳細については、「コンソールを使用してデバイスとゲートウェイを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。

API または CLI の使用

AWS IoT Wireless API を使用して、Sidewalk デバイスと LoRaWAN デバイスの両方をオンボードできます。AWS IoT Core for LoRaWAN が構築されている AWS IoT Wireless API は、AWS SDK でサポートされています。詳細については、AWS SDK およびツールキットを参照してください。

AWS CLI を使用してコマンドを実行し、LoRaWAN ゲートウェイおよびデバイスをオンボードおよび管理することができます。詳細については、「AWS IoT Wireless CLI リファレンス」を参照してください。

AWS IoT Core for LoRaWAN のリージョンとエンドポイント

AWS IoT Core for LoRaWAN では、AWS リージョン に固有のコントロールプレーンとデータプレーン API エンドポイントのサポートが提供されます。データプレーン API エンドポイントは、AWS アカウント と AWS リージョン リージョンに固有です。AWS IoT Core for LoRaWAN エンドポイントの詳細については、AWS 全般のリファレンスの「AWS IoT Core for LoRaWAN エンドポイント」を参照してください。

デバイスと AWS IoT の間のより安全な通信を実現するため、パブリックインターネットを介さず、V仮想プライベートクラウド (VPC) で AWS PrivateLink を介して AWS IoT Core for LoRaWAN にデバイスを接続できます。詳細については、「AWS IoT Core for LoRaWAN とインターフェース VPC エンドポイント (AWS PrivateLink)」を参照してください。

AWS IoT Core for LoRaWAN には、デバイス間で送信されるデバイスデータと、AWS IoT Wireless API オペレーション の最大 TPS に適用されるクォータがあります。詳細については、「AWS 全般のリファレンス」の「AWS IoT Core for LoRaWAN クォータ」を参照してください。

AWS IoT Core for LoRaWAN の料金

新しいカスタマーで、AWS にサインアップすると、AWS 無料利用枠を利用して、AWS IoT Core for LoRaWAN を無料で使い始めることができます。AWS IoT Core for LoRaWAN では、実際に使用し

た分に対してのみお支払いいただきます。全般的な製品の概要と料金の詳細については、「AWS IoT Core の料金」を参照してください。

AWS IoT Core for LoRaWAN とは

AWS IoT Core for LoRaWAN では、LoRaWAN デバイスおよびゲートウェイを AWS に接続することで、プライベート LoRaWAN Network Server (LNS) を置き換えます。AWS IoT ルールエンジンを使用して、LoRaWAN デバイスから受信したメッセージをルーティングできます。メッセージをフォーマットして、他の AWS IoT サービスに送信できます。AWS IoT とのデバイス通信のセキュリティを確保するために、AWS IoT Core for LoRaWAN では X.509 証明書を使用します。

AWS IoT Core for LoRaWAN は、AWS IoT Core が LoRaWAN ゲートウェイやデバイスと通信するために必要なサービスポリシーとデバイスポリシーを管理します。また、AWS IoT Core for LoRaWAN は、デバイスデータを他のサービスに送信する AWS IoT ルールを記述する送信先も管理します。

AWS IoT Core for LoRaWAN の機能

AWS IoT Core for LoRaWAN を使用すると、次のことが可能になります。

- LoRaWAN デバイスおよびゲートウェイを AWS IoT にオンボードして接続する。プライベート LNS をセットアップおよび管理する必要はありません。
- LoRa Alliance によって標準化された 1.0.x または 1.1 LoRaWAN 仕様に準拠する LoRaWAN デバイスを接続する。これらのデバイスは、クラス A、クラス B、またはクラス C モードで動作できます。
- LoRa Basics Station バージョン 2.0.4 以降をサポートする LoRaWAN ゲートウェイを使用します。AWS IoT Core for LoRaWAN に認定されているすべてのゲートウェイは、互換性のあるバージョンの LoRa Basics Station を実行します。
- 公開されている LoRaWAN ネットワークを使用して LoRaWAN デバイスをクラウドに接続すると、デプロイまでの時間が短縮され、プライベート LoRaWAN ネットワークを管理する必要がなくなり、時間とコストが削減されます。
- AWS IoT Core for LoRaWAN の適応データレートを使用して、信号強度、帯域幅、および拡散係数をモニタリングします。そして、必要に応じてデータレートを最適化します。また、ネットワークアナライザを使用すると、LoRaWAN リソースをリアルタイムでモニタリングできます。
- CUPS サービスを使用して LoRaWAN ゲートウェイのファームウェアを更新し、Firmware Updates Over-The-Air (FUOTA) を使用して LoRaWAN デバイスのファームウェアを更新します。

以下のトピックでは、LoRaWAN 技術と AWS IoT Core for LoRaWAN についての詳細情報を示します。

トピック

LoRaWAN とは

LoRa Alliance は、LoRaWAN について、「地域、国、またはグローバルネットワークにおいて、バッテリー駆動の「モノ」をインターネットにワイヤレスで接続するように設計された Low Power Wide Area (LPWA) ネットワーキングプロトコルであり、双方向通信、エンドツーエンドのセキュリティ、モビリティ、ローカリゼーションサービスなどの主要なモノのインターネット (IoT) の要件を満たすことを目標としています」と述べています。

LoRa と LoRaWAN

LoRaWAN プロトコルは、LoRa 上で機能する省電力広域ネットワーク (LPWAN) 通信プロトコルです。

LoRaWAN は、省電力広域ネットワークの国際標準として認識されています。詳細については、「正式に ITU 国際標準として認識されている LoRAWAN」を参照してください。LoRaWAN 仕様はオープンなので、誰でも LoRa ネットワークを設定して運用できます。

LoRa は、ライセンスフリーの無線周波数スペクトルで動作する無線オーディオ周波数技術です。LoRa は、スペクトル拡散変調を使用する物理層プロトコルで、帯域幅を犠牲にして長距離通信をサポートします。中央の周波数を持つ狭帯域波形を使用してデータを送信するため、干渉に対して堅牢になります。

LoRaWAN 技術の特徴

- 直線距離で最大 10 マイルまでの長距離通信が可能です。
- 最長 10 年間という長いバッテリー持続時間があります。バッテリー寿命を延ばすには、デバイスをクラス A またはクラス B モードで動作させます。このモードでは、ダウンリンクのレイテンシーが長くなります。
- デバイスおよびメンテナンスのコストが低く抑えられます。
- ライセンスフリーの無線スペクトラムですが、リージョン固有の規制が適用されます。

- 低消費電力ですが、データレートに応じてペイロードサイズが 51 バイトから 241 バイトまでに制限されます。データレートは、最大の 222 ペイロードサイズで 0.3 Kbit/秒 - 27 Kbit/秒にすることができます。

LoRaWAN プロトコルバージョン

LoRa Alliance は、LoRaWAN 仕様ドキュメントを使用して LoRaWAN プロトコルを指定します。リージョン固有の規制を考慮して、LoRa Alliance はリージョンパラメータドキュメントも公開しています。詳細については、「LoRaWAN リージョンパラメータと仕様」を参照してください。

LoRaWAN の初期リリースはバージョン 1.0 です。リリースされた追加バージョンは、1.0.1、1.0.2、1.0.3、1.0.4、および 1.1 です。バージョン 1.0.1 ~ 1.0.4 は、一般的に 1.0.x と呼ばれます。

LoRaWAN の詳細

以下のリンクには、LoRaWAN テクノロジーと LoRa Basics Station に関する有益な情報が含まれています。LoRa Basics Station は、LoRaWAN用にエンドデバイスを AWS IoT Core for LoRaWAN に接続するための LoRaWAN ゲートウェイで動作するソフトウェアです。

- ITU 国際標準として認識されている LoRaWAN

  LoRaWAN は、ITU により、省電力広域ネットワークの国際標準として正式に文書化されています。この標準は、推奨事項 ITU-T Y.4480「広域ワイヤレスネットワーク用の低電力プロトコル」というタイトルになっています。

- LoRaWAN のモノの基礎

  LoRaWAN のモノの基礎には、LoRaWAN の基礎を説明する紹介ビデオと、LoRa および LoRaWAN について学ぶのに役立つ一連の章が含まれています。

- LoRaWAN とは

  LoRa Alliance は、さまざまなリージョンの LoRaWAN 仕様のまとめなど、LoRa および LoRaWAN の技術的な概要を提供しています。

- LoRa Basics Station

  Semtech 社は、ゲートウェイとエンドノードの LoRa Basics に関する有用な概念を提供します。LoRa Basics Station は、LoRaWAN ゲートウェイ上で動作するオープンソースソフトウェアであり、Semtech 社の GitHub リポジトリを介して管理および配信されています。ま

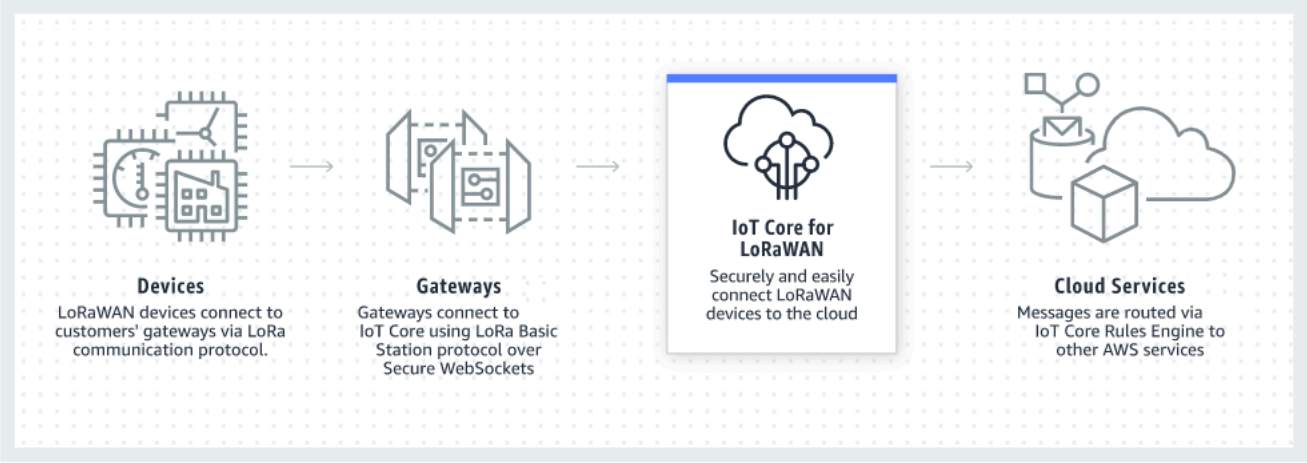
た、LoRaWAN データの交換方法や設定の更新方法について説明した LNS および CUPS プロトコルについても学習できます。

- LoRaWAN リージョンのパラメータと仕様

  RP002-1.0.2 ドキュメントには、LoRaWAN Layer 2 仕様のすべてのバージョンのサポートが含まれています。これには、LoRaWAN 仕様とリージョンパラメータ、およびさまざまな LoRaWAN バージョンに関する情報が含まれています。

AWS IoT Core for LoRaWAN の働き

LoRaWAN ネットワークアーキテクチャは、ゲートウェイがエンドデバイスと LoRaWAN ネットワークサーバー (LNS) 間で情報を中継する「star of stars」トポロジーでデプロイされています。以下に、LoRaWAN デバイスが AWS IoT Core for LoRaWAN とやりとりする方法を示します。また、AWS IoT Core for LoRaWAN が LNSとして機能し、AWS クラウド で他の AWS のサービス と通信する方法も示します。



LoRaWAN デバイスは LoRaWAN ゲートウェイを介して AWS IoT Core と通信します。AWS IoT Core for LoRaWAN は、AWS IoT Core が LoRaWAN ゲートウェイやデバイスを管理し、それらと通信するために必要なサービスポリシーとデバイスポリシーを管理します。また、AWS IoT Core for LoRaWAN は、デバイスデータを他のサービスに送信する AWS IoT ルールを記述する送信先も管理します。

AWS IoT Core for LoRaWAN の使用を開始する

次の手順は、AWS IoT Core for LoRaWAN の使用を開始する方法の概要を示しています。

1. 必要なワイヤレスデバイスと LoRaWAN ゲートウェイを選択する。

AWS Partner Device Catalog には、AWS IoT Core for LoRaWAN での使用が認定されているゲートウェイとデベロッパーキットが含まれています。詳細については、「AWS Partner Device Catalog の認定されたゲートウェイの使用」を参照してください。

2. ワイヤレスデバイスと LoRaWAN ゲートウェイを AWS IoT Core for LoRaWAN に追加します。

   AWS IoT Core for LoRaWAN へのゲートウェイとデバイスの接続 では、リソースの説明方法や、ワイヤレスデバイスと LoRaWAN ゲートウェイを AWS IoT Core for LoRaWAN に追加する方法に関する情報が提供されています。また、これらのデバイスを管理し、データを AWS のサービスに送信するために必要な他の AWS IoT Core for LoRaWAN リソースに設定する方法についても学習します。

3. AWS IoT Core for LoRaWAN ソリューションを完了します。

   サンプルの AWS IoT Core for LoRaWAN ソリューションから始めて、使いやすく調整してください。

AWS IoT Core for LoRaWAN のリソース

以下のリソースは、AWS IoT Core for LoRaWAN の詳細と使用開始方法について説明しています。

- AWS IoT Core for LoRaWAN の開始方法

  次のビデオでは、AWS IoT Core for LoRaWAN の仕組みと、AWS Management Console から LoRaWAN ゲートウェイを追加するプロセスを説明します。

- AWS IoT Core for LoRaWAN ワークショップ

  このワークショップでは、LoRaWAN テクノロジーの基礎と、AWS IoT Core for LoRaWAN での実装を扱います。また、このワークショップを使用して、ゲートウェイとデバイスを AWS IoT Core for LoRaWAN に接続してサンプル IoT ソリューションを構築する方法を詳細に説明します。

- AWS IoT による省電力広域ネットワーク (LPWAN) ソリューションの実装

  このホワイトペーパーでは、LPWAN が IoT ユースケースに適しているかどうかを判断するのに役立つ決定フレームワークを提供し、LPWAN 接続テクノロジーとその機能の概要を示し、実装ガイドラインを提供します。

AWS IoT Core for LoRaWAN へのゲートウェイとデバイスの接続

AWS IoT Core for LoRaWAN は、ワイヤレス LoRaWAN (省電力長距離広域ネットワーク) デバイスの接続と管理を支援し、LNS の開発と運用が不要になります。Long range WAN (LoRaWAN) デバイスおよびゲートウェイは、AWS IoT Core for LoRaWAN を使用することにより、AWS IoT Core に接続できます。

デバイス、ゲートウェイ、プロファイル、および送信先の命名規則

AWS IoT Core for LoRaWAN を開始してリソースを作成する前に、デバイス、ゲートウェイ、および送信先の命名規則を検討してください。

AWS IoT Core for LoRaWAN は、ワイヤレスデバイス、ゲートウェイ、およびプロファイル用に作成するリソースに一意の ID を割り当てます。ただし、リソースによりわかりやすい名前も付けて、リソースを識別しやすくすることができます。AWS IoT Core for LoRaWAN にデバイス、ゲートウェイ、プロファイル、および送信先を追加する前に、管理しやすくするために、それらの名前の付け方を検討してください。

作成したリソースにタグを追加することもできます。LoRaWAN デバイスを追加する前に、AWS IoT Core for LoRaWAN リソースを識別および管理するためのタグの使用方法を検討してください。タグは、追加後に変更できます。

命名とタグ付けの詳細については、AWS IoT Wireless リソースについて説明する を参照してください。

デバイスデータからサービスデータへのマッピング

LoRaWAN ワイヤレスデバイスからのデータは、多くの場合、帯域幅を最適化するためにエンコードされます。これらのエンコードされたメッセージは、他の AWS のサービスで簡単に使用できない可能性がある形式で AWS IoT Core for LoRaWAN に到達します。AWS IoT Core for LoRaWAN は、AWS Lambda 関数を使用して、デバイスメッセージを処理して他の AWS のサービスが使用できる形式にデコードできる AWS IoT ルールを使用します。

デバイスデータを変換して他の AWS のサービスに送信するには、次のことを理解しておく必要があります。

- ワイヤレスデバイスが送信するデータの形式と内容。
- データの送信先とするサービス。

- サービスが必要とする形式。

この情報を使用して、変換を実行する AWS IoT ルールを作成し、変換されたデータを、それを使用する AWS のサービスに送信できます。

コンソールを使用してデバイスとゲートウェイを AWS IoT Core for LoRaWAN にオンボードする

コンソールインターフェイスまたは API を使用して、LoRaWAN ゲートウェイとデバイスを追加できます。初めて AWS IoT Core for LoRaWAN を使用する場合は、コンソールを使用することをお勧めします。いくつかの AWS IoT Core for LoRaWAN リソースを一度に管理する場合には、コンソールインターフェイスが最も実用的です。大量の AWS IoT Core for LoRaWAN リソースを管理する場合は、AWS IoT Wireless API を使用して、より自動化されたソリューションを作成することを検討してください。

AWS IoT Core for LoRaWAN リソースを設定するときに入力するデータの多くは、デバイスのベンダーによって提供され、ベンダーがサポートしている LoRaWAN 仕様に固有のものです。以下のトピックでは、AWS IoT Core for LoRaWAN リソースを説明し、コンソールまたは API を使用してゲートウェイとデバイスを追加する方法を説明します。

> **Note**
>
> パブリックネットワークを使用して LoRaWAN デバイスをクラウドに接続している場合は、ゲートウェイのオンボーディングをスキップできます。詳細については、「パブリック LoRaWAN デバイスネットワーク (Everynet) からの LoRaWAN トラフィックの管理」を参照してください。

トピック

- ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする
- デバイスを AWS IoT Core for LoRaWAN にオンボードする

ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする

AWS IoT Core for LoRaWAN を初めて使用する場合は、コンソールを使用することで、最初の LoRaWAN ゲートウェイとデバイスを追加できます。

> **Note**
>
> パブリックネットワークを使用して LoRaWAN デバイスをクラウドに接続している場合は、ゲートウェイのオンボーディングをスキップできます。詳細については、「パブリック LoRaWAN デバイスネットワーク (Everynet) からの LoRaWAN トラフィックの管理」を参照してください。

ゲートウェイをオンボードする前に

ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする前に、以下のことをお勧めします。

- AWS IoT Core for LoRaWAN での使用を認定されたゲートウェイを使用してください。これらのゲートウェイは追加の構成設定なしで AWS IoT Core に接続し、バージョン 2.0.4 以降の LoRa Basics Station ソフトウェアが実行されています。詳細については、「AWS IoT Wireless によるゲートウェイの管理」を参照してください。
- リソースをより簡単に管理できるように、作成したリソースの命名規則を考慮してください。詳細については、「AWS IoT Wireless リソースについて説明する」を参照してください。
- 各ゲートウェイに固有の設定パラメータの入力準備を事前にしておくと、コンソールへのデータの入力がよりスムーズになります。AWS IoT がゲートウェイと通信および管理するために必要なワイヤレスゲートウェイの設定パラメータには、ゲートウェイの EUI とその LoRa 周波数帯域が含まれます。

ゲートウェイを AWS IoT Core for LoRaWAN にオンボードするには

- 周波数帯域の選択を検討し、必要な IAM ロールを追加する
- ゲートウェイを AWS IoT Core for LoRaWAN に追加する
- LoRaWAN ゲートウェイを接続し、接続ステータスを確認する

周波数帯域の選択を検討し、必要な IAM ロールを追加する

ゲートウェイを AWS IoT Core for LoRaWAN に追加する前に、ゲートウェイが動作する周波数帯域を検討し、ゲートウェイを AWS IoT Core for LoRaWAN に接続するために必要な IAM ロールを追加することをお勧めします。

> **Note**
>
> コンソールを使用してゲートウェイを追加する場合は、[Create role] (ロールの作成) をクリックして必要な IAM ロールを作成し、これらのステップをスキップできます。これらの手順は、CLI を使用してゲートウェイを作成する場合にのみ実行する必要があります。

ゲートウェイとデバイス接続用の LoRa 周波数帯域の選択を検討する

AWS IoT Core for LoRaWAN は、EU863-870、US902-928、AU915、および AS923-1 周波数帯域をサポートしており、お客様は、これらの周波数帯域と周波数帯域の特性をサポートする国に物理的に存在するゲートウェイとデバイスを接続するために、これらの周波数帯域を使用できます。EU863-870 および US902-928 帯域は、それぞれ欧州と北米で一般的に使用されています。AS923-1 帯域は、オーストラリア、ニュージーランド、日本、シンガポールなどで一般的に使用されています。AU915 は、とりわけオーストラリアとアルゼンチンで使用されています。お住まいの地域または国で使用する周波数帯域の詳細については、LoRaWAN® 地域別パラメータを参照してください。

LoRa Alliance は、LoRa Alliance のウェブサイトからダウンロードできる LoRaWAN 仕様と地域別パラメータドキュメントを公開しています。LoRa Alliance の地域パラメータは、企業が地域や国で使用する周波数帯域を決定するのに役立ちます。AWS IoT Core for LoRaWAN の周波数帯域の実装は、地域パラメータ仕様書の推奨事項に従います。これらの地域別パラメータは、産業、科学、および医療 (ISM) 帯域に適合する周波数割り当てとともに、一連の無線パラメータにグループ化されます。コンプライアンスチームと協力して、適用される規制要件を確実に満たすことをお勧めします。

Configuration and Update Server (CUPS) がゲートウェイ認証情報を管理することを許可するための IAM ロールを追加する

この手順では、Configuration and Update Server (CUPS) がゲートウェイ認証情報を管理することを許可する IAM ロールを追加する方法について説明します。LoRaWAN ゲートウェイが AWS IoT Core for LoRaWAN との接続を試みる前に、必ずこの手順を実行してください。ただし、これを行う必要があるのは 1 回だけです。

Configuration and Update Server (CUPS) がゲートウェイ認証情報を管理することを許可する IAM ロールを追加する

1. [Roles hub of the IAM console] (IAM コンソールのロールハブ) にログインして、[Create role] (ロールの作成) を選択します。

2. IoTWirelessGatewayCertManagerRole ロールを既に追加している可能性があると思われる場合は、検索バーに **IoTWirelessGatewayCertManagerRole** と入力します。

   検索結果に IoTWirelessGatewayCertManagerRole ロールが表示された場合、必要な IAM ロールがあります。これで手順を終了できます。

   検索結果が空の場合、必要な IAM ロールがありません。この手順を続行して、追加します。

3. [Select type of trusted entity] (信頼できるエンティティのタイプを選択) で、[Another AWS アカウント] (別の AWS アカウント) を選択します。

4. [Account ID] (アカウント ID) で AWS アカウント アカウント ID を入力し、[Next: Permissions] (次へ: アクセス許可) を選択します。

5. 検索ボックスに「**AWSIoTWirelessGatewayCertManager**」と入力します。

6. 検索結果のリストで、AWSIoTWirelessGatewayCertManager という名前のポリシーを選択します。

7. [次へ: タグ]、[次へ: 確認] の順に選択します。

8. [Role name] (ロール名) に **IoTWirelessGatewayCertManagerRole** と入力し、[Create role] (ロールの作成) を選択します。

9. 新しいロールを編集するには、確認メッセージで IoTWirelessGatewayCertManagerRole を選択します。

10. [Summary] (概要) で、[Trust relationships] (信頼関係) タブを選択し、続いて [Edit trust relationship] (信頼関係の編集) を選択します。

11. [Policy Document] (ポリシードキュメント) で、`Principal` プロパティを次の例のように変更します。

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

   `Principal` プロパティを変更すると、完全なポリシードキュメントは次の例のようになります。

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

```
      "Principal": {
        "Service": "iotwireless.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

12. 変更を保存して終了するには、[Update Trust Policy] (信頼ポリシーの更新) を選択します。

これで、IoTWirelessGatewayCertManagerRole が作成されました。これをもう一度行う必要はありません。

ゲートウェイの追加中にこの手順を実行した場合は、このウィンドウと IAM コンソールを閉じ、AWS IoT コンソールに戻り、ゲートウェイの追加を完了できます。

ゲートウェイを AWS IoT Core for LoRaWAN に追加する

コンソールまたは CLI を使用して、ゲートウェイを AWS IoT Core for LoRaWAN に追加できます。

ゲートウェイを追加する前に、「ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする」の「ゲートウェイをオンボードする前に」セクションで説明されている要素を考慮することをお勧めします。

初めてゲートウェイを追加する場合は、コンソールを使用することをお勧めします。CLI を使用してゲートウェイを追加する場合は、ゲートウェイが AWS IoT Core for LoRaWAN と接続するために必要な IAM ロールを既に作成済みである必要があります。ロールを作成する方法については、「Configuration and Update Server (CUPS) がゲートウェイ認証情報を管理することを許可するための IAM ロールを追加する」を参照してください。

コンソールを使用してゲートウェイを追加する

AWS IoT コンソールの AWS IoT Core for LoRaWAN の [Intro] (イントロダクション) ページに移動し、[Get started] (開始方法) を選択して、[Add gateway] (ゲートウェイを追加) を選択します。ゲートウェイを既に追加している場合は、[View gateway] (ゲートウェイの表示) を選択して、追加したゲートウェイを表示します。ゲートウェイをさらに追加する場合は、[Add gateway] (ゲートウェイの追加) を選択します。

1. ゲートウェイの詳細と周波数帯域情報を提供する

[Gateway details] (ゲートウェイの詳細) セクションを使用して、ゲートウェイの EUI や周波数帯域設定などのデバイス設定データに関する情報を提供します。

- ゲートウェイの EUI

  個々のゲートウェイデバイスの EUI (拡張一意識別子)。EUI は `c0ee40ffff29df10` などの 16 桁の英数字コードであり、LoRaWAN ネットワーク内のゲートウェイを一意に識別します。この情報はゲートウェイモデルに固有のものであり、ゲートウェイデバイスまたはそのユーザーマニュアルに記載されています。

  > **Note**
  >
  > ゲートウェイの EUI は、ゲートウェイデバイスに表示される Wi-Fi MAC アドレスとは異なります。EUI は、ゲートウェイを一意に識別する EUI-64 標準に準拠しているため、他の AWS アカウント とリージョンで再利用できません。

- 周波数帯域 (RFRegion)

  ゲートウェイの周波数帯。ゲートウェイがサポートするもの、およびゲートウェイの物理的な接続元の国または地域に応じて、US915、EU868、AU915、または AS923-1 から選択できます。帯域の詳細については、ゲートウェイとデバイス接続用の LoRa 周波数帯域の選択を検討する を参照してください。

2. ワイヤレスゲートウェイ設定データを指定する (オプション)

   これらのフィールドはオプションで、ゲートウェイとその設定に関する追加情報を提供するために使用できます。

   - ゲートウェイの名前、説明、タグ

     これらのオプションフィールドの情報は、ワイヤレスシステムの要素をどのように編成および説明するかによって決まります。ゲートウェイに名前を追加し、[Description] (説明) フィールドを使用してゲートウェイに関する情報を提供し、タグを使用してゲートウェイに関するメタデータのキーと値のペアを追加できます。リソースの命名と説明の詳細については、AWS IoT Wireless リソースについて説明する を参照してください。

   - サブバンドとフィルターを使用した LoRaWAN 構成

     オプションで、使用するサブバンドやトラフィックのフローを制御できるフィルターなどの LoRaWAN 設定データを指定することもできます。このチュートリアルでは、これらのフィー

ルドをスキップできます。詳細については、「ゲートウェイのサブバンドとフィルタリング機能を設定する」を参照してください。

3. AWS IoT のモノとゲートウェイを関連付ける

   AWS IoTのモノを作成し、ゲートウェイに関連付けるかどうかを指定します。AWS IoT におけるモノを使用すると、デバイスの検索と管理を簡単に行えるようになります。モノをゲートウェイに関連付けると、ゲートウェイは他の AWS IoT Core 機能にアクセスできます。

4. ゲートウェイ証明書を作成してダウンロードする

   ゲートウェイが安全に AWS IoT と通信できるようにゲートウェイを認証するには、LoRaWAN ゲートウェイは AWS IoT Core for LoRaWAN に対してプライベートキーと証明書を提示する必要があります。AWS IoT が X.509 標準を使用してゲートウェイの ID を確認できるように、ゲートウェイ証明書を作成します。

   [Create certificate] (証明書の作成) ボタンをクリックして、証明書ファイルをダウンロードします。これらはゲートウェイを設定するために後で使用します。

5. CUPS エンドポイントと LNS エンドポイントをコピーし、証明書をダウンロードする

   LoRaWAN ゲートウェイは、AWS IoT Core for LoRaWAN への接続を確立するときに、CUPS エンドポイントまたは LNS エンドポイントに接続する必要があります。CUPS エンドポイントでは設定管理も提供されるため、CUPS エンドポイントを使用することをお勧めします。AWS IoT Core for LoRaWAN エンドポイントの真正性を確認するために、ゲートウェイは CUPS エンドポイントおよび LNS エンドポイントごとに信頼証明書を使用します。

   [Copy] (コピー) ボタンをクリックして CUPS エンドポイントと LNS エンドポイントをコピーします。この情報はゲートウェイを構成するために後で必要になります。次に、[Download server trust certificates] (サーバー信頼証明書のダウンロード) ボタンをクリックして、CUPS エンドポイントおよび LNS エンドポイントの信頼証明書をダウンロードします。

6. ゲートウェイのアクセス許可のための IAM ロールを作成する

   Configuration and Update Server (CUPS) がゲートウェイ認証情報を管理することを許可する IAM ロールを追加する必要があります。

   > **Note**
   >
   > このステップでは、IoTWirelessGatewayCertManager ロールを作成します。すでにこのロールを作成している場合は、このステップをスキップできます。LoRaWAN ゲートウェ

イが AWS IoT Core for LoRaWAN との接続を試みる前に、これを行う必要があります。ただし、実行する必要があるのは 1 回だけです。

アカウントに IoTWirelessGatewayCertManager IAM ロールを作成するには、[Create role] (ロールの作成) ボタンをクリックします。ロールが既に存在する場合は、ドロップダウンリストから選択します。

[Submit] (送信) をクリックして、ゲートウェイの作成を完了します。

API を使用してゲートウェイを追加する

API または CLI を使用して初めてゲートウェイを追加する場合は、ゲートウェイが AWS IoT Core for LoRaWAN と接続できるように IoTWirelessGatewayCertManager IAM ロールを追加する必要があります。ロールの作成方法については、「Configuration and Update Server (CUPS) がゲートウェイ認証情報を管理することを許可するための IAM ロールを追加する」セクションを参照してください。

以下に、LoRaWAN ゲートウェイの追加、更新、削除に関連するタスクを実行する API アクションを示します。

AWS IoT Core for LoRaWAN ゲートウェイ向けの AWS IoT Wireless API アクション

- CreateWirelessGateway
- GetWirelessGateway
- ListWirelessGateways
- UpdateWirelessGateway
- DeleteWirelessGateway

AWS IoT Core for LoRaWAN リソースを作成および管理するために使用できるアクションとデータタイプの完全なリストについては、「AWS IoT Wireless API reference」を参照してください。

AWS CLI を使用してゲートウェイを追加する方法

AWS CLI を使用して、create-wireless-gateway コマンドによってワイヤレスゲートウェイを作成できます。次の例では、ワイヤレス LoRaWAN デバイスゲートウェイを作成します。ゲートウェイ証明書やプロビジョニング認証情報などの追加の詳細が含まれる `input.json` ファイルを指定することもできます。

> **Note**
>
> この手順は、ここに示す CLI コマンドに対応する AWS API のメソッドを使用することにより、API で行うこともできます。

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

使用可能な CLI の詳細については、「AWS CLI リファレンス」を参照してください。

LoRaWAN ゲートウェイを接続し、接続ステータスを確認する

ゲートウェイの接続ステータスを確認する前に、ゲートウェイを追加済みで、AWS IoT Core for LoRaWAN に接続済みである必要があります。ゲートウェイを追加する方法については、「ゲートウェイを AWS IoT Core for LoRaWAN に追加する」を参照してください。

ゲートウェイを AWS IoT Core for LoRaWAN に接続する

ゲートウェイを追加したら、ゲートウェイの設定インターフェイスに接続して、設定情報と信頼証明書を入力します。

AWS IoT Core for LoRaWAN にゲートウェイの情報を追加した後、AWS IoT Core for LoRaWAN 情報をゲートウェイデバイスに追加します。ゲートウェイのベンダーが提供するドキュメントでは、証明書ファイルをゲートウェイにアップロードし、AWS IoT Core for LoRaWAN と通信するようにゲートウェイデバイスを設定するプロセスについて説明されているはずです。

AWS IoT Core for LoRaWAN での使用を認定されたゲートウェイ

LoRaWAN ゲートウェイを設定する手順については、AWS IoT Core for LoRaWAN ワークショップの「Configure gateway device」セクションを参照してください。ここでは、AWS IoT Core for LoRaWAN での使用が認定されたゲートウェイを接続するための手順について説明します。

CUPS プロトコルをサポートするゲートウェイ

次の手順で、CUPS プロトコルをサポートするゲートウェイを接続する方法について説明します。

1. ゲートウェイの追加時に取得した次のファイルをアップロードします。

- ゲートウェイデバイス証明書およびプライベートキーファイル。
- CUPS エンドポイントの信頼証明書ファイル `cups.trust`。

2. 前に取得した CUPS エンドポイント URL を指定します。エンドポイントは *prefix*`.cups.lorawan.`*region*`.amazonaws.com:443` の形式です。

この情報の入手方法については、「ゲートウェイを AWS IoT Core for LoRaWAN に追加する」を参照してください。

LNS プロトコルをサポートするゲートウェイ

次の手順で、LNS プロトコルをサポートするゲートウェイを接続する方法について説明します。

1. ゲートウェイの追加時に取得した次のファイルをアップロードします。
   - ゲートウェイデバイス証明書およびプライベートキーファイル。
   - LNS エンドポイントの信頼証明書ファイル `lns.trust`。
2. 前に取得した LNS エンドポイント URL を指定します。エンドポイントは https://*prefix*`.lns.lorawan.`*region*`.amazonaws.com:443` の形式です。

この情報の入手方法については、「ゲートウェイを AWS IoT Core for LoRaWAN に追加する」を参照してください。

ゲートウェイを AWS IoT Core for LoRaWAN に接続したら、コンソールまたは API を使用して、接続のステータスを確認し、最後のアップリンクがいつ受信されたかに関する情報を取得できます。

コンソールを使用してゲートウェイ接続ステータスを確認する

コンソールを使用して接続ステータスを確認するには、AWS IoT コンソールの [Gateways] (ゲートウェイ) ページに移動し、追加したゲートウェイを選択します。ゲートウェイの詳細ページの [LoRaWAN specific details] (LoRaWAN 固有の詳細) セクションに、接続ステータスと最後にアップリンクが受信された日時が表示されます。

API を使用してゲートウェイ接続ステータスを確認する

API を使用して接続ステータスを確認するには、`GetWirelessGatewayStatistics` API を使用します。この API にはリクエストボディがなく、ゲートウェイが接続されているかどうか、および最後のアップリンクがいつ受信されたかを示すレスポンス本文のみが含まれます。

```
HTTP/1.1 200
```

```
Content-type: application/json

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

デバイスを AWS IoT Core for LoRaWAN にオンボードする

ゲートウェイをAWS IoT Core for LoRaWAN にオンボードして接続ステータスを確認したら、ワイヤレス機器をオンボードできます。ゲートウェイをオンボードする方法については、「ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。

LoRaWAN デバイスは、LoRaWAN プロトコルを使用して、クラウドホスト型アプリケーションとデータを交換します。AWS IoT Core for LoRaWAN は、LoRa Alliance によって標準化された 1.0.x または 1.1 LoRaWAN 仕様に準拠したデバイスをサポートしています。

LoRaWAN デバイスは、通常、1 つ以上のセンサーとアクターで構成されます。デバイスは、LoRaWAN ゲートウェイを介してアップリンクテレメトリデータを AWS IoT Core for LoRaWAN に送信します。クラウドでホストされたアプリケーションは、LoRaWAN ゲートウェイを介して LoRaWAN デバイスにダウンリンクコマンドを送信することで、センサーを制御できます。

ワイヤレスデバイスをオンボードする前に

ワイヤレスデバイスを AWS IoT Core for LoRaWAN にオンボードする前に、以下の情報を事前に用意しておく必要があります。

- LoRaWAN の仕様と無線機器の設定

  各デバイスに固有の設定パラメータの入力準備を事前にしておくと、コンソールへのデータの入力がよりスムーズになります。入力する必要がある特定のパラメータは、デバイスが使用する LoRaWAN 仕様によって異なります。仕様と設定パラメータの完全なリストについては、各デバイスに関するドキュメントを参照してください。
- デバイス名と説明 (オプション)

  これらのオプションフィールドの情報は、ワイヤレスシステムの要素をどのように編成および説明するかによって決まります。リソースの命名と説明の詳細については、AWS IoT Wireless リソースについて説明する を参照してください。
- デバイスおよびサービスプロファイル

多くのデバイスで共有され、AWS IoT Core for LoRaWAN にデバイスおよびサービスプロファイルとして保存できる一部のワイヤレスデバイス設定パラメータを用意しておきます。設定パラメータは、デバイスのドキュメントまたはデバイス自体に記載されています。デバイスを追加する前に、デバイスの設定パラメータに一致するデバイスプロファイルを特定するか、必要に応じて作成します。詳細については、「プロファイルを AWS IoT Core for LoRaWAN に追加する」を参照してください。

- AWS IoT Core for LoRaWAN 送信先

  各デバイスは、メッセージを処理して AWS IoT やその他のサービスに送信する送信先に割り当てる必要があります。デバイスメッセージを処理および送信する AWS IoT ルールは、デバイスのメッセージ形式に固有です。デバイスからのメッセージを処理して正しいサービスに送信するには、デバイスのメッセージで使用するために作成する送信先を特定し、デバイスに割り当てます。

ワイヤレスデバイスを AWS IoT Core for LoRaWAN にオンボードするには

- ワイヤレスデバイスを AWS IoT Core for LoRaWAN に追加する
- プロファイルを AWS IoT Core for LoRaWAN に追加する
- AWS IoT Core for LoRaWAN に送信先を追加する
- LoRaWAN デバイスメッセージを処理するルールを作成する
- LoRaWAN デバイスを接続し、接続ステータスを確認する

ワイヤレスデバイスを AWS IoT Core for LoRaWAN に追加する

初めてワイヤレスデバイスを追加する場合は、コンソールを使用することをお勧めします。AWS IoT コンソールの AWS IoT Core for LoRaWAN の [Intro] (イントロダクション) ページに移動し、[Get started] (開始方法) を選択して、[Add device] (デバイスを追加) を選択します。デバイスを既に追加している場合は、[View device] (デバイスの表示) を選択して、追加したゲートウェイを表示します。デバイスを追加する場合は、[Add device] (デバイスの追加) を選択します。

または、AWS IoT コンソールの [Devices] (デバイス) ページからワイヤレスデバイスを追加することもできます。

コンソールを使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する

アクティベーション方法と LoRaWAN のバージョンに基づいて、ワイヤレスデバイスの仕様を選択します。選択すると、AWS が所有して管理するキーによってデータが暗号化されます。

OTAA および ABP アクティベーションモード

LoRaWAN デバイスがアップリンクデータを送信する前に、アクティベーションまたは参加手順と呼ばれるプロセスを完了する必要があります。デバイスをアクティベートするには、OTAA (無線通信経由アクティベーション) または ABP (パーソナライゼーションによるアクティベーション) のいずれかを使用できます。

ABP は参加手順を必要とせず、静的キーを使用します。OTAA を使用すると、LoRaWAN デバイスが参加要求を送信し、ネットワークサーバーが要求を許可することができます。アクティベーションごとに新しいセッションキーが生成されて安全性が高まるため、デバイスのアクティベーションには OTAA を使用することをお勧めします。

LoRaWAN バージョン

OTAA を使用すると、LoRaWAN デバイスとクラウドホスト型アプリケーションがルートキーを共有します。これらのルートキーは、バージョン v1.0.x と v1.1 のどちらを使用しているかによって異なります。v1.0.x には AppKey (アプリケーションキー) の 1 つのルートキーしかありません。v1.1 には AppKey (アプリケーションキー) と NwkKey (ネットワークキー) の 2 つのルートキーがあります。セッションキーは、各アクティベーションのルートキーに基づいて生成されます。NwkKey と AppKey はいずれも、ワイヤレスベンダーが提供した 32 桁の 16 進数値です。

ワイヤレスデバイスの EUI

ワイヤレスデバイスの仕様を選択すると、コンソールに表示されているワイヤレスデバイスの EUI (拡張一意識別子) パラメータが表示されます。この情報は、デバイスまたはワイヤレスベンダーのドキュメントに記載されています。

- DevEUI: お使いのデバイスに固有の 16 桁の 16 進数値で、デバイスのラベルまたはドキュメントに記載されています。
- AppEUI: 参加サーバーに固有の 16 桁の 16 進数値で、デバイスのドキュメントに記載されています。LoRaWAN バージョン v1.1 では、AppEUI は JoinEUI と呼ばれます。

一意識別子、セッションキー、およびルートキーの詳細については、LoRa Alliance ドキュメントを参照してください。

API を使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する

API を使用してワイヤレスデバイスを追加する場合は、ワイヤレスデバイスを作成する前に、まずデバイスプロファイルとサービスプロファイルを作成する必要があります。ワイヤレスデバイスを作

成するときは、デバイスプロファイル ID とサービスプロファイル ID を使用します。API を使用してこれらのプロファイルを作成する方法については、「API を使用してデバイスプロファイルを追加する」を参照してください。

以下に、サービスプロファイルの追加、更新、削除に関連するタスクを実行する API アクションを示します。

サービスプロファイルに対する AWS IoT Wireless API アクション

- CreateWirelessDevice
- GetWirelessDevice
- ListWirelessDevices
- UpdateWirelessDevice
- DeleteWirelessDevice

AWS IoT Core for LoRaWAN リソースを作成および管理するために使用できるアクションとデータタイプの完全なリストについては、「AWS IoT Wireless API reference」を参照してください。

AWS CLI を使用してワイヤレスデバイスを作成する方法

AWS CLI を使用して、create-wireless-device コマンドによってワイヤレスデバイスを作成できます。次の例では、input.json ファイルを使用してパラメータを入力することにより、ワイヤレスデバイスを作成します。

> **Note**
>
> この手順は、ここに示す CLI コマンドに対応する AWS API のメソッドを使用することにより、API で行うこともできます。

input.json の内容

```
{
    "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
}
```

create-wireless-device コマンドへの入力としてこのファイルを指定できます。

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

使用可能な CLI の詳細については、「AWS CLI リファレンス」を参照してください。

プロファイルを AWS IoT Core for LoRaWAN に追加する

デバイスおよびサービスプロファイルを定義して、一般的なデバイス設定を記述できます。これらのプロファイルは、デバイスによって共有される設定パラメータを記述し、これらのデバイスを簡単に追加できるようにします。AWS IoT Core for LoRaWAN では、デバイスプロファイルとサービスプロファイルがサポートされています。

これらのプロファイルに入力する設定パラメータと値は、デバイスの製造元によって提供されます。

デバイスプロファイルを追加する

デバイスプロファイルは、ネットワークサーバーが LoRaWAN 無線アクセスサービスを設定するために使用するデバイス機能とブートパラメータを定義します。これには、LoRa 周波数帯域、LoRa 地域別パラメータバージョン、デバイスの MAC バージョンなどのパラメータの選択が含まれます。さまざまな周波数帯域については、ゲートウェイとデバイス接続用の LoRa 周波数帯域の選択を検討する を参照してください。

コンソールを使用してデバイスプロファイルを追加する

コンソールを使用してワイヤレスデバイスを追加する場合は、「コンソールを使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する」で説明されているように、ワイヤレスデバイスの仕様を追加した後、デバイスプロファイルを追加できます。または、AWS IoT コンソール

の LoRaWAN タブの [Profiles] (プロファイル) ページからワイヤレスデバイスを追加することもできます。

デフォルトのデバイスプロファイルから選択するか、新しいデバイスプロファイルを作成できます。デフォルトのデバイスプロファイルを使用することをお勧めします。アプリケーションでデバイスプロファイルの作成が必要な場合は、デバイスプロファイル名を指定し、[Frequency band (RfRegion)] (周波数帯 (RfRegion)) を設定します。デバイスのマニュアルで特に指定されていない限り、その他の設定はデフォルト値のままにします。

API を使用してデバイスプロファイルを追加する

API を使用してワイヤレスデバイスを追加する場合は、ワイヤレスデバイスを作成する前にデバイスプロファイルを作成する必要があります。

以下に、サービスプロファイルの追加、更新、削除に関連するタスクを実行する API アクションを示します。

サービスプロファイルに対する AWS IoT Wireless API アクション

- CreateDeviceProfile
- GetDeviceProfile
- ListDeviceProfiles
- UpdateDeviceProfile
- DeleteDeviceProfile

AWS IoT Core for LoRaWAN リソースを作成および管理するために使用できるアクションとデータタイプの完全なリストについては、「AWS IoT Wireless API reference」を参照してください。

AWS CLI を使用してデバイスプロファイルを作成する方法

AWS CLI を使用して、create-device-profile コマンドによってデバイスプロファイルを作成できます。次の例では、デバイスプロファイルを作成します。

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

このコマンドを実行すると、ワイヤレスデバイスの作成時に使用できる ID を持つデバイスプロファイルが自動的に作成されます。これで、次の API を使用してサービスプロファイルを作成し、デバ

イスプロファイルとサービスプロファイルを使用してワイヤレスデバイスを作成できるようになりました。

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

使用可能な CLI の詳細については、AWS CLI リファレンスを参照してください。

サービスプロファイルを追加する

サービスプロファイルは、デバイスがアプリケーションサーバーと通信するために必要な通信パラメータを記述します。

コンソールを使用してサービスプロファイルを追加する

コンソールを使用してワイヤレスデバイスを追加する場合は、「コンソールを使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する」で説明されているように、デバイスプロファイルを追加した後、サービスプロファイルを追加できます。または、AWS IoT コンソールの LoRaWAN タブの [Profiles] (プロファイル) ページからワイヤレスデバイスを追加することもできます。

AddGWMetaData 設定は有効のままにしておくことをお勧めします。有効にすると、データ転送の RSSI や SNR など、ペイロードごとに追加のゲートウェイメタデータを受信できるようになります。

API を使用してサービスプロファイルを追加する

API を使用してワイヤレスデバイスを追加する場合は、ワイヤレスデバイスを作成する前にサービスプロファイルを作成する必要があります。

以下に、サービスプロファイルの追加、更新、削除に関連するタスクを実行する API アクションを示します。

サービスプロファイルに対する AWS IoT Wireless API アクション

- CreateServiceProfile
- GetServiceProfile
- ListServiceProfiles

- UpdateServiceProfile
- DeleteServiceProfile

AWS IoT Core for LoRaWAN リソースを作成および管理するために使用できるアクションとデータタイプの完全なリストについては、「AWS IoT Wireless API reference」を参照してください。

AWS CLI を使用してサービスプロファイルを作成する方法

AWS CLI を使用して、create-service-profile コマンドによってサービスプロファイルを作成できます。次の例では、サービスプロファイルを作成します。

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

このコマンドを実行すると、ワイヤレスデバイスの作成時に使用できる ID を持つサービスプロファイルが自動的に作成されます。これで、デバイスプロファイルとサービスプロファイルを使用して、ワイヤレスデバイスを作成できるようになりました。

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

AWS IoT Core for LoRaWAN に送信先を追加する

AWS IoT Core for LoRaWAN の送信先は、AWS のサービスで使用するデバイスのデータを処理する AWS IoT ルールを記述します。

ほとんどの LoRaWAN デバイスは、AWS のサービスで使用できる形式で AWS IoT Core for LoRaWAN にデータを送信しないため、AWS IoT ルールで最初にデータを処理する必要があります。AWS IoT ルールには、デバイスのデータを解釈する SQL ステートメントと、SQL ステートメントの結果を、それを使用するサービスに送信するトピックルールアクションが含まれています。

送信先を初めて追加している場合は、コンソールの使用をお勧めします。

コンソールを使用して送信先を追加します

「コンソールを使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する」で説明されているように、コンソールを使用してワイヤレスデバイスを追加する場合は、ワイヤレスデ

バイスの仕様とプロファイルを AWS IoT Core for LoRaWAN に前述のように追加したら、先に進んで送信先を追加することができます。

または、AWS IoT コンソールの [Destinations] (送信先) ページから AWS IoT Core for LoRaWAN の送信先を追加することもできます。

デバイスのデータを処理するには、AWS IoT Core for LoRaWAN の送信先を作成するときに以下のフィールドを指定して、[Add destination] (送信先を追加) を選択します。

- 送信先の詳細

  [Destination name] (送信先名) と、必要に応じて送信先の説明を入力します。
- ルール名

  デバイスが送信したメッセージを評価し、デバイスのデータを処理するために設定された AWS IoT ルールです。ルール名は送信先にマップされます。送信先には、受信したメッセージを処理するためのルールが必要です。メッセージは、AWS IoT ルールの呼び出し、または AWS IoT メッセージブローカーへのパブリッシュのいずれかによって処理されるように選択できます。
  - [Enter a rule name] (ルール名を入力) を選択する場合は、名前を入力し、次に [Copy] (コピー) をクリックして、AWS IoT ルールを作成するときに入力するルール名をコピーします。[Create rule] (ルールの作成) を選択して今すぐルールを作成するか、AWS IoT コンソールの [Rules] (ルール) ハブを開き、その名前のルールを作成します。

    ルールを入力し、[Advanced] (アドバンスト) 設定を使用してトピック名を指定することもできます。トピック名はルールの呼び出し中に提供され、ルール内の `topic` 式を使用してアクセスします。AWS IoT ルールの詳細については、「https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html」を参照してください。
  - [Publish to AWS IoT message broker] (IoT メッセージブローカーに発行) を選択する場合は、トピック名を入力します。その後、MQTT トピック名をコピーできます。また、複数のサブスクライバーがこのトピックにサブスクライブして、そのトピックに発行されたメッセージを受信できます。詳細については、「https://docs.aws.amazon.com/iot/latest/developerguide/topics.html」を参照してください。

  送信先の AWS IoT ルールの詳細については、LoRaWAN デバイスメッセージを処理するルールを作成する を参照してください。
- ロール名

  [Rule name] (ルール名) で名前を付けたルールにアクセスするための許可をデバイスのデータに付与する IAM ロール。コンソールでは、新しいサービスロールを作成する、または既存の

サービスロールを選択することができます。新しいサービスロールを作成する場合は、ロール名 (例、**IoTWirelessDestinationRole**) を入力するか、あるいは、空白のままにして AWS IoT Core for LoRaWAN で新しいロール名を作成することできます。その後、適切なアクセス許可を持つ IAM ロールが AWS IoT Core for LoRaWAN で自動的に作成されます。

IAM ロールの詳細については、IAM ロールを使用するを参照してください。

API を使用して送信先を追加します

CLI を使用して送信先を追加する場合は、送信先のルールと IAM ロールが既に作成されている必要があります。ロールに関する送信先の要件の詳細については、「送信先の IAM ロールを作成する」を参照してください。

以下のリストには、送信先の追加、更新、または削除に関連付けられたタスクを実行する API アクションが記載されています。

送信先に対する AWS IoT Wireless API アクション

- CreateDestination
- GetDestination
- ListDestinations
- UpdateDestination
- DeleteDestination

AWS IoT Core for LoRaWAN リソースを作成および管理するために使用できるアクションとデータタイプの完全なリストについては、「AWS IoT Wireless API reference」を参照してください。

AWS CLI を使用して送信先を追加する方法

AWS CLI を使用して、create-destination コマンドによって送信先を追加できます。以下の例は、`RuleName` を `expression-type` パラメータの値として使用してルール名を入力することによって送信先を作成する方法を示しています。メッセージブローカーにパブリッシュまたはサブスクライブするためのトピック名を指定する場合は、`expression-type` パラメータの値を`MqttTopic` に変更してください。

```
aws iotwireless create-destination \
    --name IoTWirelessDestination \
    --expression-type RuleName \
    --expression IoTWirelessRule \
```

```
    --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

このコマンドを実行すると、指定した送信先名、ルール名、ロール名を持つ送信先が作成されます。送信先のルール名およびロール名については、LoRaWAN デバイスメッセージを処理するルールを作成する と 送信先の IAM ロールを作成する を参照してください。

使用可能な CLI の詳細については、AWS CLI リファレンスを参照してください。

送信先の IAM ロールを作成する

AWS IoT Core for LoRaWAN の送信先には、AWS IoT ルールにデータを送信するために必要なアクセス許可を AWS IoT Core for LoRaWAN に付与する IAM ロールが必要です。そのようなロールがまだ定義されていない場合は、ロールのリストに表示されるように定義する必要があります。

コンソールを使用して送信先を追加するときは、このトピックで先ほど説明したように、AWS IoT Core for LoRaWAN が自動的に IAM ロールを作成します。API または CLI を使用して送信先を追加する場合、送信先の IAM ロールを作成する必要があります。

AWS IoT Core for LoRaWAN 送信先ロール用の IAM ポリシーを作成するには

1. IAM コンソールのポリシーハブを開きます。
2. [Create policy] (ポリシーの作成) を選択し、[JSON] タブを選択します。
3. エディタで、エディタからコンテンツを削除し、このポリシードキュメントを貼り付けます。

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

4. [Review policy] (ポリシーの確認) を選択し、[Name] (名前) でこのポリシーの名前を入力します。この名前は、次の手順で使用するために必要です。

   必要に応じて、このポリシーを [Description] (説明) で説明することもできます。

5. [Create policy] (ポリシーの作成) を選択します。

AWS IoT Core for LoRaWAN 送信先の IAM ロールを作成するには

1. [Roles hub of the IAM console] (IAM コンソールのロールハブ) にログインして、[Create role] (ロールの作成) を選択します。
2. [Select type of trusted entity] (信頼できるエンティティのタイプを選択) で、[Another AWS アカウント] (別の AWS アカウント) を選択します。
3. [Account ID] (アカウント ID) で AWS アカウント アカウント ID を入力し、[Next: Permissions] (次へ: アクセス許可) を選択します。
4. 検索ボックスで、前の手順で作成した IAM ポリシーの名前を入力します。
5. 検索結果で、前の手順で作成した IAM ポリシーを確認します。
6. [次へ: タグ]、[次へ: 確認] の順に選択します。
7. [Role name] (ロール名) でこのロールの名前を入力し、[Create role] (ロールの作成) を選択します。
8. 確認メッセージで、新しいロールを編集するために作成したロールの名前を選択します。
9. [Summary] (概要) で、[Trust relationships] (信頼関係) タブを選択し、続いて [Edit trust relationship] (信頼関係の編集) を選択します。
10. [Policy Document] (ポリシードキュメント) で、`Principal` プロパティを次の例のように変更します。

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

`Principal` プロパティを変更すると、完全なポリシードキュメントは次の例のようになります。

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

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

11. 変更を保存して終了するには、[Update Trust Policy] (信頼ポリシーの更新) を選択します。

このロールを定義すると、AWS IoT Core for LoRaWAN の送信先を設定するときに、ロールのリストにそのロールが表示されます。

LoRaWAN デバイスメッセージを処理するルールを作成する

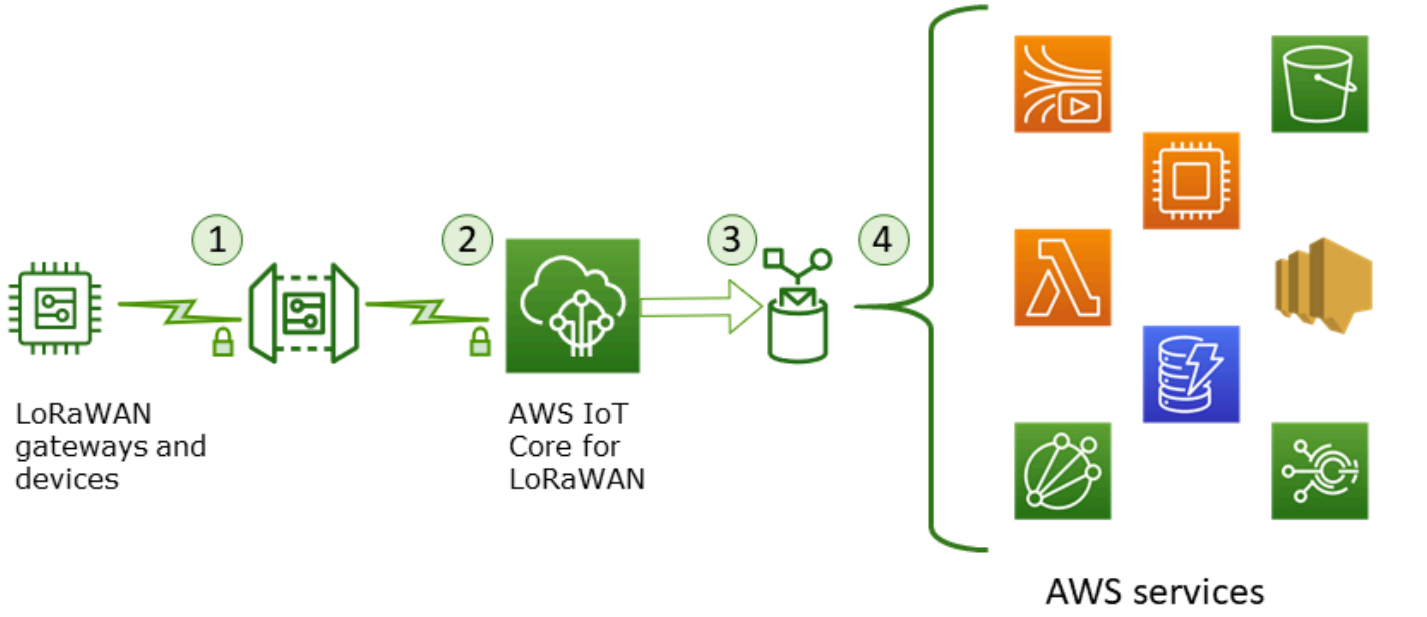
AWS IoT ルールは、デバイスメッセージを他のサービスに送信します。AWS IoT は、LoRaWAN デバイスから受信したバイナリメッセージを処理して、メッセージを他の形式に変換して、他のサービスで使用しやすくすることもできます。

AWS IoT Core for LoRaWAN の送信先は、デバイスのメッセージデータを処理して他のサービスに送信するルールにワイヤレスデバイスを関連付けます。このルールは、AWS IoT Core for LoRaWAN がデバイスのデータを受信するとすぐに、そのデータに対して機能します。AWS IoT Core for LoRaWAN の送信先は、メッセージが同じデータ形式で、データを同じサービスに送信するすべてのデバイスで共有できます。

AWS IoT ルールがデバイスメッセージを処理する方法

AWS IoT ルールがデバイスのメッセージデータを処理する方法は、データを受信するサービス、デバイスのメッセージデータの形式、およびサービスが必要とするデータ形式によって異なります。通常、このルールは AWS Lambda 関数を呼び出して、デバイスのメッセージデータをサービスが必要とする形式に変換し、その結果をサービスに送信します。

次の図は、メッセージデータがワイヤレスデバイスから AWS のサービスに移動するときに、メッセージデータがどのように保護および処理されるかを示しています。



1. LoRaWAN ワイヤレスデバイスは、バイナリメッセージを送信する前に AES128 CTR モードを使用して暗号化します。
2. AWS IoT Core for LoRaWAN は、バイナリメッセージを復号し、復号されたバイナリメッセージペイロードを base64 文字列としてエンコードします。
3. 結果として生じる base64 でエンコードされたメッセージは、メッセージペイロード (JSON ドキュメントとしてフォーマットされていない) として、デバイスに割り当てられた送信先に記載されている AWS IoT ルールに送信されます。
4. AWS IoT ルールは、ルールの設定に記載されているサービスにメッセージデータを送信します。

ワイヤレスデバイスから受信した、暗号化されたバイナリペイロードは、AWS IoT Core for LoRaWAN によって変更および解釈されません。復号されたバイナリメッセージペイロードは、base64 文字列としてのみエンコードされます。サービスがバイナリメッセージペイロードのデータ要素にアクセスするためには、ルールによって呼び出される関数によってデータ要素をペイロードから解析する必要があります。base64 でエンコードされたメッセージペイロードは ASCII 文字列であるため、後で解析するために保存することができます。

LoRaWAN デバイスのルールを作成する

AWS IoT Core for LoRaWAN は、メッセージブローカーを使用することなく、AWS IoT ルールを使用して、デバイスメッセージを他の AWS のサービスに直接安全に送信します。取り込みパスからメッセージブローカーを削除することで、コストを削減し、データフローを最適化します。

デバイスメッセージを他の AWS のサービスに送信するための AWS IoT Core for LoRaWAN ルールには、AWS IoT Core for LoRaWAN 送信先と、その送信先に割り当てられた AWS IoT ルールが必要です。AWS IoT ルールには、SQL クエリステートメントと少なくとも 1 つのルールアクションが含まれている必要があります。

通常、AWS IoT ルールクエリステートメントは、次のもので構成されます。

- メッセージペイロードからデータを選択してフォーマットする SQL SELECT 句
- 使用するメッセージを識別するトピックフィルター (ルールクエリステートメントの FROM オブジェクト)
- アクションを実行する条件を指定するオプションの条件文 (SQL WHERE 句)

ルールクエリステートメントの例を以下に示します。

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

LoRaWAN デバイスからのペイロードを処理するために AWS IoT ルールを構築する場合、ルールクエリオブジェクトの一部として FROM 句を指定する必要はありません。ルールクエリステートメントには SQL SELECT 句が含まれている必要があり、オプションで WHERE 句を含めることができます。クエリステートメントで FROM 句が使用されている場合は、無視されます。

LoRaWAN デバイスからペイロードを処理できるルールクエリステートメントの例を次に示します。

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

この例では、PayloadData は、LoRaWAN デバイスによって送信される base64 でエンコードされたバイナリペイロードです。

受信ペイロードのバイナリデコードを実行し、JSON などの別の形式に変換できるルールクエリステートメントの例を次に示します。

```
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
```

SELECT AND WHERE 句の使用の詳細については、「https://docs.aws.amazon.com/iot/latest/developerguide/iot-sql-reference.html」を参照してください。

AWS IoT ルールとその作成方法および使用方法については、https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html と https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules-tutorial.html を参照してください。

AWS IoT Core for LoRaWAN 送信先の作成と使用については、「AWS IoT Core for LoRaWAN に送信先を追加する」を参照してください。

ルールでのバイナリメッセージペイロードの使用については、https://docs.aws.amazon.com/iot/latest/developerguide/binary-payloads.html を参照してください。

送信中のメッセージペイロードを保護するために使用されるデータセキュリティと暗号化の詳細については、AWS IoT Wireless でのデータ保護 を参照してください。

IoT ルールのバイナリデコードと実装例を示すリファレンスアーキテクチャについては、GitHub の「AWS IoT Core for LoRaWAN Solution Samples」 を参照してください。

LoRaWAN デバイスを接続し、接続ステータスを確認する

デバイスの接続ステータスを確認する前に、デバイスが追加済みで、AWS IoT Core for LoRaWAN に接続済みである必要があります。デバイスの追加方法については、「ワイヤレスデバイスを AWS IoT Core for LoRaWAN に追加する」を参照してください。

デバイスを追加したら、デバイスのユーザーマニュアルを参照して、LoRaWAN デバイスからアップリンクメッセージの送信を開始する方法をご確認ください。

コンソールを使用してデバイスの接続ステータスを確認する

コンソールを使用して接続ステータスを確認するには、AWS IoT コンソールの [Devices] (デバイス) ページに移動し、追加したデバイスを選択します。ワイヤレスデバイスの詳細ページの [Details] (詳細) セクションに、最後にアップリンクが受信された日時が表示されます。

API を使用してデバイスの接続ステータスを確認する

API を使用して接続ステータスを確認するには、`GetWirelessDeviceStatistics` API を使用します。この API にはリクエストボディがなく、最後のアップリンクがいつ受信されたかを示すレスポンス本文のみが含まれます。

```
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"
}
```

次のステップ

これで、デバイスを接続して接続ステータスを確認できたので、AWS IoT コンソールの [Test] (テスト) ページの MQTT テストクライアントを使用して、デバイスから受信したアップリンクメタデータの形式を確認できます。詳細については、「LoRaWAN デバイスから送信されたアップリンクメッセージの形式の表示」を参照してください。

AWS IoT Core for LoRaWAN によるワイヤレスリソースの位置の設定

この機能を使用する前に、LoRaWAN デバイスの位置情報を解決するために選択したサードパーティープロバイダーは、国際 GNSS サービス (IGS)、NASA 経由の EarthData、その他のサードパーティーによって提供または管理されているデータフィードとデータセットに依存していることに注意してください。これらのデータフィードとデータセットは、(カスタマーアグリーメントで定義されている) サードパーティーのコンテンツであり、現状のまま提供されます。詳細については、「AWS サービス条件」を参照してください。

AWS IoT Core for LoRaWAN を使用して静止位置データを指定するか、サードパーティーのソルバーでリアルタイムにデバイスの位置を特定するための測位を有効化できます。位置情報

は、LoRaWAN デバイスとゲートウェイのどちらか、あるいは両方に追加したり更新したりすることができます。

また、位置情報は、デバイスやゲートウェイを AWS IoT Core for LoRaWAN に追加する場合、またはデバイスやゲートウェイの設定詳細を編集する場合に指定します。位置情報は GeoJSON ペイロードとして指定されます。GeoJSON 形式は、地理データ構造をエンコードするために使用される形式です。ペイロードには、世界測地系 (座標系) (WGS84) に基づくデバイス位置の緯度と経度の座標が含まれています。

ソルバーによってリソースの位置を計算すると、Amazon Location Service を使用している場合は、リソースの位置が表示される Amazon Location マップを有効化できます。位置データを使用すると、次のことが可能になります。

- 測位を有効にして、LoRaWAN デバイスの位置を特定して取得する。
- ゲートウェイとデバイスの位置を追跡して監視する。
- 位置データへの更新を処理し、他の AWS のサービス にルーティングする AWS IoT ルールを定義する。ルールアクションのリストについては、AWS IoT デベロッパーガイドの「AWS IoT ルールアクション」を参照してください。
- 位置データと Amazon SNS を使用して、異常なアクティビティが発生した場合に備えてアラートを作成し、デバイスで通知を受け取る。

LoRaWAN デバイスの測位の仕組み

測位機能を有効にすると、サードパーティーの Wi-Fi および GNSS ソルバーを使用してデバイスの位置を特定できます。この情報は、デバイスの追跡と監視に使用できます。以下の手順は、LoRaWAN デバイスの測位を有効にして位置情報を表示する方法を示しています。

> **Note**
>
> サードパーティーのソルバーは、LoRa Edge チップを搭載した LoRaWAN デバイスでのみ使用できます。LoRaWAN ゲートウェイでは使用できません。ゲートウェイの場合でも、静止位置情報を指定することで、Amazon Location マップで位置を特定できます。

1. デバイスを追加する

   測位を有効にするには、最初にデバイスを AWS IoT Core for LoRaWAN に追加します。LoRaWAN デバイスには、ジオロケーションアプリケーションを対象とした長距離 LoRa ト

ランシーバー、マルチコンステレーション GNSS スキャナー、パッシブ Wi-Fi MAC スキャナーを統合した超低電力プラットフォームである LoRa Edge チップセットを搭載する必要があります。

2. 測位を有効にする

   デバイスの位置をリアルタイムで取得するには、測位を有効にします。LoRaWAN デバイスがアップリンクメッセージを送信すると、メッセージに含まれる Wi-Fi および GNSS スキャンデータが、ジオロケーションフレームポートを使用して AWS IoT Core for LoRaWAN に送信されます。

3. 位置情報を取得する

   トランシーバーからのスキャン結果を基に計算されたソルバーから、デバイスの推定位置を取得します。Wi-Fi と GNSS の両方のスキャン結果を使用して位置情報を計算した場合、AWS IoT Core for LoRaWAN はさらに精度の高い推定位置を選択します。

4. 位置情報を表示する

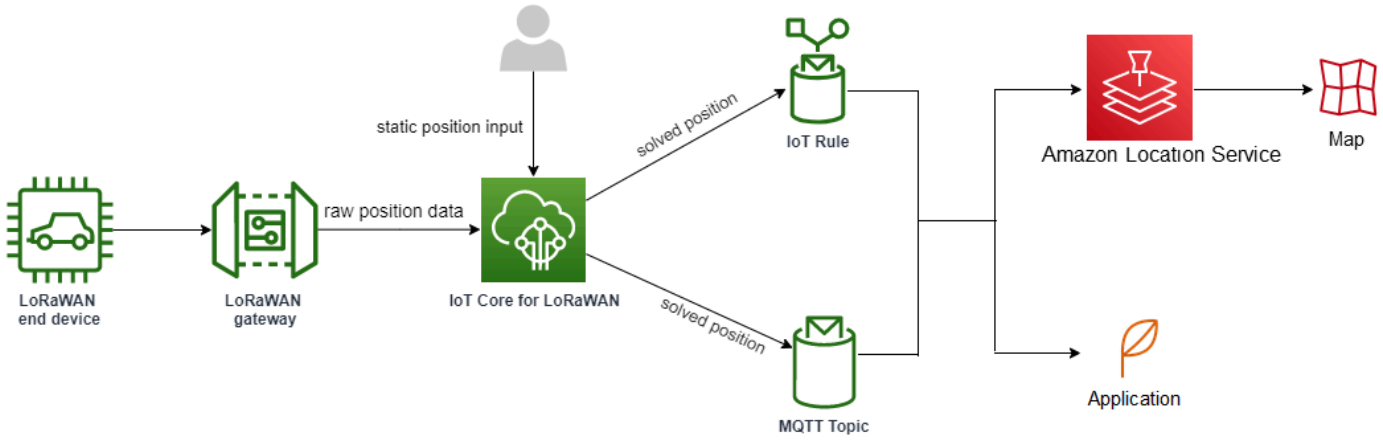
   ソルバーで位置情報が計算されると、計算した位置と入力した静止位置情報との差を示す精度情報も表示されます。デバイスの位置は Amazon Location マップでも確認できます。

   > **Note**
   >
   > ソルバーは LoRaWAN ゲートウェイには使用できないため、精度情報は `0.0` と表示されます。

測位ソルバーに使用されるアップリンクメッセージの形式と周波数ポートの詳細については、「AWS IoT Core for LoRaWAN からルールエンジンへのアップリンクメッセージ」を参照してください。

測位ワークフローの概要

次の図は、AWS IoT Core for LoRaWAN で、デバイスとゲートウェイの位置情報が保存および更新される方法を示しています。



1. リソースの静止位置を指定する

   緯度と経度の座標を使用して、GeoJSON ペイロードとしてデバイスまたはゲートウェイの静止位置情報を指定します。オプションで高度座標を指定することもできます。使用する座標は WGS84 座標系に基づきます。詳細については、「World Geodetic System (WGS84)」(世界測地系 (WGS84)) を参照してください。

2. デバイスの測位を有効にする

   LoRa Edge チップを搭載した LoRaWAN デバイスを使用している場合、オプションで測位を有効にして、デバイスの位置をリアルタイムで追跡できます。デバイスがアップリンクメッセージを送信すると、ジオロケーションフレームポートを使用して、GNSS および Wi-Fi スキャンデータが AWS IoT Core for LoRaWAN に送信されます。次に、ソルバーでは、この情報を使用してデバイスの位置を決定します。

3. 送信先を追加して位置データをルーティングする

   デバイスデータを処理するための IoT ルールを記述する送信先を追加することで、更新された位置情報を AWS IoT Core for LoRaWAN にルーティングすることができます。また、Amazon Location マップでリソースの最新の位置を確認することもできます。

リソースの位置を設定する

リソースの位置は AWS Management Console、AWS IoT Wireless API、または AWS CLI を使用して設定できます。

デバイスに LoRa Edge チップが搭載されている場合は、測位を有効にしてリアルタイムの位置情報を計算することもできます。ゲートウェイの場合は、静止位置座標を入力し、Amazon Location を使用して Amazon Location マップでゲートウェイの位置を追跡できます。

トピック

LoRaWAN ゲートウェイの位置を設定する

AWS IoT Core for LoRaWAN にゲートウェイを追加すると、静止位置データを指定できます。Amazon Location Service マップを有効にすると、位置データが Amazon Location マップに表示されます。

> **Note**
>
> サードパーティーソルバーは LoRaWAN ゲートウェイでは使用できません。ゲートウェイの場合でも、静止位置座標を指定できます。ゲートウェイの場合のように、ソルバーを使用して位置を計算しない場合、精度情報は `0.0` と表示されます。

ゲートウェイの位置は、AWS Management Console、AWS IoT Wireless API、または AWS CLI を使用して設定できます。

コンソールを使用してゲートウェイの位置を設定する

AWS Management Console を使用してゲートウェイリソースの位置を設定するには、まずコンソールにサインインし、AWS IoT コンソールの [Gateways] (ゲートウェイ) ハブページに移動します。

位置情報を追加する

ゲートウェイの位置設定を追加するには

1. [Gateways] (ゲートウェイ) ハブページで、[Add gateway] (ゲートウェイの追加) を選択します。
2. ゲートウェイの EUI、周波数帯域 (RFRegion)、およびその他のゲートウェイの詳細と LoRaWAN 設定情報を入力します。詳細については、「コンソールを使用してゲートウェイを追加する」を参照してください。

3. [Position information - Optional] (位置情報 - オプション) セクションに移動して、緯度と経度の座標と、オプションの高度座標を使用して、ゲートウェイの位置情報を入力します。位置情報は WGS84 座標系に基づいています。

ゲートウェイの位置を表示する

ゲートウェイの位置を設定したら、AWS IoT Core for LoRaWAN は `iotwireless.map` という Amazon Location マップを作成します。このマップは、[Position] (位置) タブのゲートウェイの詳細ページで確認できます。指定した位置座標に基づいて、ゲートウェイの位置がマップ上にマーカーとして表示されます。ズームインまたはズームアウトして、マップ上のゲートウェイの位置をはっきりと表示できます。[Position] (位置) タブには、精度情報とゲートウェイの位置を決定した際のタイムスタンプも表示されます。

> **Note**
>
> Amazon Location Service のマップがインストールされていない場合は、マップにアクセスしてゲートウェイの位置を表示するには、Amazon Location Service を使用する必要があることを示すメッセージが表示されます。Amazon Location Service マップを使用すると、AWS アカウント に追加料金が発生する可能性があります。詳細については、「AWS IoT Core 料金表」を参照してください。

`iotwireless.map` は、`GetMapTile` といった `Get` API オペレーションを使用してアクセスされるマップデータのソースとして機能します。マップに使用される `Get` API に関する情報は、「Amazon Location Service API リファレンス」を参照してください。

このマップに関する追加情報を取得するには、Amazon Location Service コンソールに移動し、[maps] (マップ) を選択してから、iotwireless.map を選択します。詳細については、「Amazon Location Service デベロッパーガイド」の「マップ」を参照してください。

ゲートウェイの位置設定を更新する

ゲートウェイの位置設定を変更するには、ゲートウェイの詳細ページで [Edit] (編集) を選択し、位置情報と送信先を更新します。

> **Note**
>
> 過去の位置データに関する情報はありません。ゲートウェイの位置座標を更新すると、以前に報告された位置データが上書きされます。位置を更新したら、ゲートウェイ詳細の

[Position] (位置) タブに新しい位置情報が表示されます。タイムスタンプの変更は、それがゲートウェイの最新位置に対応していることを示しています。

API を使用してゲートウェイの位置を設定する

位置情報の指定や、ゲートウェイの位置を設定するには、AWS IoT Wireless API か AWS CLI を使用します。

> **Important**
>
> API アクションの UpdatePosition、GetPosition、PutPositionConfiguration、GetPositionConfiguration、ListPositionConfi はサポートされなくなりました。代わりに、測位情報を更新および取得するための呼び出しには、GetResourcePosition と UpdateResourcePosition API オペレーションを使用する必要があります。

位置情報を追加する

特定のワイヤレスゲートウェイの静止位置情報を追加するには、UpdateResourcePosition API オペレーションか update-resource-position CLI コマンドを使用して、座標を指定します。`ResourceType` に `WirelessGateway`、`ResourceIdentifier` に更新するワイヤレスゲートウェイの ID、GeoJSON ペイロードに位置情報を指定します。

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

以下は、*gatewayposition.json* ファイルの内容を示しています。

gatewayposition.json の内容

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

```
      }
}
```

このコマンドを実行しても、出力は生成されません。指定した位置情報を表示するには、GetResourcePosition API オペレーションを使用します。

位置情報を取得する

特定のワイヤレスゲートウェイの位置情報を取得するには、GetResourcePosition API オペレーションか get-resource-position CLI コマンドを使用します。resourceType に WirelessGateway を指定し、resourceIdentifier にワイヤレスゲートウェイの ID を入力します。

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

このコマンドを実行すると、ワイヤレスゲートウェイの位置情報が GeoJSON ペイロードとして表示されます。位置座標、位置座標タイプ、ゲートウェイの最新位置に対応するタイムスタンプなどの追加プロパティの情報が表示されます。

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

LoRaWAN デバイスの位置を設定する

デバイスを AWS IoT Core for LoRaWAN に追加すると、静止位置情報の指定、測位の有効化 (オプション)、送信先の指定が可能になります。送信先には、デバイスの位置情報を処理し、更新された位置を Amazon Location Service にルーティングする IoT ルールを記述します。デバイスの位置を設定すると、精度情報、指定した送信先とともに、位置データが Amazon Location マップに表示されます。

デバイスの位置の設定には、AWS Management Console、AWS IoT Wireless API、または AWS CLI を使用できます。

アップリンクメッセージのフレームポートと形式

ポジショニングを有効にする場合は、デバイスから AWS IoT Core for LoRaWAN に Wi-Fi および GNSS スキャンデータを通信するため、ジオロケーションフレームポートを指定する必要があります。位置情報は、このフレームポートを使用して AWS IoT Core for LoRaWAN に送信されます。

LoRaWAN は、データ配信フィールド (FRMPayload) と Port フィールド (FPort) を備え、異なるタイプのメッセージを区別する仕様になっています。位置情報を送信するために、フレームポートには 1 から 223 までの値を指定できます。FPort 0 は MAC メッセージ用に、FPort 224 は MAC コンプライアンステスト用に、ポート 225〜255 は将来的なアプリケーション拡張の標準化に向けて予約されています。

AWS IoT Core for LoRaWAN からルールエンジンへのアップリンクメッセージ

送信先を追加すると、ルールエンジンを使用してデータを Amazon Location Service にルーティングするための AWS IoT ルールが作成されます。その後、更新された位置情報が Amazon Location マップに表示されます。測位を有効化していない場合、デバイスの静止位置座標を更新すると、送信先で位置データがルーティングされます。

次のコードは、AWS IoT Core for LoRaWAN から送信されたアップリンクメッセージの形式を、位置情報、精度、ソルバー設定、ワイヤレスメタデータとともに示しています。強調表示されているフィールドはオプションです。垂直精度情報がない場合、値は `null` になります。

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

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

    //Parameters controlled by AWS IoT Core for LoRaWAN
```

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

        }
    }
}
```

コンソールを使用してデバイスの位置を設定する

AWS Management Console を使用してデバイスの位置を設定および管理するには、まずコンソールにサインインし、次に AWS IoT コンソールの [Devices] (デバイス) ハブページに移動します。

位置情報を追加する

デバイスの位置情報を追加するには、次の手順を実行します。

1. [Devices] (デバイス) ハブページで、[Add wireless device] (ワイヤレスデバイスの追加) を選択します。
2. ワイヤレスデバイスの仕様、デバイスとサービスのプロファイル、およびデータを他の AWS のサービス にルーティングするための IoT ルールを定義する送信先を入力します。詳細については、「デバイスを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。
3. 位置情報を入力して、ジオロケーションを有効化し (オプション)、メッセージのルーティングに使用する位置データの送信先を指定します。

   - 位置情報

     緯度と経度の座標、およびオプションで高度座標を使用して、デバイスの位置データを指定します。位置情報は WGS84 座標系に基づいています。

   - ジオロケーション

     AWS IoT Core for LoRaWAN でデバイスの位置を計算するためにジオロケーションを使用する場合は、ポジショニングを有効化します。サードパーティーの GNSS および Wi-Fi ソルバーを使用して、デバイスの位置をリアルタイムで特定します。

     ジオロケーション情報を入力するには、[ポジショニングを有効化] を選択し、GNSS および Wi-Fi スキャンデータを AWS IoT Core for LoRaWAN と通信するための位置情報フレームポートを入力します。参照用にデフォルトの FPort が入力されていますが、1 から 223 までの任意の値を選択できます。

   - 位置データの送信先

     デバイスの位置データを処理して AWS IoT Core for LoRaWAN に転送する AWS IoT ルールを記述する送信先を選択します。この送信先は、位置データのルーティングのみに使用されます。これは、デバイスデータを他の AWS のサービス にルーティングするために使用するものとは異なる送信先である必要があります。

デバイスの位置設定を表示する

デバイスの位置を設定したら、AWS IoT Core for LoRaWAN は、`iotwireless.map` という Amazon Location マップを作成します。このマップは、デバイスの [Position] (位置) タブの詳細ページで確認できます。指定した位置座標またはサードパーティーソルバーによって計算された位置に基づいて、デバイスの位置がマップ上にマーカーとして表示されます。ズームインまたはズームアウトすることで、マップ上のデバイスの位置をはっきりと表示できます。デバイスの詳細ページの [Position] (位置) タブには、精度情報、デバイスの位置が決定されたタイムスタンプ、および指定した位置データの送信先も表示されます。

> **Note**
>
> Amazon Location Service の地図を有効にしていない場合、地図にアクセスして位置を表示するには、Amazon Location Service を使用する必要があることを示すメッセージが表示されます。Amazon Location Service マップを使用すると、AWS アカウント に追加料金が発生する可能性があります。詳細については、「AWS IoT Core 料金表」を参照してください。

`iotwireless.map` は、`GetMapTile` といった `Get` API オペレーションを使用してアクセスされるマップデータのソースとして機能します。マップに使用される `Get` API に関する情報は、「Amazon Location Service API リファレンス」を参照してください。

このマップに関する追加情報を取得するには、Amazon Location Service コンソールに移動し、[maps] (マップ) を選択してから、iotwireless.map を選択します。詳細については、「Amazon Location Service デベロッパーガイド」の「マップ」を参照してください。

デバイスの位置設定を更新する

デバイスの位置設定を変更するには、デバイスの詳細ページで [Edit] (編集) を選択し、次に、位置情報と、ジオロケーション設定、送信先を更新します。

> **Note**
>
> 過去の位置データに関する情報はありません。デバイスの位置座標を更新すると、以前に報告された位置データが上書きされます。ポジションを更新したら、デバイス詳細の [Position] (位置) タブに、新しい位置情報が表示されます。タイムスタンプの変更は、それがデバイスの最新位置に対応していることを示しています。

API を使用してデバイスの位置を設定する

AWS IoT Wireless API または AWS CLI を使用することで、位置情報の指定、デバイス位置の設定、ジオロケーションの有効化 (オプション) が可能になります。

> **Important**
>
> API アクションの UpdatePosition、GetPosition、PutPositionConfiguration、GetPositionConfiguration、ListPositionConfi はサポートされなくなりました。代わりに、測位情報を更新および取得するための呼び出し

には、GetResourcePosition と UpdateResourcePosition API オペレーションを使用する必要があります。

位置情報と設定を追加する

特定のワイヤレスデバイスの静止位置情報を追加するには、UpdateResourcePosition API オペレーションか、update-resource-position CLI コマンドを使用して、座標を指定します。ResourceType に WirelessDevice、ResourceIdentifier に更新するワイヤレスデバイス ID、位置情報を指定します。

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

以下は、*deviceposition.json* ファイルの内容を示しています。ジオロケーション情報データを送信するための FPort 値を指定するには、CreateWirelessDevice および UpdateWirelessDevice API オペレーションで測位オブジェクトを使用します。

deviceposition.json の内容

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

このコマンドを実行しても、出力は生成されません。指定した位置情報を表示するには、GetResourcePosition API オペレーションを使用します。

位置情報と設定を取得する

特定のワイヤレスデバイスの位置情報を取得するには、GetResourcePosition API または get-resource-position CLI コマンドを使用します。resourceType に WirelessDevice を指定し、また resourceIdentifier にワイヤレスデバイスの ID を入力します。

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

このコマンドを実行すると、ワイヤレスデバイスの位置情報が GeoJSON ペイロードとして表示されます。位置座標、位置のタイプ、精度情報、デバイスの最新位置に対応するタイムスタンプなどのプロパティに関する情報が表示されます。

```
{
     "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"
      }
}
```

AWS IoT Wireless によるゲートウェイの管理

AWS IoT Core for LoRaWAN でゲートウェイを使用する場合の重要な考慮事項を次に示します。ゲートウェイを AWS IoT Core for LoRaWAN に追加する方法については、「ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。

LoRa Basic Station ソフトウェア要件

AWS IoT Core for LoRaWAN に接続するためには、LoRaWAN ゲートウェイで LoRa Basics Station というソフトウェアが実行されている必要があります。LoRa Basics Station は、Semtech 社によって管理され、GitHub リポジトリによって配信されているオープンソースのソフトウェアです。AWS IoT Core for LoRaWAN では、LoRa Basics Station バージョン 2.0.4 以降がサポートされています。最新バージョンは 2.0.6 です。

AWS Partner Device Catalog の認定されたゲートウェイの使用

AWS Partner Device Catalog には、AWS IoT Core for LoRaWAN での使用が認定されているゲートウェイとデベロッパーキットが含まれています。ゲートウェイを AWS IoT Core に接続するための埋め込みソフトウェアを変更する必要がないため、これらの認定ゲートウェイを使用することをお勧

めします。これらのゲートウェイには、AWS IoT Core for LoRaWAN と互換性のあるバージョンの BasicStation ソフトウェアが既に搭載されています。

> **Note**
>
> AWS IoT Core for LoRaWAN の認定ゲートウェイとして Partner Catalog に記載されていないゲートウェイをお持ちの場合、そのゲートウェイがバージョン 2.0.4 以降の LoRa Basics Station ソフトウェアを実行していれば、使用できる可能性があります。LoRaWAN ゲートウェイの認証に TLS Server and Client Authentication を使用していることを確認してください。

CUPS および LNS プロトコルの使用

LoRa Basics Station ソフトウェアには、ネットワークサーバーにゲートウェイを接続するための 2 つのサブプロトコル、LoRaWAN Network Server (LNS) プロトコルと Configuration and Update Server (CUPS) プロトコルが含まれています。

LNS プロトコルは、LoRa Basics Station 互換ゲートウェイとネットワークサーバーの間にデータ接続を確立します。LoRa アップリンクおよびダウンリンクメッセージは、安全な WebSockets を介して、このデータ接続経由で交換されます。

CUPS プロトコルは、認証情報の管理、ゲートウェイのリモート設定とファームウェアの更新を可能にします。AWS IoT Core for LoRaWAN は、LoRaWAN データの取り込みとリモートゲートウェイの管理それぞれに対して LNS エンドポイントと CUPS エンドポイントの両方を提供します。

詳細については、「LNS プロトコル」および「CUPS プロトコル」を参照してください。

トピック

- LoRaWAN ゲートウェイのビーコンとフィルタリング機能を設定
- AWS IoT Core for LoRaWAN で CUPS サービスを使用してゲートウェイファームウェアを更新する
- LoRaWAN ダウンリンクデータトラフィックを受信するゲートウェイの選択

LoRaWAN ゲートウェイのビーコンとフィルタリング機能を設定

LoRaWAN デバイスを使用する場合、LoRaWAN ゲートウェイの特定のオプションパラメータを設定できます。パラメータには以下が含まれます。

- ビーコン

  クラス B LoRaWAN デバイスのブリッジとして機能する LoRaWAN ゲートウェイのビーコンパラメータを設定できます。これらのデバイスはスケジュールされた時間帯にダウンリンクメッセージを受信するため、これらの時間同期ビーコンを送信するようにゲートウェイのビーコンパラメータを設定する必要があります。
- フィルタリング

  LoRaWAN ゲートウェイの `NetID` および `JoinEUI` パラメータを設定することで、デバイスのデータトラフィックをフィルタリングすることができます。トラフィックをフィルタリングすることで、帯域幅の使用量を節約し、ゲートウェイと LNS 間のトラフィックフローを削減することができます。
- サブバンド

  ゲートウェイのサブバンドを設定して、使用する特定のサブバンドを指定できます。さまざまなサブバンド間をホップできないワイヤレスデバイスでは、この機能を使用して、その特定のサブバンドの周波数チャネルのみを使用してデバイスと通信できます。

次のトピックでは、これらのパラメータとその設定方法について詳しく説明します。ビーコンのパラメータは、AWS Management Console では使用できず、AWS IoT Wireless API または AWS CLI を使用してのみ指定できます。

トピック

- クラス B デバイスにビーコンを送信するようにゲートウェイを設定する
- ゲートウェイのサブバンドとフィルタリング機能を設定する

クラス B デバイスにビーコンを送信するようにゲートウェイを設定する

クラス B ワイヤレスデバイスを AWS IoT Core for LoRaWAN にオンボードすると、デバイスはスケジュールされたタイムスロットにダウンリンクメッセージを受信します。デバイスは、ゲートウェイから送信される時間同期ビーコンに基づいてこれらのスロットを開きます。ゲートウェイがこれらの時間同期ビーコンを送信するには、AWS IoT Core for LoRaWAN を使用してゲートウェイの特定のビーコン関連パラメータを設定します。

これらのビーコンパラメータを設定するには、ゲートウェイが LoRa Basics Station ソフトウェアバージョン 2.0.6 を実行している必要があります。「AWS Partner Device Catalog の認定されたゲートウェイの使用」を参照してください。

ビーコンパラメータを設定する方法

> **Note**
>
> ゲートウェイがクラス B ワイヤレスデバイスと通信している場合のみ、ビーコンパラメータを設定する必要があります。

CreateWirelessGateway API オペレーションを使用して AWS IoT Core for LoRaWAN にゲートウェイを追加する際にビーコンパラメータの設定を行います。API オペレーションを呼び出す際には、ゲートウェイの Beaconing オブジェクトを使用して、以下のパラメータを指定します。パラメータを設定すると、ゲートウェイは 128 秒間隔でデバイスにビーコンを送信します。

- DataRate: ビーコンを送信しているゲートウェイのデータレート。
- Frequencies: ゲートウェイがビーコンを送信する周波数のリスト。

次の例は、ゲートウェイのパラメータを設定する方法を示しています。input.json ファイルにはゲートウェイ証明書やプロビジョニング認証情報などの追加の詳細が含まれます。CreateWirelessGatewayAPI オペレーションを使用して AWS IoT Core for LoRaWAN にゲートウェイを追加する方法の詳細については、「API を使用してゲートウェイを追加する」を参照してください。

> **Note**
>
> ビーコンのパラメータは、AWS IoT コンソールを使用してゲートウェイを AWS IoT Core for LoRaWAN に追加する際には利用できません。

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

以下は、input.json ファイルの内容を示しています。

input.json の内容

```
{
```

```
    "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]
    }
}
```

以下のコードは、このコマンドを実行したときの出力例を示しています。

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

ビーコンパラメータに関する情報を取得する

ゲートウェイのビーコンパラメータに関する情報は、GetWirelessGateway API オペレーションを使用して取得できます。

> **Note**
>
> ゲートウェイがすでにオンボーディングされている場合は、ビーコンパラメータの設定に UpdateWirelessGateway API オペレーションを使用することはできません。パラメータを設定するには、ゲートウェイを削除してから、CreateWirelessGateway API オペレーションを使用してゲートウェイを追加するときにパラメータを指定する必要があります。

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

```
    --identifier-type WirelessGatewayId
```

このコマンドを実行すると、ゲートウェイとビーコンパラメータに関する情報が返されます。

ゲートウェイのサブバンドとフィルタリング機能を設定する

LoRaWAN ゲートウェイは、ゲートウェイが AWS IoT Core for LoRaWAN に接続できるようにする LoRa Basics Station ソフトウェアを実行します。AWS IoT Core for LoRaWAN に接続するために、LoRa ゲートウェイはまず CUPS サーバーに LNS エンドポイントをクエリし、そのエンドポイントとの WebSockets データ接続を確立します。接続が確立されたら、その接続を介してアップリンクフレームとダウンリンクフレームを交換できます。

ゲートウェイが受信する LoRa データフレームのフィルタリング

LoRaWAN ゲートウェイがエンドポイントへの接続を確立した後、AWS IoT Core for LoRaWAN は、LoRa ゲートウェイの設定の一連のパラメータを指定する `router_config` メッセージで応答します。これには、`NetID` や `JoinEui` のフィルタリングパラメータが含まれます。`router_config` についてと、LoRaWAN Network Server (LNS) との接続が確立される方法については、LNS プロトコルを参照してください。

```
{
"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
}
```

ゲートウェイは、通常、Wi-Fi、イーサネット、またはセルラーなどの広帯域幅ネットワーク経由で、LoRaWAN デバイスデータを LNS との間で送受信します。ゲートウェイは通常、すべてのメッセージをピックアップし、ゲートウェイに送信されるトラフィックを AWS IoT Core for LoRaWAN に渡します。ただし、一部のデバイスデータトラフィックをフィルターするようにゲートウェイを設定できます。これにより、帯域幅の使用量を節約し、ゲートウェイと LNS 間のトラフィックフローを削減できます。

データフレームをフィルターするように LoRa ゲートウェイを設定するには、`router_config` メッセージ内のパラメータ `NetID` および `JoinEui` を使用できます。`NetID` は、受け入れられる NetID 値のリストです。リストされているもの以外のデータフレームを含む LoRa データフレームは除外されます。`JoinEui` は、JoinEUI 値の範囲をエンコードする整数値のペアのリストです。参加要求フレームは、メッセージ内の `JoinEui` フィールドが [BegEui,EndEui] の範囲にない限り、ゲートウェイによって除外されます。

周波数チャネルとサブバンド

US915 および AU915 の RF リージョンでは、ワイヤレスデバイスには 64 個の 125 KHz アップリンクチャネルと 8 個の 500 KHz アップリンクチャネルがあり、LoRa ゲートウェイを使用して LoRaWAN ネットワークにアクセスできます。アップリンク周波数チャネルは 8 つのサブバンドに分かれており、それぞれに 8 つの 125 KHz チャネルと 1 つの 500 KHz チャネルがあります。AU915 リージョンの通常ゲートウェイごとに、1 つ以上のサブバンドがサポートされます。

一部のワイヤレスデバイスはサブバンド間でホップできず、AWS IoT Core for LoRaWAN への接続時に 1 つのサブバンドでのみ周波数チャネルを使用します。これらのデバイスからのアップリンクパケットを送信するには、LoRa ゲートウェイをその特定のサブバンドを使用するように設定します。EU868 など、他の RF リージョンのゲートウェイでは、この設定は必要ありません。

コンソールを使用して、フィルタリングおよびサブバンドを使用するようにゲートウェイを設定する

特定のサブバンドを使用するようにゲートウェイを設定したり、LoRa データフレームをフィルタリングする機能を有効にしたりすることができます。コンソールを使用してこれらのパラメータを指定するには、次のようにします。

1. AWS IoT コンソールの AWS IoT Core for LoRaWAN の [Gateways] (ゲートウェイ) ページに移動し、[Add gateway] (ゲートウェイの追加) を選択します。
2. [Gateway's Eui] (ゲートウェイのEUI)、[Frequency band (RFRegion)] (周波数帯 (RFRegion))、オプションで [Name] (名前) および [Description] (説明) など、ゲートウェイの詳細を指定して、AWS IoT のモノをゲートウェイに関連付けるかどうかを選択します。ゲートウェイを追加する方法については、「コンソールを使用してゲートウェイを追加する」を参照してください。
3. [LoRaWAN configuration] (LoRaWAN 設定) セクションでは、サブバンドとフィルタリング情報を指定できます。
   - `SubBands`: サブバンドを追加するには、Add SubBandを選択し、ゲートウェイでサポートされているサブバンドを示す整数値のリストを指定します。`SubBands` パラメータは、`RfRegion` US915 および AU915 でのみ設定でき、これらのサポート対象リージョン内のいずれかの `[1,8]` の範囲内にある値を持つ必要があります。

- `NetIdFilters`: アップリンクフレームをフィルターするには、[Add NetId] (NetID の追加) をクリックし、ゲートウェイが使用する文字列値のリストを指定します。ワイヤレスデバイスからの着信アップリンクフレームの NetID は、リストされている値の少なくとも 1 つと一致する必要があります。一致しない場合、フレームは除外されます。
- `JoinEuiFilters`: [Add JoinEui range] (参加範囲の追加) を選択し、ゲートウェイが LoRa フレームをフィルターするために使用する文字列値のペアのリストを指定します。ワイヤレスデバイスからの参加要求の一部として指定された JoinEUI 値は、それぞれが [BegEui, EndEui] のペアとしてリストされた、少なくとも 1 つの JoinEuiRange 値の範囲内にある必要があります。そうでない場合、フレームは除外されます。

4. 「コンソールを使用してゲートウェイを追加する」で説明されている手順に従って、ゲートウェイの設定を続行できます。

ゲートウェイを追加した後、AWS IoT コンソールで AWS IoT Core for LoRaWAN の [Gateways] (ゲートウェイ)ページで、追加したゲートウェイを選択すると、[Gateway details] (ゲートウェイの詳細) ページの [LoRaWAN specific details] (LoRaWAN 固有の詳細) セクションに SubBands、フィルター `NetIdFilters` および `JoinEuiFilters` が表示されます。

API を使用して、フィルタリングおよびサブバンドを使用するようにゲートウェイを設定する

ゲートウェイの作成に使用する CreateWirelessGateway API を使用して、使用するサブバンドを設定し、フィルタリング機能を有効にすることができます。`CreateWirelessGateway` API を使用すると、LoRaWAN フィールドを使用して指定したゲートウェイ設定情報の一部としてサブバンドとフィルターを指定できます。以下に、この情報を含むリクエストトークンを示します。

```
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
}
```

また、UpdateWirelessGateway API を使用して、サブバンドは更新せずにフィルターを更新することができます。JoinEuiFilters と NetIdfilters の値が null の場合、そのフィールドに更新がないことを意味します。値が null でなく、空のリストが含まれている場合は、更新が適用されています。指定したフィールドの値を取得するには、GetWirelessGateway API を使用します。

AWS IoT Core for LoRaWAN で CUPS サービスを使用してゲートウェイファームウェアを更新する

ゲートウェイで実行される LoRa Basics Station ソフトウェアは、Configuration and Update Server (CUPS) プロトコルを使用して、認証情報管理とファームウェア更新インターフェイスを提供します。CUPS プロトコルは、ECDSA シグニチャによるセキュアなファームウェアアップデート配信を提供します。

ゲートウェイのファームウェアは頻繁に更新する必要があります。AWS IoT Core for LoRaWAN で CUPS サービスを使用して、ファームウェアの更新をゲートウェイに提供できます。また、更新に署名することもできます。ゲートウェイのファームウェアを更新するには、SDK または CLI を使用できますが、コンソールは使用できません。

更新プロセスは完了までに約 45 分かかります。AWS IoT Core for LoRaWAN への接続用にゲートウェイを初めて設定する場合、さらに時間がかかることがあります。ゲートウェイの製造元は、通常、独自のファームウェア更新ファイルと署名を提供しているため、代わりにそれを使用して「S3 バケットにファームウェアファイルをアップロードし、IAM ロールを追加する」に進むことができます。

ファームウェア更新ファイルがない場合は、「ファームウェア更新ファイルと署名を生成する」で、アプリケーションに適応させるために使用できる例を参照してください。

ゲートウェイのファームウェア更新を実行するには、次の手順に従います。

- ファームウェア更新ファイルと署名を生成する
- S3 バケットにファームウェアファイルをアップロードし、IAM ロールを追加する
- タスク定義を使用してファームウェア更新をスケジュールし、実行する

ファームウェア更新ファイルと署名を生成する

この手順のステップはオプションで、使用しているゲートウェイによって異なります。ゲートウェイの製造元は、独自のファームウェア更新を更新ファイルまたはスクリプトの形式で提供し、Basics Station はこのスクリプトをバックグラウンドで実行します。この場合、ほとんどの場合、使用しているゲートウェイのリリースノートにファームウェア更新ファイルが記載されています。その更新ファイルまたはスクリプトを代わりに使用して、「S3 バケットにファームウェアファイルをアップロードし、IAM ロールを追加する」に進むことができます。

このスクリプトがない場合は、以下に示す、ファームウェア更新ファイルを生成するために実行するコマンドを使用してください。また、更新に署名して、コードが変更および破損していないこと、信頼できる作成者のみが発行したコードがデバイスで実行されるようにすることもできます。

この手順では、次の操作を行います。

- ファームウェア更新ファイルを生成する
- ファームウェア更新の署名を生成する
- 次のステップを確認する

ファームウェア更新ファイルを生成する

ゲートウェイで動作している LoRa Basics Station ソフトウェアは、CUPS 応答でファームウェア更新を受信できます。製造元から提供されたスクリプトがない場合は、Raspberry Pi ベースの RAKWireless ゲートウェイ用に書かれた次のファームウェア更新スクリプトを参照してください。基本スクリプトと、新しいステーションバイナリ、バージョンファイル、`station.conf` が添付されています。

> **Note**
>
> スクリプトは RAKWireless ゲートウェイに固有のものなので、使用しているゲートウェイに応じてアプリケーションに適応させる必要があります。

基本スクリプト

以下は、Raspberry Pi ベースの RAKWireless ゲートウェイのサンプル基本スクリプトを示しています。次のコマンドを `base.sh` ファイルに保存し、端末で、Raspberry Pi のウェブブラウザからスクリプトを実行できます。

```
*#!/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
```

ペイロードスクリプトを追加する

基本スクリプトに、Basics Station バイナリ、更新するバージョンを識別する version.txt、および `station.conf` を追加して、`addpayload.sh` というスクリプトにしました。次に、このスクリプトを実行します。

```
*#!/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
```

これらのスクリプトを実行したら、端末で次のコマンドを実行して、ファームウェア更新ファイル fwstation を生成します。

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

ファームウェア更新の署名を生成する

LoRa Basics Stationソフトウェアは、ECDSA 署名付きファームウェア更新を提供します。署名付き更新をサポートするには、以下が必要です。

- ECDSA プライベートキーによって生成され、128 バイト未満である必要がある署名。
- 署名に使用されるプライベートキー。sig-%d.key の形式のファイル名を持つゲートウェイに格納する必要があります。ファイル名 sig-0.key を使用することをお勧めします。
- プライベートキーを介した 32 ビットの CRC。

署名と CRC は AWS IoT Core for LoRaWAN API に渡されます。前述のファイルを生成するには、次の gen.sh スクリプトを使用できます。これは、GitHub リポジトリの basicstation の例から着想を得たものです。

```
*#!/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
```

スクリプトによって生成されたプライベートキーは、ゲートウェイに保存する必要があります。キーファイルはバイナリ形式です。

```
./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
```

次のステップを確認する

ファームウェアと署名の生成が完了したので、次のトピックに進み、ファームウェアファイル `fwstation` を Amazon S3 バケットにアップロードします。バケットは、ファームウェア更新ファイルをオブジェクトとして格納するコンテナです。S3 バケット内のファームウェア更新ファイルを読み取る許可を CUPS サーバーに付与する IAM ロールを追加できます。

S3 バケットにファームウェアファイルをアップロードし、IAM ロールを追加する

Amazon S3 を使用して、バケットを作成します。バケットは、ファームウェア更新ファイルを保存できるコンテナです。ファイルを S3 バケットにアップロードし、CUPS サーバーがバケットから更新ファイルを読み取ることを許可する IAM ロールを追加できます。Amazon S3 の詳細については、「Amazon S3 の使用開始」を参照してください。

アップロードするファームウェア更新ファイルは、使用しているゲートウェイによって異なります。「ファームウェア更新ファイルと署名を生成する」で説明されている手順と同様の手順に従った場合は、スクリプトを実行して生成された `fwstation` ファイルをアップロードします。

この手順の完了には 20 分ほどかかります。

ファームウェアファイルをアップロードするには、以下を行います。

- Amazon S3 バケットを作成し、更新ファイルをアップロードする
- S3 バケットを読み取る許可を持つ IAM ロールを作成する
- 次のステップを確認する

Amazon S3 バケットを作成し、更新ファイルをアップロードする

AWS Management Console を使用して Amazon S3 バケットを作成し、次に、ファームウェア更新ファイルをバケットにアップロードします。

S3 バケットの作成

S3 バケットを作成するには、Amazon S3 コンソールを開きます。サインインしていない場合はサインインしてから、次の手順を実行します。

1. [バケットを作成する] を選択します。
2. [Bucket name] (バケット名) に、一意のわかりやすい名前を入力します (例: `iotwirelessfwupdate`)。バケットの推奨命名規則については、https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html を参照してください。
3. LoRaWAN ゲートウェイとデバイスの作成に使用するものとして選択されている AWS リージョン を選択したことと、[Block all public access] (すべてのパブリックアクセスをブロック) 設定が選択され、バケットがデフォルトのアクセス権限を使用するようになっていることを確認してください。

4. [Bucket versioning] (バケットバージョニング) で [Enable] (有効) を選択します。これは、複数のバージョンのファームウェア更新ファイルを同じバケット内に保持するうえで役立ちます。
5. [Server-side encryption] (サーバー側の暗号化) が [Disable] (無効) に設定されていることを確認して、[Create bucket] (バケットを作成する) を選択します。

ファームウェアの更新ファイルをアップロードする

これで、AWS Management Console に表示されているバケットのリストに、自分のバケットが表示されます。バケットを選択し、次の手順を実行してファイルをアップロードします。

1. バケットを選択し、[Upload] (アップロード) を選択します。
2. [Add file] (ファイルを追加する) を選択し、ファームウェア更新ファイルをアップロードします。「ファームウェア更新ファイルと署名を生成する」で説明されている手順に従った場合は、`fwstation` ファイルをアップロードします。それ以外の場合は、ゲートウェイの製造元から提供されたファイルをアップロードします。
3. すべての設定がデフォルトに設定されていることを確認します。[Predefined ACLs] (事前定義された ACL) が [private] (プライベート) に設定されていることを確認し、[Upload] (アップロード) を選択してファイルをアップロードします。
4. アップロードしたファイルの S3 URI をコピーします。バケットを選択すると、アップロードしたファイルが [Objects] (オブジェクト) に表示されます。ファイルを選択し、[Copy S3 URI (S3 URI をコピーする) を選択します。前に説明した例 (`fwstation`) と同様の名前をバケットに付けた場合、URI は `s3://iotwirelessfwupdate/fwstation` のようになります。S3 URI は IAM ロールを作成するときに使用します。

S3 バケットを読み取る許可を持つ IAM ロールを作成する

次に、S3 バケットからファームウェア更新ファイルを読み取る許可を CUPS に付与する IAM ロールとポリシーを作成します。

ロールに IAM ポリシーを作成する

AWS IoT Core for LoRaWAN 送信先ロールに IAM ポリシーを作成するには、IAM コンソールの [Policies] (ポリシー) ハブを開き、次の手順を完了します。

1. [Create policy] (ポリシーの作成) を選択し、[JSON] タブを選択します。

2. エディタからすべてのコンテンツを削除し、このポリシードキュメントを貼り付けます。ポリシーは、iotwireless バケットと、オブジェクト内に格納されているファームウェア更新ファイル fwstation へのアクセス権限を提供します。

```
{
    "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. [Review policy] (ポリシーの確認) を選択し、[Name] (名前) に、このポリシーの名前を入力します (例: IoTWirelessFwUpdatePolicy)。この名前は、次の手順で使用するために必要です。
4. [Create policy] (ポリシーの作成) を選択します。

ポリシーがアタッチされた IAM ロールを作成する

次に、IAM ロールを作成して、先ほど S3 バケットへのアクセス用に作成したポリシーをアタッチします。IAM コンソールの [Roles] (ロール) ハブを開き、以下のステップを完了します。

1. [Create role] (ロールの作成) を選択します。
2. [Select type of trusted entity] (信頼できるエンティティのタイプを選択) で、[Another AWS アカウント] (別の AWS アカウント) を選択します。
3. [Account ID] (アカウント ID) で AWS アカウント アカウント ID を入力し、[Next: Permissions] (次へ: アクセス許可) を選択します。
4. 検索ボックスで、前の手順で作成した IAM ポリシーの名前を入力します。検索結果で、先ほど作成した IAM ポリシー (例: IoTWirelessFwUpdatePolicy) を確認し、それを選択します。

5. [次へ: タグ]、[次へ: 確認] の順に選択します。
6. [Role name] (ロール名) に、このロールの名前を入力し (例: `IoTWirelessFwUpdateRole`)、[Create role] (ロールの作成) を選択します。

IAM ロールの信頼関係を編集する

前のステップを実行した後に表示される確認メッセージで、作成したロールの名前を選択して、ロールを編集します。ロールを編集して、次の信頼関係を追加します。

1. 作成したロールの [Summary] (概要) セクションで、[Trust relationships] (信頼関係) タブを選択し、続いて [Edit trust relationship] (信頼関係の編集) を選択します。
2. [Policy Document] (ポリシードキュメント) で、`Principal` プロパティを次の例のように変更します。

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

   `Principal` プロパティを変更すると、完全なポリシードキュメントは次の例のようになります。

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

3. 変更を保存して終了するには、[Update Trust Policy] (信頼ポリシーの更新) を選択します。
4. ロールの ARN を取得します。IAM ロールを選択すると、[Summary] (概要) セクションに [Role ARN] (ロールの ARN) が表示されます (例: `arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`)。この [Role ARN] (ロールの ARN) をコピーします。

次のステップを確認する

これで、S3 バケットと、CUPS サーバーが S3 バケットを読み取ることを許可する IAM ロールが作成されました。次のトピックに進み、ファームウェア更新をスケジュールして実行します。先ほどコピーした [S3 URI] および [Role ARN] (ロールの ARN) はそのまま保持してください。これらは、ファームウェア更新を行うために実行するタスク定義を作成するときに入力します。

タスク定義を使用してファームウェア更新をスケジュールし、実行する

タスク定義を使用して、ファームウェア更新に関する詳細を含めたり、更新を定義できます。AWS IoT Core for LoRaWAN では、ゲートウェイに関連付けられた次の 3 つのフィールドからの情報に基づいてファームウェアの更新が提供されます。

- ステーション

  Basics Station ソフトウェアのバージョンとビルド時間。この情報を特定するために、ゲートウェイによって実行されている Basics Station ソフトウェアを使用して生成することもできます (例:`2.0.5(rpi/std) 2021-03-09 03:45:09`)。
- PackageVersion (パッケージバージョン)

  ゲートウェイの `version.txt` ファイルによって指定されたファームウェアのバージョン。この情報はゲートウェイに存在しない可能性がありますが、ファームウェアのバージョンを定義する方法として推奨しています (例: `1.0.0`)。
- モデル

  ゲートウェイで使用されているプラットフォームまたはモデル (例: Linux)。

この手順の完了には 20 分かかります。

この手順を完了するには、以下を行います。

- ゲートウェイで現在実行されているバージョンを取得する
- ワイヤレスゲートウェイタスク定義を作成する
- ファームウェア更新タスクを実行し、進捗状況を追跡する

ゲートウェイで現在実行されているバージョンを取得する

ゲートウェイがファームウェア更新の対象であるかどうかを判断するために、CUPS サーバーは、CUPS 要求時にゲートウェイが `Station`、`PackageVersion`、`Model` のフィールドを提示し

たときに、これら 3 つのフィールドすべてについて、一致を確認します。タスク定義を使用する場合、これらのフィールドは CurrentVersion フィールドの一部として格納されています。

AWS IoT Core for LoRaWAN API または AWS CLI を使用して、ゲートウェイの CurrentVersion を取得できます。次のコマンドは、CLI を使用してこの情報を取得する方法を示しています。

1. ゲートウェイを既にプロビジョニングしている場合は、get-wireless-gateway コマンドを使用して、ゲートウェイに関する情報を取得できます。

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

   以下は、このコマンドの出力例です。

```
{
    "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. get-wireless-gateway コマンドによって報告されたワイヤレスゲートウェイ ID を使用して、get-wireless-gateway-firmware-information コマンドによって CurrentVersion を取得できます。

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

   以下は、このコマンドの出力例で、3 つのフィールドすべての情報が CurrentVersion によって表示されています。

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

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

ワイヤレスゲートウェイタスク定義を作成する

タスク定義を作成するときは、AutoCreateTasks パラメータを使用してタスクの自動作成を指定することをお勧めします。AutoCreateTasks は、前述の 3 つのパラメータすべてに一致するゲートウェイに適用されます。このパラメータが無効になっている場合は、パラメータを手動でゲートウェイに割り当てる必要があります。

AWS IoT Core for LoRaWAN API または AWS CLI を使用してワイヤレスゲートウェイのタスク定義を作成できます。以下のコマンドは、CLI を使用してタスク定義を作成する方法を示します。

1. input.json ファイルを作成します。このファイルには、CreateWirelessGatewayTaskDefinition API に渡す情報が含まれます。input.json ファイルで、前に取得した次の情報を指定します。

   - UpdateDataSource

     S3 バケットにアップロードしたファームウェア更新ファイルを含むオブジェクトへのリンクを指定します (例えば、s3://iotwirelessfwupdate/fwstation)。

   - UpdateDataRole

     S3 バケットを読み取る許可を提供する、作成した IAM ロールのロール ARN へのリンクを指定します (例えば、arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole)。

   - SigKeyCRC と UpdateSignature

     この情報はゲートウェイの製造元から提供されている可能性がありますが、「ファームウェア更新ファイルと署名を生成する」で説明されている手順に従った場合は、署名の生成時にこの情報を見つけることができます。

   - CurrentVersion

     先ほど get-wireless-gateway-firmware-information コマンドを実行して取得した CurrentVersion 出力を指定します。

   ```
   cat input.json
   ```

以下は、input.json ファイルの内容を示しています。

```
{
    "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. input.json ファイルを create-wireless-gateway-task-definition コマンドに渡して、タスク定義を作成します。

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

以下は、このコマンドの出力を示しています。

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

ファームウェア更新タスクを実行し、進捗状況を追跡する

ゲートウェイはファームウェア更新を受信する準備ができており、電源が入ると CUPS サーバーに接続します。CUPS サーバーは、ゲートウェイのバージョンで一致するものを検出すると、ファームウェア更新をスケジュールします。

タスクとは、処理中のタスク定義です。AutoCreateTasks を True に設定して自動タスク作成を指定したので、一致するゲートウェイが見つかるとすぐにファームウェア更新タスクが開始されます。

GetWirelessGatewayTask API を使用してタスクの進行状況を追跡できます。get-wireless-gateway-task コマンドを初めて実行すると、タスクのステータスが IN_PROGRESS として表示されます。

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

以下は、このコマンドの出力を示しています。

```
{
    "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"
}
```

次回にこのコマンドを実行した場合、ファームウェアの更新が有効になると、更新されたフィールド Package、Version、Model が表示され、タスクのステータスが COMPLETED に変わります。

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

以下は、このコマンドの出力を示しています。

```
{
    "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"
```

```
}
```

この例では、Raspberry Pi ベースの RAKWireless ゲートウェイを使用したファームウェア更新を紹介しました。更新された Package、Version、および Model フィールドを保存するために、ファームウェア更新スクリプトによって、実行中の BasicStation が停止します。そのため、BasicStation を再起動する必要があります。

```
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
```

ファームウェアのアップデートが失敗すると、CUPS サーバーからの FIRST_RETRY のステータスが表示され、ゲートウェイは同じ要求を送信します。SECOND_RETRY の後に CUPS サーバーがゲートウェイに接続できない場合、FAILED のステータスが表示されます。

前のタスクが COMPLETED または FAILED になったら、新しいタスクを開始する前に、delete-wireless-gateway-task コマンドを実行して古いタスクを削除します。

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

LoRaWAN ダウンリンクデータトラフィックを受信するゲートウェイの選択

デバイスに AWS IoT Core for LoRaWAN からダウンリンクメッセージを送信する場合、ダウンリンクデータトラフィックに使用するゲートウェイを選択できます。個々のゲートウェイを指定することも、ダウンリンクトラフィックを受信するゲートウェイのリストから選択することもできます。

ゲートウェイリストの指定方法

SendDataToWirelessDevice API オペレーションを使用して、デバイスからダウンリンクメッセージを送信する際に使用する個々のゲートウェイまたはゲートウェイのリストを指定することができます。API オペレーションを呼び出す際には、ゲートウェイの ParticipatingGateways オブジェクトを使用して、以下のパラメータを指定します。

> **Note**
>
> 使用するゲートウェイのリストは、AWS IoT コンソールでは使用できません。このゲートウェイのリストは、SendDataToWirelessDevice API オペレーションまたは CLI を使用する場合にのみ使用するように指定することができます。

- DownlinkMode: ダウンリンクメッセージをシーケンシャルモードで送信するか、並行モードで送信するかを示します。クラス A デバイスの場合は、前回のアップリンクメッセージ送信で選択したゲートウェイのみを使用するように UsingUplinkGateway を指定します。
- GatewayList: ダウンリンクデータトラフィックの送信に使用するゲートウェイのリスト。ダウンリンクペイロードは、指定された頻度で指定されたゲートウェイに送信されます。これは、リストを GatewayId と DownlinkFrequency のペアで構成される GatewayListItem オブジェクトのリストを使用して示されます。
- TransmissionInterval: AWS IoT Core for LoRaWAN が次のゲートウェイにペイロードを送信する前に待機する時間。

> **Note**
>
> このゲートウェイリストを指定して、ダウンリンクメッセージをクラス B またはクラス C のワイヤレスデバイスに送信する場合にのみ使用できます。クラス A デバイスを使用する場合、ダウンリンクメッセージがデバイスに送信されるときに、アップリンクメッセージの送信時に選択したゲートウェイが使用されます。

次の例は、ゲートウェイのパラメータを指定する方法を示しています。input.json ファイルには追加の詳細が含まれます。SendDataToWirelessDevice API オペレーションを使用したダウンリンクメッセージの送信の詳細については、「API を使用してダウンリンクキューオペレーションを実行する」を参照してください。

> **Note**
>
> 参加しているゲートウェイのリストを指定するパラメータは、AWS IoT コンソールを使用して AWS IoT Core for LoRaWAN からダウンリンクメッセージを送信するときには利用できません。

```
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
```

以下は、input.json ファイルの内容を示しています。

input.json の内容

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

このコマンドを実行したときの出力により、ダウンリンクメッセージの MessageId が生成されます。場合によっては、MessageId を受信した場合でも、パケットはドロップされる可能性があります。この問題を解決する方法の詳細については、「ダウンリンクメッセージキューエラーのトラブルシューティング」を参照してください。

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

参加しているゲートウェイのリストに関する情報を取得する

ダウンリンクキュー内のメッセージを一覧表示することで、ダウンリンクメッセージの受信に参加しているゲートウェイのリストに関する情報を取得できます。メッセージを一覧表示するには、`ListQueuedMessages` API を使用します。

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

このコマンドを実行すると、キュー内のメッセージとそのパラメータに関する情報が返されます。

AWS IoT Core for LoRaWAN によるデバイスの管理

次に、デバイスで AWS IoT Core for LoRaWAN を使用する際の重要な考慮事項をいくつか示します。AWS IoT Core for LoRaWAN へのデバイスの追加方法については、「デバイスを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。

デバイスに関する考慮事項

AWS IoT Core for LoRaWAN との通信に使用するデバイスを選択する際は、次の点を考慮してください。

- 使用可能なセンサー
- バッテリー容量
- エネルギー消費量
- Cost
- アンテナの種類と送信範囲

AWS IoT Core for LoRaWAN での使用を認定されたゲートウェイでのデバイスの使用

使用するデバイスは、AWS IoT Core for LoRaWAN での使用を認定されたワイヤレスゲートウェイとペアにできます。これらのゲートウェイとデベロッパーキットは、AWS Partner Device Catalog にあります。また、これらのデバイスの、ゲートウェイへの近接性も考慮することをお勧めします。詳細については、「AWS Partner Device Catalog の認定されたゲートウェイの使用」を参照してください。

LoRaWAN バージョン

AWS IoT Core for LoRaWAN は、LoRa Alliance によって標準化された 1.0.x または 1.1 LoRaWAN 仕様に準拠するすべてのデバイスをサポートしています。

アクティベーションモード

LoRaWAN デバイスがアップリンクデータを送信する前に、アクティベーションまたは参加手順と呼ばれるプロセスを完了する必要があります。デバイスをアクティベートするには、OTAA (無線通信経由アクティベーション) または ABP (パーソナライゼーションによるアクティベーション) のいずれかを使用できます。アクティベーションごとに新しいセッションキーが生成されて安全性が高まるため、デバイスのアクティベーションには OTAA を使用することをお勧めします。

ワイヤレスデバイスの仕様は、LoRaWAN のバージョンと、アクティベーションごとに生成されるルートキーとセッションキーを決定するアクティベーションモードに基づいています。詳細については、「コンソールを使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する」を参照してください。

デバイスクラス

LoRaWAN デバイスは、いつでもアップリンクメッセージを送信できます。ダウンリンクメッセージをリッスンすると、バッテリー容量が消費され、バッテリー持続時間が短縮されます。LoRaWAN プロトコルは、LoRaWAN デバイスの 3 つのクラスを指定します。

- クラス A デバイスは、ほとんどの時間をスリープ状態にし、短時間だけダウンリンクメッセージをリッスンします。これらのデバイスは、最大 10 年のバッテリー寿命を持つバッテリー駆動のセンサーです。
- クラス B デバイスは、スケジュールされたダウンリンクスロットでメッセージを受信できます。これらのデバイスは、主にバッテリー駆動のアクチュエータです。
- クラス C デバイスはスリープせず、継続的に着信メッセージをリッスンすることがないため、メッセージの受信にそれほど遅延はありません。これらのデバイスは、主に主電源駆動のアクチュエータです。

これらのワイヤレスデバイスの考慮事項の詳細については、「LoRaWAN の詳細」で説明したリソースを参照してください。

トピック

AWS IoT Core for LoRaWAN でのアダプティブデータレート (ADR) の実行

エンドデバイスからのメッセージがゲートウェイで受信されるようにしつつ、デバイスの送信電力消費量を最適化するために、AWS IoT Core for LoRaWAN では、アダプティブデータレートを使用します。アダプティブデータレートは、ゲートウェイで受信されるパケットのエラーレートを下げながら、データレート、送信電力、再送信回数を最適化するようにエンドデバイスに指示します。例えば、エンドデバイスがゲートウェイの近くにある場合、アダプティブデータレートによって送信電力が低下し、データレートが向上します。

トピック

アダプティブデータレート (ADR) の仕組み

ADR を有効にするには、デバイスがフレームヘッダーに ADR ビットを設定する必要があります。ADR ビットが設定されると、AWS IoT Core for LoRaWAN は `LinkADRReq` MAC コマンドを送信し、デバイスは ADR コマンドの ACK ステータスを含む `LinkADRAns` コマンドで応答します。デバイスが ADR コマンドを ACK すると、AWS IoT Core for LoRaWAN からの ADR 指示に従って、最適なデータレートを得るように送信パラメータ値を調整します。

AWS IoT Core for LoRaWAN ADR アルゴリズムは、アップリンクメタデータ履歴の SINR 情報を使用して、デバイスに使用する最適な送信電力とデータレートを決定します。このアルゴリズムは、ADR ビットがフレームヘッダーに設定された後に始まる最新の 20 個のアップリンクメッセージを使用します。再送信の数を決定するために、パケットエラー率 (PER) を使用します。PER は、失われたパケットの合計数に対する割合です。このアルゴリズムを使用すると、データレートの範囲、つまりデータレートの最小制限と最大制限のみを制御できます。

データレート制限 (CLI) を設定する

デフォルトでは、LoRaWAN デバイスのフレームヘッダーに ADR ビットを設定すると、AWS IoT Core for LoRaWAN は ADR を実行します。AWS IoT Wireless API オペレーション `CreateServiceProfile`、または AWS CLI コマンド `create-service-profile` を使用し

て、LoRaWAN デバイスのサービスプロファイルを作成するときに、データレートの最小制限と最大制限を制御できます。

> **Note**
>
> AWS Management Console からサービスプロファイルを作成するときに、最大データレート制限と最小データレート制限を指定することはできません。AWS IoT Wireless API または AWS CLI を使用してのみ指定にすることができます。

データレートの最小制限と最大制限を指定するには、`CreateServiceProfile` API オペレーションで `DrMin` および `DrMax` パラメータを使用します。デフォルトの最小および最大データレート制限は 0 と 15 です。例えば、次の CLI コマンドは、最小データレート制限を 3、最大制限を 12 に設定します。

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

このコマンドを実行すると、サービスプロファイルの ID と Amazon リソースネーム (ARN) が生成されます。

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

AWS IoT Wireless API オペレーション GetServiceProfile、または AWS CLI コマンド get-service-profile を使用して、指定されたパラメータの値を取得できます。

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

このコマンドを実行すると、サービスプロファイルパラメータの値が生成されます。

```
{
    "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
    }
}
```

複数のプロファイルを作成した場合は、API オペレーション、ListServiceProfiles、または AWS CLI コマンド list-service-profiles を使用して、AWS アカウント のサービスプロファイルを一覧表示し、次に GetServiceProfile API または get-service-profile CLI コマンドを使用して、データレート制限をカスタマイズしたサービスプロファイルを取得できます。

LoRaWAN デバイスと AWS IoT 間の通信の管理

LoRaWAN デバイスを AWS IoT Core for LoRaWAN に接続した後、デバイスはクラウドへのメッセージの送信を開始できます。アップリンクメッセージは、デバイスから送信され、AWS IoT Core for LoRaWAN によって受信されるメッセージです。LoRaWAN デバイスはいつでもアップリンクメッセージを送信できます。その後、AWS のサービス およびクラウドホスト型アプリケーションへ転送されます。AWS IoT Core for LoRaWAN およびその他の AWS のサービス やアプリケーションからデバイスへ送信されるメッセージは、ダウンリンクメッセージと呼ばれます。

以下は、デバイスとクラウド間で送信されるアップリンクメッセージとダウンリンクメッセージを表示および管理する方法を示しています。ダウンリンクメッセージのキューを保持し、キューに追加された順序でこれらのメッセージをデバイスに送信できます。

トピック

- LoRaWAN デバイスから送信されたアップリンクメッセージの形式の表示
- LoRaWAN デバイスに送信するダウンリンクメッセージをキューに入れる

LoRaWAN デバイスから送信されたアップリンクメッセージの形式の表示

LoRaWAN デバイスを AWS IoT Core for LoRaWAN に接続したら、ワイヤレスデバイスから受信するアップリンクメッセージの形式を確認できます。

アップリンクメッセージを確認する前に

データを送受信できるように、ワイヤレスデバイスをオンボーディングし、デバイスを AWS IoT に接続済みである必要があります。AWS IoT Core for LoRaWAN へのデバイスのオンボーディングについては、「デバイスを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。

アップリンクメッセージに含まれているもの

LoRaWAN デバイスは、LoRaWAN ゲートウェイを使用して AWS IoT Core for LoRaWAN に接続できます。デバイスから受信したアップリンクメッセージには、次の情報が含まれます。

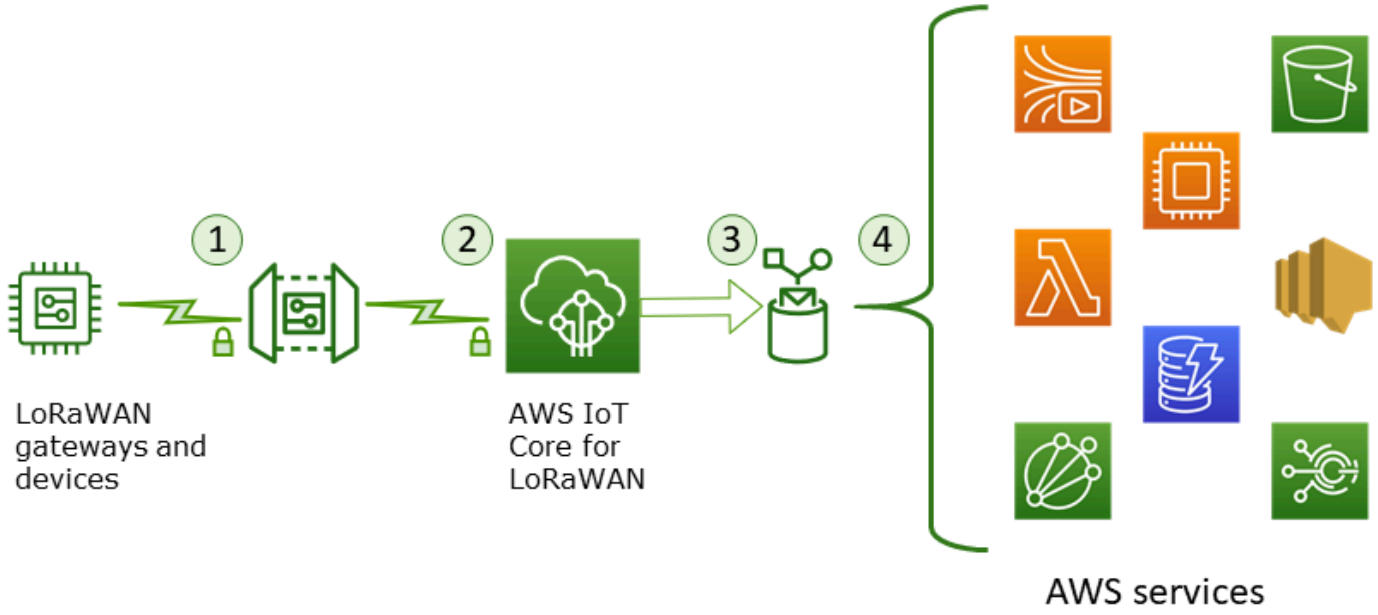
- ワイヤレスデバイスから送信される暗号化されたペイロードメッセージに対応するペイロードデータ。
- 以下を含むワイヤレスメタデータ
  - DevEUI、データレート、デバイスが動作している周波数チャネルなどのデバイス情報。
  - デバイスに接続されているゲートウェイのオプションの追加パラメータおよびゲートウェイ情報。ゲートウェイパラメータには、ゲートウェイの EUI、SNR、および RSSI が含まれます。

  ワイヤレスメタデータを使用すると、ワイヤレスデバイスや、デバイスと AWS IoT の間で送信されるデータに関する有用な情報が得られます。例えば、`AckedMessageId` パラメータを使用して、最後に確認されたダウンリンクメッセージがデバイスによって受信されたかどうかを確認できます。必要に応じて、ゲートウェイ情報を含めることを選択した場合、デバイスに近い強力なゲートウェイチャネルに切り替えるかどうかを識別できます。

アップリンクメッセージを確認する方法

デバイスをオンボーディングした後、AWS IoT コンソールの [Test] (テスト) ページで MQTT テストクライアントを使用して、送信先を作成するときに指定したトピックにサブスクライブできます。デバイスが接続され、ペイロードデータの送信が開始されると、メッセージが表示されるようになります。

この図は、AWS IoT Core for LoRaWAN に接続された LoRaWAN システムの主要な要素を識別するとともに、プライマリデータプレーンおよびシステム内のデータフローを示しています。



ワイヤレスデバイスがアップリンクデータの送信を開始すると、AWS IoT Core for LoRaWAN によりワイヤレスメタデータの情報がペイロードでラッピングされ、AWS アプリケーションに送信されます。

アップリンクメッセージの例

次の例は、デバイスから受信したアップリンクメッセージの形式を示しています。

```
{
    "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"

        }
    }
}
```

アップリンクメタデータからゲートウェイメタデータを除外する

アップリンクメタデータからゲートウェイメタデータ情報を除外する場合は、サービスプロファイルの作成時に AddGwMetadata パラメータを無効にします。このパラメータの無効化については、「サービスプロファイルを追加する」を参照してください。

この場合、次の例に示すように、アップリンクメタデータに Gateways セクションは表示されません。

```
{
    "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"
    }
  }
```

```
}
```

LoRaWAN デバイスに送信するダウンリンクメッセージをキューに入れる

クラウドホスト型アプリケーションおよび AWS のサービス は、ワイヤレスデバイスにダウンリンクメッセージを送信できます。ダウンリンクメッセージは、AWS IoT Core for LoRaWAN からワイヤレスデバイスに送信されるメッセージです。AWS IoT Core for LoRaWAN にオンボーディングした各デバイスに、ダウンリンクメッセージをスケジュールして送信できます。

ダウンリンクメッセージを送信する対象のデバイスが複数ある場合は、マルチキャストグループを使用できます。マルチキャストグループ内のデバイスは、同じマルチキャストアドレスを共有し、その後、受信側デバイスのグループ全体に配信されます。詳細については、「マルチキャストグループを作成してダウンリンクのペイロードを複数のデバイスに送信する」を参照してください。

ダウンリンクメッセージキューのしくみ

LoRaWAN デバイスのデバイスクラスによって、キュー内のメッセージがデバイスにどのように送信されるかが決まります。クラス A デバイスは、デバイスがダウンリンクメッセージを受信できることを示すために、アップリンクメッセージを AWS IoT Core for LoRaWAN に送信します。。クラス B デバイスは、通常のダウンリンクスロットでメッセージを受信できます。クラス C デバイスは、いつでもダウンリンクメッセージを受信できます。デバイスクラスの詳細については、「デバイスクラス」を参照してください。

次に、メッセージがキューに入れられ、クラス A デバイスに送信される方法を示します。

1. AWS IoT Core for LoRaWAN は、フレームポート、ペイロードデータ、および AWS IoT コンソールまたは AWS IoT Wireless API を使用して指定した確認モードパラメータを使用して、キューに追加したダウンリンクメッセージをバッファします。
2. LoRaWAN デバイスは、オンラインであることを示すアップリンクメッセージを送信し、ダウンリンクメッセージの受信を開始できます。
3. 複数のダウンリンクメッセージをキューに追加した場合は、AWS IoT Core for LoRaWAN は、[acknowledge (ACK)] (確認) フラグが設定された状態で、キュー内の最初のダウンリンクメッセージをデバイスに送信します。
4. デバイスは、アップリンクメッセージを直ちに AWS IoT Core for LoRaWAN に送信する、または次のアップリンクメッセージまでスリープし、メッセージに ACK フラグを含めます。
5. AWS IoT Core for LoRaWAN は ACK フラグ付きのアップリンクメッセージを受信し、ダウンリンクメッセージをキューからクリアし、デバイスがダウンリンクメッセージを正常に受信したこと

を示します。3 回チェックした後に ACK フラグがアップリンクメッセージに表示されない場合、メッセージは破棄されます。

コンソールを使用してダウンリンクキューオペレーションを実行する

必要に応じて、AWS Management Console を使用してダウンリンクメッセージをキューに入れ、個々のメッセージ、またはキュー全体をクリアすることができます。クラス A デバイスの場合、アップリンクがデバイスから受信されてオンラインであることを示すと、キューに入れられたメッセージがデバイスに送信されます。メッセージが送信されると、キューから自動的にクリアされます。

ダウンリンクメッセージをキューに入れる

ダウンリンクメッセージキューを作成するには

1. [Devices hub of the AWS IoT console] (IoT コンソールのデバイスハブ) へ移動し、ダウンリンクメッセージをキューに入れるデバイスを選択します。
2. デバイスの詳細ページの [Downlink messages] (ダウンリンクメッセージ) セクションで、[Queue downlink messages] (ダウンリンクメッセージをキューに入れる) を選択します。
3. ダウンリンクメッセージを設定するには、次のパラメータを指定します。
   - FPort: デバイスが AWS IoT Core for LoRaWAN と通信するフレームポートを選択します。
   - Payload: デバイスに送信するペイロードメッセージを指定します。最大ペイロードサイズは 242 バイトです。アダプティブデータレート (ADR) が有効になっている場合、AWS IoT Core for LoRaWAN では、ペイロードサイズに最適なデータレートを選択するために使用します。必要に応じて、データレートをさらに最適化できます。
   - 確認モード : デバイスがダウンリンクメッセージを受信したかどうかを確認します。メッセージにこのモードが必要な場合は、データストリームに ACK フラグを含むアップリンクメッセージが表示され、メッセージはキューからクリアされます。
4. ダウンリンクメッセージをキューに追加するには、[Submit] (送信) を選択します。

ダウンリンクメッセージがキューに追加されました。メッセージが表示されない場合、またはエラーを受信した場合は、ダウンリンクメッセージキューエラーのトラブルシューティング の説明に従ってエラーをトラブルシューティングできます。

> **Note**
>
> ダウンリンクメッセージがキューに追加されると、FPort、Payload、確認モード パラメータを編集できなくなります。これらのパラメータに異なる値を持つダウンリンクメッセージを

送信するには、このメッセージを削除し、更新されたパラメータ値を含む新しいダウンリンクメッセージをキューに入れます。

キューには、追加したダウンリンクメッセージが一覧表示されます。デバイスと AWS IoT Core for LoRaWAN の間で交換されるアップリンクメッセージおよびダウンリンクメッセージのペイロードを表示するには、ネットワークアナライザを使用します。詳細については、「ネットワークアナライザを使用したワイヤレスリソースフリートのリアルタイムでのモニタリング」を参照してください。

ダウンリンクメッセージキューの一覧表示

作成したダウンリンクメッセージがキューに追加されます。後続の各ダウンリンクメッセージは、このメッセージの後にキューに追加されます。ダウンリンクメッセージのリストは、デバイスの詳細ページの [Downlink messages] (ダウンリンクメッセージ) セクションに表示されます。アップリンクを受信すると、メッセージがデバイスに送信されます。デバイスがダウンリンクメッセージを受信すると、そのメッセージはキューから削除されます。次のメッセージがキュー内で上に移動し、デバイスに送信されます。

個々のダウンリンクメッセージを削除するまたはキュー全体をクリアする

各ダウンリンクメッセージは、デバイスに送信された後、キューから自動的にクリアされます。個々のメッセージを削除したり、ダウンリンクキュー全体をクリアすることもできます。これらの操作は元に戻すことができません。

- 送信したくないメッセージがキュー内に見つかった場合は、メッセージを選択し、[Delete] (削除) を選択します。
- キューからデバイスにメッセージを送信したくない場合は、[Clear downlink queue] (ダウンリンクキューをクリア) を選択して、キュー全体をクリアできます。。

API を使用してダウンリンクキューオペレーションを実行する

必要に応じて、AWS IoT Wireless APIを使用してダウンリンクメッセージをキューに入れたり、個々のメッセージ、またはキュー全体をクリアすることができます。

ダウンリンクメッセージをキューに入れる

ダウンリンクメッセージキューを作成するには、`SendDataToWirelessDevice` API オペレーションまたは `send-data-to-wireless-device` CLI コマンドを使用します。

```
aws iotwireless send-data-to-wireless-device \
```

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

このコマンドを実行したときの出力により、ダウンリンクメッセージの MessageId が生成されます。場合によっては、MessageId を受信した場合でも、パケットはドロップされる可能性があります。この問題を解決する方法の詳細については、「ダウンリンクメッセージキューエラーのトラブルシューティング」を参照してください。

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

キュー内のダウンリンクメッセージを一覧表示する

キュー内のすべてのダウンリンクメッセージを一覧表示するには、ListQueuedMessages API オペレーションまたは list-queued-messages CLI コマンドを使用します。

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

デフォルトでは、このコマンドを実行すると、最大 10 個のダウンリンクメッセージが表示されます。

個々のダウンリンクメッセージを削除するまたはキュー全体をクリアする

キューから個々のメッセージを削除したり、キュー全体をクリアしたりするには、DeleteQueuedMessages API オペレーションまたは delete-queued-messages CLI コマンドを使用します。

- 個々のメッセージを削除するには、wirelessDeviceId により指定されたワイヤレスデバイスから削除するメッセージの messageID を指定します。
- ダウンリンクキュー全体をクリアするには、wirelessDeviceId により指定されたワイヤレスデバイスの messageID を * に指定します。

ダウンリンクメッセージキューエラーのトラブルシューティング

ここでは、期待通りの結果が得られない場合に確認すべきことを説明します。

- ダウンリンクメッセージが AWS IoT コンソールに表示されない

  コンソールを使用してダウンリンクキューオペレーションを実行する の説明に従って、ダウンリンクメッセージを追加した後にキューにダウンリンクメッセージが表示されない場合、デバイスが[activation] (アクティベーション) または[join procedure] (結合手順) と呼ばれるプロセスを完了していないことが原因である可能性があります。この手順は、デバイスが AWS IoT Core for LoRaWAN にオンボードしているときに完了します。詳細については、「コンソールを使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する」を参照してください。

  デバイスを AWS IoT Core for LoRaWAN にオンボーディングした後、ネットワークアナライザーまたは Amazon CloudWatch を使用して、デバイスをモニタリングして、結合と再結合が成功したかどうかを確認できます。詳細については、「モニタリングツール」を参照してください。

- API の使用時にダウンリンクメッセージパケットが見つからない

  `SendDataToWirelessDevice` API オペレーションを使用する場合、API は一意の `MessageId` を返します。ただし、LoRaWAN デバイスがダウンリンクメッセージを受信したかどうかは確認できません。ダウンリンクパケットは、デバイスが結合手順を完了していない場合などにドロップされる可能性があります。このエラーを解決する方法の詳細については、前のセクションを参照してください。

- ダウンリンクメッセージの送信時に ARN エラーが発生しない

  キューからデバイスにダウンリンクメッセージを送信すると、Amazon リソースネーム (ARN) が見つからないというエラーが表示されることがあります。このエラーは、ダウンリンクメッセージを受信しているデバイスの宛先が正しく指定されていないために発生する可能性があります。このエラーを解決するには、デバイスの送信先の詳細を確認します。

パブリック LoRaWAN デバイスネットワーク (Everynet) からの LoRaWAN トラフィックの管理

公開されている LoRaWAN ネットワークを使用すれば、LoRaWAN デバイスを数分でクラウドに接続できます。AWS IoT Core for LoRaWAN は、米国と英国の Everynet のネットワークカバレッジをサポートするようになりました。パブリックネットワークを使用する場合、デバイスごとにパブリックネットワーク接続料金が毎月請求されます。この料金は、パブリックネットワーク接続が提供されるすべての AWS リージョン に適用されます。この機能の価格設定の詳細については、「AWS IoT Core 料金表」を参照してください。

> **Important**
>
> パブリックネットワークは Everynet が直接サービスとして運営、提供しています。この機能を使用する前に、該当する「AWSサービス条件」を参照してください。さらに、AWS IoT Core for LoRaWAN を通してパブリックネットワークを利用する場合、`DevEUI` や `JoinEUI` などの特定の LoRaWAN デバイス情報は、AWS IoT Core for LoRaWAN が利用可能なリージョン間でレプリケーションされます。

AWS IoT Core for LoRaWAN は、「LoRaWAN バックエンドインターフェイス 1.0 仕様」で説明されているように、ローミングの LoRa Alliance 仕様に従ってパブリック LoRaWAN ネットワークをサポートします。パブリックネットワーク機能を使用して、ホームネットワークの外部にあるエンドデバイスを接続できます。この機能をサポートするために、AWS IoT Core for LoRaWAN は Everynet と提携して無線通信範囲を拡大しています。

パブリック LoRaWAN ネットワークを使用する利点

LoRaWAN デバイスは、パブリックネットワークを使用してクラウドに接続できるため、デプロイまでの時間が短縮され、プライベート LoRaWAN ネットワークの維持に必要な時間とコストを削減できます。

パブリック LoRaWAN ネットワークを使用すると、カバレッジの拡張、無線ネットワークなしでのコアの実行、カバレッジの高密度化などの利点が得られます。この機能は次の用途に使用できます。

- パブリック LoRaWAN ネットワークサポートアーキテクチャ セクションの図のデバイス A など、ホームネットワークから移動するデバイスにカバレッジを提供します。
- パブリック LoRaWAN ネットワークサポートアーキテクチャ セクションの図のデバイス B など、接続する LoRa ゲートウェイがないデバイスにもカバレッジを拡張します。その後、デバイスはパートナーが提供するゲートウェイを使用してホームネットワークに接続できます。

LoRaWAN デバイスは、ローミング機能を使用してパブリックネットワークを使用してクラウドに接続できるため、デプロイまでの時間が短縮され、プライベート LoRaWAN ネットワークの維持に必要な時間とコストを削減できます。

以下のセクションでは、パブリックネットワークサポートのアーキテクチャ、パブリック LoRaWAN ネットワークサポートの仕組み、およびこの機能の使用方法について説明します。

トピック

- LoRaWAN パブリックネットワークサポートの仕組み
- パブリックネットワークサポートの使用方法

LoRaWAN パブリックネットワークサポートの仕組み

AWS IoT Core for LoRaWAN は、LoRa Alliance の仕様に従って、パッシブローミング機能をサポートしています。パッシブローミングでは、ローミングプロセスはエンドデバイスに対して完全に透過的です。ホームネットワークの外部をローミングするエンドデバイスは、そのネットワーク内のゲートウェイに接続し、アプリケーションサーバーを使用してアップリンクデータとダウンリンクデータを交換できます。デバイスは、ローミングプロセス全体を通してホームネットワークに接続されたままになります。

> **Note**
>
> AWS IoT Core for LoRaWAN は、パッシブローミングのステートレス機能のみをサポートします。ハンドオーバーローミングはサポートされていません。ハンドオーバーローミングでは、デバイスがホームネットワークの外部に移動すると、別の通信事業者に切り替わります。

トピック

- パブリック LoRaWAN ネットワークの概念
- パブリック LoRaWAN ネットワークサポートアーキテクチャ

パブリック LoRaWAN ネットワークの概念

AWS IoT Core for LoRaWAN でサポートされているパブリックネットワーク機能では、次の概念が使用されます。

LoRaWAN network server (LNS)

LNS はスタンドアロンのプライベートサーバーで、オンプレミスで実行することも、クラウドベースのサービスにすることもできます。AWS IoT Core for LoRaWAN は、クラウド上でサービスを提供する LNS です。

ホームネットワークサーバー (hNS)

ホームネットワークは、デバイスが属するネットワークです。ホームネットワークサーバー (hNS) は、AWS IoT Core for LoRaWAN がデバイスのプロビジョニングデータ (DevEUI、AppEUI、セッションキーなど) を保存する LNS です。

訪問先ネットワークサーバー (vNS)

訪問先ネットワークとは、デバイスがホームネットワークを離れるときに通信可能になるネットワークです。訪問先ネットワークサーバー (vNS) は、エンドデバイスにサービスを提供できるように hNS とビジネス上および技術上の契約を結んでいる LNS です。AWS パートナーの Everynet は訪問先のネットワークとして機能し、カバレッジを提供します。

サービス提供ネットワークサーバー (sNS)

サービス提供ネットワークサーバー (sNS) は、デバイスの MAC コマンドを処理する LNS です。1 つの LoRa セッションには 1 つの sNS しか存在できません。

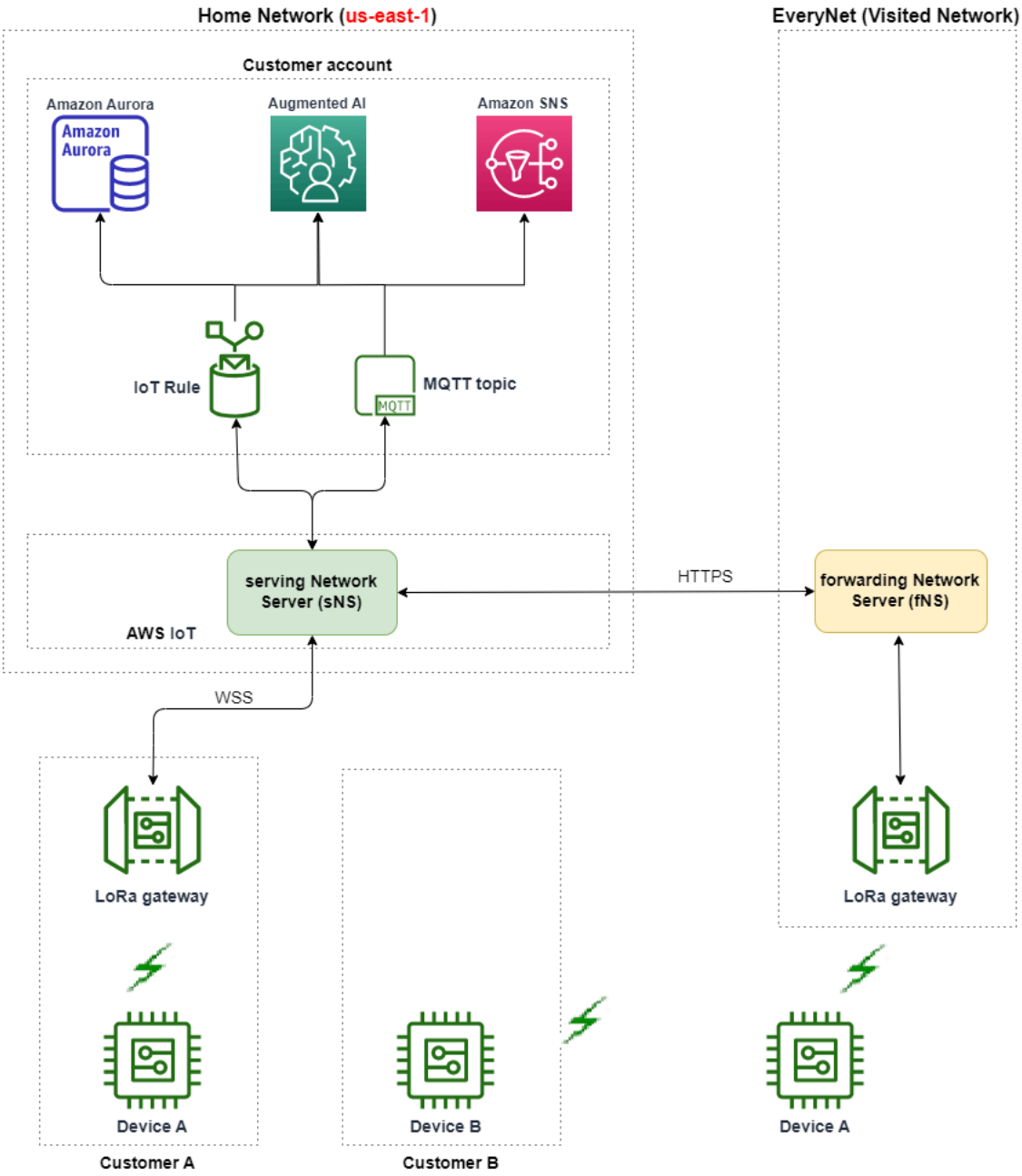
転送ネットワークサーバー (fNS)

転送ネットワークサーバー (fNS) は、無線ゲートウェイを管理する LNS です。1 つの LoRa セッションには、0 個以上の fNS が関与する可能性があります。このネットワークサーバーは、デバイスから受信したデータパケットのホームネットワークへの転送を管理します。

パブリック LoRaWAN ネットワークサポートアーキテクチャ

次のアーキテクチャ図は、AWS IoT Core for LoRaWAN が Everynet と提携してパブリックネットワーク接続を提供する方法を示しています。この場合、デバイス A は、LoRa ゲートウェイを介して AWS IoT Core for LoRaWAN によって提供される hNS (ホームネットワークサーバー) に接続されます。デバイス A がホームネットワークから移動すると、訪問先のネットワークに入り、Everynet が提供する訪問先ネットワークサーバー (vNS) によってカバーされます。また、vNS は、接続する LoRa ゲートウェイを持たないデバイス B にもカバレッジを拡張します。

次のセクションで説明するように、パブリックネットワークのカバレッジ情報を AWS IoT コンソールに表示できます。



AWS IoT Core for LoRaWAN は、LoRa Alliance LoRaWAN Roaming Hub Technical Recommendation に従ってローミングハブ機能を使用しています。ローミングハブは、Everynet がエンドデバイスから受信したトラフィックをルーティングするためのエンドポイントを提供します。この場合、Everynet はデバイスから受信したトラフィックを転送する転送ネットワークサーバー (fNS) として機能します。LoRa Alliance 仕様で定義されている HTTP RESTful API を使用しています。

> **Note**
>
> デバイスがホームネットワークから移動して、ホームネットワークと Everynet の両方が通信できる場所に入ると、先着順のポリシーを使用して、LoRa ゲートウェイに接続するか、Everynet のゲートウェイに接続するかを決定します。

パブリックネットワークにアクセスする場合、HN とサービングネットワークサーバー (SN) は分離されます。次に、sNS と hNS の間でアップリンクパケットとダウンリンクパケットが交換されます。

パブリックネットワークサポートの使用方法

Everynet のパブリックネットワークサポートを指定にするには、サービスプロファイルの作成時に特定のローミングパラメータを指定します。このベータリリースでは、AWS IoT Wireless API または AWS CLI を使用するときにこれらのパラメータを使用できます。次のセクションでは、有効にする必要があるパラメータと、AWS CLI を使用してパブリックネットワークを有効する方法を示します。

> **Note**
>
> パブリックネットワークサポートは、新しいサービスプロファイルを作成する場合のみ有効にできます。これらのパラメータを使用して既存のプロファイルを更新してパブリックネットワークを有効にすることはできません。

トピック

- ローミングパラメータ
- デバイスのパブリックネットワークサポートを有効にします。

ローミングパラメータ

デバイスのサービスプロファイルを作成するときに、次のパラメータを指定します。AWS IoT コンソールのプロファイルハブからサービスプロファイルを追加するとき、または AWS IoT Wireless API オペレーションの `CreateServiceProfile`、または AWS CLI コマンドの `create-service-profile` を使用するときに、これらのパラメータを指定します。

> **Note**
>
> AWS IoT Core for LoRaWAN は、ハンドオーバーローミングをサポートしていません。サービスプロファイルを作成する場合、ハンドオーバーローミングを使用するかどうかを指定する `HrAllowed` パラメータを有効にすることはできません。

- ローミングアクティベーションが許可されました (`RaAllowed`): このパラメータは、ローミングアクティベーションを有効にするかどうかを指定します。ローミングアクティベーションにより、エンドデバイスを vNS の通信範囲内でアクティブ化できます。ローミング機能を使用する場合は、`RaAllowed` を `true` に設定する必要があります。
- パッシブローミングが許可されました (`PrAllowed`): このパラメータは、パッシブローミングを有効にするかどうかを指定します。ローミング機能を使用する場合は、`PrAllowed` を `true` に設定する必要があります。

デバイスのパブリックネットワークサポートを有効にします。

デバイスでパブリック LoRaWAN ネットワークサポートを有効にするには、次の手順を実行します。

> **Note**
>
> パブリックネットワーク機能は OTAA デバイスでのみ有効にできます。この機能は、アクティベーション方法として ABP を使用するデバイスではサポートされません。

1. ローミングパラメータを含むサービスプロファイルを作成する

   ローミングパラメータを有効にしてサービスプロファイルを作成します。

   > **Note**
   >
   > このサービスプロファイルに関連付けるデバイスのデバイスプロファイルを作成するときは、`RxDelay1` パラメータに 2 秒以上の大きな値を指定することをお勧めします。

   - AWS IoT コンソールを使用する場合

AWS IoT コンソールのプロファイルハブに移動し、[サービスプロファイルの追加] を選択します。プロファイルを作成するときは、[パブリックネットワークを有効にする] を選択します。

- AWS IoT Wireless API を使用する場合

  サービスプロファイルの作成時にローミングを有効にするには、以下の例に示すように、CreateServiceProfile API オペレーションまたは `create-service-profile` CLI コマンドを使用します。

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

  このコマンドを実行すると、サービスプロファイルの ARN と ID が出力として返されます。

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

2. サービスプロファイルのローミングパラメータを確認する

   指定したローミングパラメータを確認するには、次の例に示すように、コンソールでサービスプロファイルを表示するか、`get-service-profile` CLI コマンドを使用します。

- AWS IoT コンソールを使用する場合

  AWS IoT コンソールのプロファイルハブに移動し、作成したプロファイルを選択します。詳細ページの [プロファイル設定] タブで、[RaAllowed] と [PrAllowed] が [`true`] に設定されていることがわかります。

- AWS IoT Wireless API を使用する場合

  有効にしたローミングパラメータを表示するには、以下の例に示すように、GetServiceProfile API オペレーションまたは `get-service-profile` CLI コマンドを使用します。

```
aws iotwireless get-service-profile \
    --region us-east-1 \
```

```
    --id 12345678-a1b2-3c45-67d8-e90fa1b2c34d
```

このコマンドを実行すると、ローミングパラメータの値や RaAllowed および PrAllowed などのサービスプロファイルの詳細が出力として返されます。

```
{
    "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. サービスプロファイルをデバイスにアタッチする

ローミングパラメータを使用して作成したサービスプロファイルをエンドデバイスにアタッチします。デバイスプロファイルを作成し、ワイヤレスデバイスの宛先を追加することもできます。この宛先を使用して、デバイスから送信されるアップリンクメッセージをルーティングします。デバイスプロファイルと宛先の作成の詳細については、「デバイスプロファイルを追加する」と「AWS IoT Core for LoRaWAN に送信先を追加する」を参照してください。

- 新しいデバイスのオンボーディング

  デバイスをまだオンボーディングしていない場合は、デバイスを AWS IoT Core for LoRaWAN に追加するときにこのサービスプロファイル使用するように指定します。次のコマンドは、create-wireless-device CLI コマンドを使用して、作成したサービスプロファ

イルの ID を使用してデバイスを追加する方法を示しています。コンソールを使用したサービスプロファイルを追加する方法については、「コンソールを使用してワイヤレスデバイスの仕様を AWS IoT Core for LoRaWAN に追加する」を参照してください。

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

以下は、*createdevice.json* ファイルの内容を示しています。

createdevice.json の内容

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

このコマンドを実行すると、ワイヤレスデバイスの ARN と ID が出力として生成されます。

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

- 既存のデバイスの更新

  デバイスを既にオンボーディングしている場合は、このサービスプロファイルを使用するように既存のワイヤレスデバイスを更新できます。次のコマンドは、update-wireless-device CLI コマンドを使用して、作成したサービスプロファイルの ID を使用してデバイスを更新する方法を示しています。

```
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"
```

このコマンドでは、出力が生成されません。GetWirelessDevice API または get-wireless-device CLI コマンドを使用して、更新された情報を取得できます。

4. Everynet を使用してデバイスをクラウドに接続する

ローミングが有効になっているため、デバイスは結合を実行して新しい DevAddr を取得する必要があります。OTAA を使用している場合、LoRaWAN デバイスが参加要求を送信し、ネットワークサーバーが要求を許可することができます。その後、Everynet が提供するネットワークカバレッジを使用して AWS クラウド クラウドに接続できます。デバイスのアクティベーション手順や参加方法については、デバイスのマニュアルを参照してください。

> **Note**
>
> - ローミング機能を有効にし、アクティベーション方法として OTAA を使用するデバイスのパブリックネットワークへの接続を有効にできます。ABP デバイスはサポートされていません。デバイスのアクティベーション手順や参加方法については、デバイスのマニュアルを参照してください。「アクティベーションモード」を参照してください。
> - デバイスのローミング機能を無効にするには、このサービスプロファイルからデバイスの関連付けを解除し、ローミングパラメータが false に設定されている別のサービスプロファイルにデバイスを関連付けます。このサービスプロファイルに切り替えた後、デバイスがパブリックネットワークで実行され続けないように、別の結合を実行する必要があります。

5. アップリンクとダウンリンクのメッセージを交換する

デバイスが AWS IoT Core for LoRaWAN に接続されたら、デバイスとクラウドの間でメッセージの交換を開始できます。

- アップリンクメッセージを表示する

  デバイスからアップリンクメッセージを送信すると、AWS IoT Core for LoRaWAN は以前に設定した宛先を使用してこれらのメッセージを AWS アカウント アカウントに配信します こ

れらのメッセージは、デバイスから Everynet のネットワーク経由でクラウドに送信されます。

AWS IoT ルール名を使用してメッセージを表示することも、MQTT クライアントを使用して宛先の作成時に指定された MQTT トピックをサブスクライブすることもできます。指定するルール名やその他の宛先の詳細については、「コンソールを使用して送信先を追加します」を参照してください。

アップリンクメッセージの表示とその形式の詳細については、「LoRaWAN デバイスから送信されたアップリンクメッセージの形式の表示」を参照してください。

- ダウンリンクメッセージを送信する

  コンソールから、または AWS IoT Wireless API コマンド `SendDataToWirelessDevice`、または AWS CLI コマンド `send-data-to-wireless-device` を使用して、ダウンリンクメッセージをキューに入れてデバイスに送信できます。ダウンリンクメッセージのキューイングと送信については、「LoRaWAN デバイスに送信するダウンリンクメッセージをキューに入れる」を参照してください。

  次のコードは、`send-data-to-wireless-device` CLIコマンドを使用してダウンリンクメッセージを送信する方法の例を示しています。データを受信するワイヤレスデバイスの ID、ペイロード、確認モードを使用するかどうか、およびワイヤレスメタデータを指定します。

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

  このコマンドを実行したときの出力により、ダウンリンクメッセージの `MessageId` が生成されます。

  > **Note**
  >
  > 場合によっては、`MessageId` を受信した場合でも、パケットはドロップされる可能性があります。このようなシナリオのトラブルシューティングと解決方法については、「ダウンリンクメッセージキューエラーのトラブルシューティング」を参照してください。

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

- カバレッジ情報を表示する

  パブリックネットワークを有効にすると、ネットワークカバレッジ情報を AWS IoT コンソールに表示できます。AWS IoT コンソールの [カバレッジ] ハブに移動して場所を検索すると、デバイスのカバレッジ情報が地図上に表示されます。

  > **Note**
  >
  > この機能は、Amazon Location Service を使用して、デバイスのカバレッジ情報を Amazon ロケーションマップに表示します。Amazon ロケーションマップを使用する前に、Amazon Location Service 利用規約を確認してください。AWS は、選択したサードパーティーデータプロバイダーに API クエリを送信する場合があることに注意してください。このプロバイダーは現在使用している AWS リージョン の外部にある可能性があります。詳細については、「AWS サービス条件」を参照してください。

LoRaWAN デバイスおよびマルチキャストグループのファームウェア更新無線通信経由 (FUOTA) を実行する

ファームウェア更新を無線通信経由で実行して、単一の LoRaWAN デバイスまたはデバイスのグループのデバイスファームウェアを更新できます。デバイスファームウェアを更新するか、またはダウンリンクのペイロードを複数のデバイスに送信するには、マルチキャストグループを作成します。マルチキャストを使用すると、送信元は 1 つのマルチキャストグループにデータを送信できます。その後、データは受信側デバイスのグループに配信されます。

FUOTA およびマルチキャストグループに対する AWS IoT Core for LoRaWAN のサポートは、LoRa Alliance の次の仕様に基づいています。

- LoRaWAN Remote Multicast Setup Specification、TS005-2.0.0
- LoRaWAN Fragmented Data Block Transportation Specification、TS004-2.0.0
- LoRaWAN Application Layer Clock Synchronization Specification、TS003-2.0.0

> **Note**
>
> AWS IoT Core for LoRaWAN では、LoRa Alliance 仕様に従い、クロック同期が自動的に実行されます。ClockSync シグナリングを使用してサーバー側の時刻をリクエストするデバイスに返信するには AppTimeReq 関数を使用します。

以下のトピックでは、マルチキャストグループを作成し、FUOTA を実行する方法を示します。

トピック

- マルチキャストおよび FUOTA 設定用のデバイスを準備する
- マルチキャストグループを作成してダウンリンクのペイロードを複数のデバイスに送信する
- AWS IoT Core for LoRaWAN デバイス用の Firmware Updates Over-The-Air (FUOTA)

マルチキャストおよび FUOTA 設定用のデバイスを準備する

AWS IoT Core for LoRaWAN にワイヤレスデバイスを追加すると、コンソールまたは CLI を使用して、マルチキャストおよび FUOTA 設定用にワイヤレスデバイスを準備できます。初めて設定を行う場合は、コンソールを使用することをお勧めします。マルチキャストグループを管理し、グループに多数のデバイスを追加または削除するには、CLI を使用して多数のリソースを管理することをお勧めします。

GenAppKey および FPorts

ワイヤレスデバイスを追加する際、デバイスをマルチキャストグループに追加したり FUOTA を実行する前に、次のパラメータを設定します。これらのパラメータを設定する前に、お使いのデバイスが FUOTA およびマルチキャストをサポートしており、ワイヤレスデバイスの仕様が 0TAA v1.1 または 0TAAv1.0.x であることを確認してください。

- GenAppKey: LoRaWAN バージョン 1.0.x をサポートし、マルチキャストグループを使用するデバイスの場合、GenAppKey はマルチキャストグループのセッションキーの取得元となるデバイス固有のルートキーです。

> **Note**
>
> ワイヤレス仕様 0TAA v1.1 を使用する LoRaWAN デバイスの場合、AppKey は、GenAppKey と同じ目的で使用されます。

データ転送を開始するパラメータを設定するため、AWS IoT Core for LoRaWAN によりエンドデバイスにセッションキーが配布されます。LoRaWAN のバージョンの詳細については、「LoRaWAN バージョン」を参照してください。

> **Note**
>
> AWS IoT Core for LoRaWAN には、暗号化された形式で提供される GenAppKey の情報が保存されます。

- FPorts: FUOTA およびマルチキャストグループの LoRaWAN 仕様に従い、AWS IoT Core for LoRaWAN により FPorts パラメータの次のフィールドにデフォルト値が割り当てられます。次のいずれかの FPort 値が既に割り当てられている場合、1 から 223 までの別の値を選択できます。
  - Multicast: 200

    この FPort 値は、マルチキャストグループに使用されます。
  - FUOTA: 201

    この FPort 値は FUOTA に使用されます。
  - ClockSync: 202

    この FPort 値は、クロック同期に使用されます。

マルチキャストおよび FUOTA 用のデバイスプロファイル

マルチキャストセッションの開始時、グループ内のデバイスへのダウンリンクメッセージの送信に、クラス B またはクラス C の配信ウィンドウが使用されます。マルチキャストおよび FUOTA 用に追加するデバイスでは、クラス B またはクラス C のオペレーションモードがサポートされている必要があります。デバイスがサポートするデバイスクラスに応じて、クラス B モードかクラス C モードのいずれか、または両方が有効になっているデバイスのデバイスプロファイルを選択します。

デバイスプロファイルの詳細については、「プロファイルを AWS IoT Core for LoRaWAN に追加する」を参照してください。

コンソールを使用してマルチキャストおよび FUOTA 用のデバイスを準備する

コンソールを使用して、マルチキャストの設定および FUOTA の FPorts パラメータと GenAppKey パラメータを指定するには、次の操作を実行します。

1. AWS IoT コンソールの [Devices] (デバイス) ハブに移動し、[Add wireless device] (ワイヤレスデバイスの追加) を選択します。
2. [Wireless device specification] (ワイヤレスデバイスの仕様) を選択します。デバイスでは、デバイスのアクティベーション用に OTAA を使用する必要があります。OTAA v1.0.x または OTAA v1.1 を選択すると、[FUOTA configuration-Optional] (FUOTA 設定-オプション) セクションが表示されます。
3. ワイヤレスデバイスの EUI (拡張一意識別子) パラメータを入力します。
4. [FUOTA configuration-Optional] (FUOTA 設定-オプション) セクションを展開し、[This device supports firmware updates over the air (FUOTA)] (このデバイスは FUOTA をサポートしています) を選択します。これにより、マルチキャスト、FUOTA、クロック同期の FPort の値を入力できます。ワイヤレスデバイスの仕様に `OTAA v1.0.x` を選択した場合、GenAppKey を入力します。
5. プロファイルとルーティングメッセージの送信先を選択して、AWS IoT Core for LoRaWAN にデバイスを追加します。デバイスにリンクされているデバイスプロファイルについては、[Supports Class B] (クラス B をサポート) または [Supports Class C] (クラス C をサポート) モードのどちらか一方または両方を選択していることを確認してください。

> **Note**
>
> FUOTA の設定パラメータを指定するには、AWS IoT コンソールの [Devices] (デバイス) ハブを使用する必要があります。これらのパラメータは、AWS IoT コンソールの [Intro] (Intro) ページを使用してデバイスをオンボードしている場合は表示されません。

ワイヤレスデバイスの仕様とデバイスのオンボーディングについての詳細は、「ワイヤレスデバイスを AWS IoT Core for LoRaWAN に追加する」を参照してください。

> **Note**
>
> これらのパラメータは、ワイヤレスデバイスの作成時にのみ指定できます。既存のデバイスの更新時に、パラメータを変更または指定することはできません。

API オペレーションを使用してマルチキャストおよび FUOTA 用のデバイスを準備する

マルチキャストグループを使用したり FUOTA を実行するには、CreateWirelessDevice API オペレーションまたは create-wireless-device CLI コマンドを使用して、これらのパラメータを設定します。アプリケーションキーと FPorts パラメータを指定することに加えて、デバイスにリンクされているデバイスプロファイルが、クラス B モードまたはクラス C モードのいずれか、または両方をサポートしていることを確認してください。

create-wireless-device コマンドへの入力として input.json ファイルを指定できます。

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

各パラメータの意味は次のとおりです。

input.json の内容

```
{
    "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
}
```

使用可能な CLI コマンドの詳細については、AWS CLI リファレンスを参照してください。

> **Note**
>
> これらのパラメータの値を指定した後は、UpdateWirelessDevice API オペレーションを使用して更新を行うことはできません。代わりに、パラメータ GenAppKey および FPorts の値を持つ新しいデバイスを作成できます。

これらのパラメータに指定された値に関する情報を取得するには、GetWirelessDevice API オペレーションまたは get-wireless-device CLI コマンドを使用します。

次のステップ

パラメータを設定したら、マルチキャストグループと FUOTA タスクを作成して、ダウンリンクのペイロードを送信したり、LoRaWAN デバイスのファームウェアを更新できます。

- マルチキャストグループの作成についての詳細は、「マルチキャストグループを作成しグループにデバイスを追加する」を参照してください。
- FUOTA タスクの作成についての詳細は、「FUOTA タスクを作成およびファームウェアイメージの指定」を参照してください。

マルチキャストグループを作成してダウンリンクのペイロードを複数のデバイスに送信する

ダウンリンクのペイロードを複数のデバイスに送信するには、マルチキャストグループを作成します。マルチキャストを使用すると、送信元は 1 つのマルチキャストアドレスにデータを送信できます。その後、データは受信側デバイスのグループ全体に配信されます。

マルチキャストグループ内のデバイスでは、同じマルチキャストアドレス、セッションキー、およびフレームカウンターが共有されます。同じセッションキーを使用することで、マルチキャストグループ内のデバイスで、ダウンリンクの送信開始時にメッセージを復号化できます。マルチキャストグループではダウンリンクのみがサポートされています。ダウンリンクのペイロードがデバイスによって受信されたかどうかは確認されません。

AWS IoT Core for LoRaWAN のマルチキャストグループでは、次のことが行えます。

- デバイスプロファイル、RFRegion、またはデバイスクラスを使用して、デバイスのリストをフィルタリングし、これらのデバイスをマルチキャストグループに追加します。

- 48 時間の配信時間内に、1 つ以上のペイロードのダウンリンクメッセージをマルチキャストグループ内のデバイスにスケジュールして送信します。
- ダウンリンクメッセージを受信するために、マルチキャストセッションの開始時に、デバイスを一時的にクラス B モードまたはクラス C モードに切り替えます。
- マルチキャストグループの設定とそのデバイスの状態をモニタリングし、問題のトラブルシューティングも行います。
- Firmware Updates-Over-The-Air (FUOTA) を使用して、マルチキャストグループ内のデバイスにファームウェアの更新を安全にデプロイします。

次の動画では、AWS IoT Core for LoRaWAN マルチキャストグループの作成方法について説明し、グループにデバイスを追加し、グループにダウンリンクメッセージをスケジュールするプロセスについて説明します。

次に、マルチキャストグループを作成し、ダウンリンクメッセージをスケジュールする方法を示します。

トピック

- マルチキャストグループを作成しグループにデバイスを追加する
- マルチキャストグループおよびグループ内のデバイスのステータスのモニタリングとトラブルシューティング
- マルチキャストグループ内のデバイスに送信するダウンリンクメッセージをスケジュールする

マルチキャストグループを作成しグループにデバイスを追加する

コンソールまたは CLI を使用して、マルチキャストグループを作成できます。初めてマルチキャストグループを作成する場合は、コンソールを使用してマルチキャストグループを追加することをお勧めします。マルチキャストグループの管理、グループからのデバイスの追加または削除には、CLI を使用できます。

追加したエンドデバイスとシグナリングを交換すると、AWS IoT Core for LoRaWAN により エンドデバイスとの共有キーが確立され、データ転送のパラメータが設定されます。

前提条件

マルチキャストグループを作成しグループにデバイスを追加するには、次の操作を実行します。

- FUOTA の設定パラメータ GenAppKey および FPorts を指定して、マルチキャストおよび FUOTA の設定用にデバイスを準備します。詳細については、「マルチキャストおよび FUOTA 設定用のデバイスを準備する」を参照してください。
- デバイスがクラス B またはクラス C のオペレーションモードをサポートしているかどうかを確認します。デバイスがサポートするデバイスクラスに応じて、[Supports Class B] (クラス B をサポート) または [Supports Class C] (クラス C をサポート) モードの一方または両方が有効なデバイスプロファイルを選択します。デバイスプロファイルの詳細については、「プロファイルを AWS IoT Core for LoRaWAN に追加する」を参照してください。

  マルチキャストセッションの開始時、グループ内のデバイスへのダウンリンクメッセージの送信に、クラス B またはクラス C の配信ウィンドウが使用されます。

コンソールを使用してマルチキャストグループを作成する

コンソールを使用してマルチキャストグループを作成するには、AWS IoT コンソールの [Multicast groups] (マルチキャストグループ) ページに移動して、[Create multicast group] (マルチキャストグループの作成) を選択します。

1. マルチキャストグループの作成

   マルチキャストグループを作成するには、グループのマルチキャストプロパティおよびタグを指定します。

   1. マルチキャストプロパティの指定

      マルチキャストプロパティを指定するには、マルチキャストグループについての次の情報を入力します。

      - Name (名前): マルチキャストグループの一意の名前を入力します。名前には、文字、数字、ハイフン、またはアンダースコアのみを使用できます。スペースを含めることはできません。
      - Description (説明): マルチキャストグループの説明をオプションで提供できます。説明の長さは最大 2,048 文字です。

   2. マルチキャストグループのタグ

      オプションで、マルチキャストグループの Tags (タグ) として任意のキーと値の組み合わせを指定できます。マルチキャストグループの作成を続行するには、[Next] (次へ) を選択します。

2. マルチキャストグループにデバイスを追加

   個々のデバイスまたはデバイスのグループをマルチキャストグループに追加できます。デバイスを追加するには、次の操作を実行します。

   1. RFRegion の指定

      マルチキャストグループの 周波数帯域 または RFRegion を指定します。マルチキャストグループの RFRegion は、マルチキャストグループに追加するデバイスの RFRegion と一致する必要があります。RFRegion の詳細については、「ゲートウェイとデバイス接続用の LoRa 周波数帯域の選択を検討する」を参照してください。

   2. マルチキャストのデバイスクラスを選択する

      マルチキャストセッションの開始時に、マルチキャストグループ内のデバイスをクラス B モードまたはクラス C モードのどちらに切り替えるかを選択します。クラス B のセッションでは通常のダウンリンクスロットでダウンリンクメッセージを受信でき、クラス C ではいつでもダウンリンクメッセージを受信できます。

   3. グループに追加するデバイスを選択

      マルチキャストグループにデバイスを個別に追加するか、一括で追加するかを選択します。

      - デバイスを個別に追加するには、グループに追加する各デバイスのワイヤレスデバイス ID を入力します。
      - デバイスを一括で追加するには、追加するデバイスをデバイスプロファイルまたはタグでフィルタリングできます。デバイスプロファイルの場合、クラス B、クラス C、または両方のデバイスクラスをサポートするプロファイルを持つデバイスを追加できます。

   4. マルチキャストグループを作成するには、[Create] (作成) を選択します。

      マルチキャストグループの詳細と追加したデバイスがグループに表示されます。マルチキャストグループおよびデバイスのステータスに関する情報、および問題のトラブルシューティングについては、「マルチキャストグループおよびグループ内のデバイスのステータスのモニタリングとトラブルシューティング」を参照してください。

マルチキャストグループの作成後は、[Action] (アクション) を選択して、デバイスを編集、削除、またはマルチキャストグループに追加できます。デバイスを追加したら、ダウンリンクのペイロードがグループ内のデバイスに送信されるようにセッションをスケジュールできます。

API を使用してマルチキャストグループを作成

API を使用してマルチキャストグループを作成し、グループにデバイスを追加するには、次の操作を実行します。

1. マルチキャストグループの作成

   マルチキャストグループの作成には、CreateMulticastGroup API オペレーションまたは create-multicast-group CLI コマンドを使用します。create-multicast-group コマンドへの入力として input.json ファイルを指定できます。

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

   各パラメータの意味は次のとおりです。

   input.json の内容

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

   マルチキャストグループを作成したら、次の API オペレーションまたは CLI コマンドを使用して、マルチキャストグループに関する情報を更新、削除、取得できます。

   - UpdateMulticastGroup 、、または update-multicast-group
   - GetMulticastGroup 、、または get-multicast-group
   - ListMulticastGroups 、、または list-multicast-groups
   - DeleteMulticastGroup 、、または delete-multicast-group
2. マルチキャストグループにデバイスを追加

   個別に、または一括でマルチキャストグループにデバイスを追加できます。

- マルチキャストグループにデバイスを一括で追加するには、StartBulkAssociateWirelessDeviceWithMulticastGroup API オペレーションまたは start-bulk-associate-wireless-device-with-multicast-group CLI コマンドを使用します。マルチキャストグループに関連付けるデバイスを一括でフィルタリングするには、クエリ文字列を指定します。次に、指定された ID がリンクされたデバイスプロファイルを持つデバイスのグループを追加する方法を示します。

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

各パラメータの意味は次のとおりです。

input.json の内容

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

この multicast-groups/d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf/bulk は、デバイスをグループに関連付けるために使用する URL です。

- マルチキャストグループにデバイスを個別に追加するには、AssociateWirelessDeviceWithMulticastGroup API オペレーションまたは associate-wireless-device-with-multicast-group CLI コマンドを使用します。グループに追加する各デバイスのワイヤレスデバイス ID を指定します。

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

マルチキャストグループを作成したら、次の API オペレーションまたは CLI コマンドを使用して、マルチキャストグループに関する情報を取得したり、デバイスの関連付けを解除できます。

- DisassociateWirelessDeviceFromMulticastGroup 、、または disassociate-wireless-device-from-multicast-group
- StartBulkDisassociateWirelessDeviceFromMulticastGroup 、、または start-bulk-disassociate-wireless-device-from-multicast-group
- ListWirelessDevices 、、または list-wireless-devices

> **Note**
>
> ListWirelessDevices API オペレーションは、一般的なワイヤレスデバイス、およびマルチキャストグループまたは FUOTA タスクに関連付けられているワイヤレスデバイスを一覧表示するために使用できます。
>
> - マルチキャストグループに関連付けられているワイヤレスデバイスを一覧表示するには、フィルタとして MulticastGroupID を持つ ListWirelessDevices API オペレーションを使用します。
> - FUOTA タスクに関連付けられているワイヤレスデバイスを一覧表示するには、フィルタとして ListWirelessDevices を持つ FuotaTaskID API オペレーションを使用します。

次のステップ

マルチキャストグループを作成してデバイスを追加したら、デバイスの追加を続行したり、マルチキャストグループとデバイスのステータスをモニタリングできます。デバイスがグループに正常に追加されている場合、ダウンリンクメッセージがデバイスに送信されるように設定およびスケジュールできます。ダウンリンクメッセージを送信するには、デバイスのステータスが [Multicast setup ready] (マルチキャストのセットアップの準備が完了しました) である必要があります。ダウンリンクメッセージをスケジュールすると、ステータスは [Session attempting] (セッションの試行中) に変わります。詳細については、「マルチキャストグループ内のデバイスに送信するダウンリンクメッセージをスケジュールする」を参照してください。

マルチキャストグループ内のデバイスのファームウェアを更新する場合、AWS IoT Core for LoRaWAN で Firmware Updates Over-The-Air (FUOTA) を実行できます。詳細については、「AWS

IoT Core for LoRaWAN デバイス用の Firmware Updates Over-The-Air (FUOTA)」を参照してください。

デバイスが追加されなかった場合、またはマルチキャストグループやデバイスのステータスにエラーが表示された場合は、エラーにカーソルを合わせると詳細情報を取得でき、解決することができます。それでもエラーが表示される場合は、問題のトラブルシューティングと解決方法の詳細について、「マルチキャストグループおよびグループ内のデバイスのステータスのモニタリングとトラブルシューティング」を参照してください。

マルチキャストグループおよびグループ内のデバイスのステータスのモニタリングとトラブルシューティング

デバイスを追加してマルチキャストグループを作成したら、AWS Management Console を開きます。AWS IoT コンソールの [Multicast groups] (マルチキャストグループ) ページに移動して、作成したマルチキャストグループを選択し詳細を表示します。マルチキャストグループに関する情報、追加されたデバイスの数、デバイスのステータスに関する詳細が表示されます。ステータス情報を使用して、マルチキャストセッションの進行状況を追跡し、エラーのトラブルシューティングを行うことができます。

マルチキャストグループのステータス

マルチキャストグループには、AWS Management Console で次のステータスメッセージのいずれかが表示されることがあります。

- 保留中

  このステータスは、マルチキャストグループを作成したが、まだマルチキャストセッションがないことを示します。グループが作成されると、このステータスメッセージが表示されます。この間、マルチキャストグループを更新したり、デバイスをグループに関連付けたり、関連付けを解除したりできます。ステータスが [Pending] (保留中) から変更された後は、他のデバイスをグループに追加することは使用できません。

- セッションの試行中

  デバイスがマルチキャストグループに正常に追加された後、グループにスケジュールされたマルチキャストセッションがある場合、このステータスメッセージが表示されます。この間、デバイスを更新したりマルチキャストグループに追加することはできません。マルチキャストセッションをキャンセルすると、グループのステータスは [Pending] (保留中) に変わります。

- セッション中

マルチキャストセッションで、セッション時間が最も早い場合、このステータスメッセージが表示されます。マルチキャストグループは、ファームウェアの更新セッションが進行中の FUOTA タスクに関連付けられている場合も、この状態のままです。

セッションに関連した FUOTA タスクがなく、セッション時間がタイムアウトの時間を超えたためにマルチキャストセッションがキャンセルされた場合、またはユーザーがマルチキャストセッションをキャンセルした場合、グループのステータスは [Pending] (保留中) に変わります。

- 削除の待機中

  マルチキャストグループを削除すると、グループのステータスが [Delete waiting] (削除の待機中) に変わります。削除は永続的で、元に戻すことはできません。この操作には時間がかかる場合があり、マルチキャストグループが削除されるまでグループのステータスは [Delete_Waiting] (削除の待機中) になります。マルチキャストグループがこの状態になった後は、他の状態には移行できません。

マルチキャストグループ内のデバイスのステータス

マルチキャストグループ内のデバイスには、AWS Management Console で次のステータスメッセージのいずれかが表示されます。各ステータスメッセージにカーソルを合わせると、そのメッセージが示す内容の詳細を確認できます。

- パッケージの試行中

  デバイスがマルチキャストグループに関連付けられると、デバイスのステータスは [Package attempting] (パッケージの試行中) になります。このステータスは、AWS IoT Core for LoRaWAN により、デバイスがマルチキャストの設定とオペレーションをサポートしているかどうかがまだ確認されていないことを示します。

- パッケージはサポートされていません

  デバイスがマルチキャストグループに関連付けられると、AWS IoT Core for LoRaWAN によりデバイスのファームウェアがマルチキャストの設定とオペレーションに対応しているかどうかが確認されます。サポートされているマルチキャストパッケージがデバイスにない場合、ステータスは [Package unsupported] (パッケージはサポートされていません) になります。このエラーを解決するには、デバイスのファームウェアがマルチキャストの設定とオペレーションに対応しているかどうかを確認します。

- マルチキャストのセットアップを試行中

マルチキャストグループに関連付けられているデバイスがマルチキャストの設定およびオペレーションに対応している場合、ステータスは [Multicast setup attempting] (マルチキャストのセットアップを試行中) になります。このステータスは、デバイスがマルチキャストのセットアップをまだ完了していないことを示します。

- マルチキャストのセットアップの準備完了

  デバイスでマルチキャストのセットアップが完了し、マルチキャストグループに追加されています。このステータスは、デバイスでマルチキャストセッションの準備が完了しており、ダウンリンクメッセージをそれらのデバイスに送信できることを示します。ステータスには、FUOTA を使用してグループ内のデバイスのファームウェアを更新できるタイミングも示されます。

- セッションの試行中

  マルチキャストグループ内のデバイス用にマルチキャストセッションがスケジュールされています。マルチキャストグループのセッション開始時のデバイスのステータスは [Session attempting] (試行中のセッション) で、セッション用にクラス B またはクラス C の配信ウィンドウを開始できるかどうかのリクエストが送信されます。マルチキャストセッションの設定にかかる時間がタイムアウトの時間を超えた場合、またはユーザーがマルチキャストセッションをキャンセルした場合、ステータスは [Multicast setup done] (マルチキャストのセットアップが完了しました) に変わります。

- セッション中

  このステータスは、クラス B またはクラス C の配信ウィンドウが開始され、デバイスでマルチキャストセッションが進行中であることを示します。この間、ダウンリンクメッセージは AWS IoT Core for LoRaWAN からマルチキャストグループ内のデバイスに送信されます。セッション時間を更新すると、現在のセッションが上書きされ、ステータスが [Session attempting] (セッションの試行中) に変わります。セッション時間の終了時、またはユーザーがマルチキャストセッションをキャンセルした場合、ステータスは [Multicast setup ready] (マルチキャストのセットアップの準備が完了しました) になります。

次のステップ

これまで、マルチキャストグループおよびグループ内のデバイスのさまざまなステータス、デバイスがマルチキャストの設定に対応していない場合などの問題のトラブルシューティング方法について学習したので、ダウンリンクメッセージがデバイスに送信されるようスケジュールを行えます。マルチキャストグループは [In session] (セッション中) になります。ダウンリンクメッセージのスケジュー

ルについては、「マルチキャストグループ内のデバイスに送信するダウンリンクメッセージをスケジュールする」を参照してください。

マルチキャストグループ内のデバイスに送信するダウンリンクメッセージをスケジュールする

マルチキャストグループがデバイスに正常に追加されたら、マルチキャストセッションを開始し、それらのデバイスに送信されるダウンリンクメッセージを設定できます。ダウンリンクメッセージは、48 時間以内にスケジュールする必要があります。また、マルチキャストの開始時刻は、現在の時刻から 30 分以上後である必要があります。

> **Note**
>
> マルチキャストグループのデバイスでは、ダウンリンクメッセージを受信したときに確認応答ができません。

前提条件

ダウンリンクメッセージを送信する前に、マルチキャストグループを作成し、ダウンリンクメッセージを送信するグループにデバイスを正常に追加しておく必要があります。マルチキャストセッションの開始時刻がスケジュールされた後に、デバイスを追加することはできません。詳細については、「マルチキャストグループを作成しグループにデバイスを追加する」を参照してください。

いずれかのデバイスが正常に追加されなかった場合、マルチキャストグループとデバイスステータスには、エラーの解決に役立つ情報が含まれます。それでもエラーが続く場合は、エラーのトラブルシューティングについて「マルチキャストグループおよびグループ内のデバイスのステータスのモニタリングとトラブルシューティング」を参照してください。

コンソールを使用してダウンリンクメッセージをスケジュールする

コンソールを使用してダウンリンクメッセージを送信するには、AWS IoT コンソールの [Multicast groups] (マルチキャストグループ) ページに移動して、作成したマルチキャストグループを選択します。マルチキャストグループの詳細ページで、[Schedule downlink message] (ダウンリンクメッセージのスケジュール) を選択し、その後 [Schedule downlink session] (ダウンリンクセッションのスケジュール) を選択します。

1. ダウンリンクメッセージのウィンドウをスケジュールする

   マルチキャストグループ内のデバイスにダウンリンクメッセージを送信するようタイムウィンドウを設定できます。ダウンリンクメッセージは 48 時間以内にスケジュールする必要があります。

   マルチキャストセッションをスケジュールするには、次のパラメータを指定します。

   - [Start date] (開始日) および [Start time] (開始時刻): 開始日および開始時刻は、現在の時刻から 30 分以上後で 48 時間より前である必要があります。

     > **Note**
     >
     > 指定する時刻は UTC なので、ダウンリンクウィンドウをスケジュールする際は、タイムゾーンとの時差を確認してください。

   - Session timeout (セッションタイムアウト): ダウンリンクメッセージが受信されなかった場合、マルチキャストセッションがタイムアウトするまでの時間です。タイムアウトの最短時間は 60 秒です。最大タイムアウト値は、クラス B のマルチキャストグループで 2 日、クラス C のマルチキャストグループで 18 時間です。

2. ダウンリンクメッセージの設定

   ダウンリンクメッセージを設定するには、次のパラメータを指定します。

   - Data rate (データレート): ダウンリンクメッセージのデータレートを選択します。データレートは、RFRegion およびペイロードサイズによって異なります。デフォルトのデータレートは、US915 リージョンで 8、EU868 リージョンで 0 です。
   - Frequency (頻度): ダウンリンクメッセージを送信する頻度を選択します。メッセージの競合を回避するため、RFRegion に応じて使用可能な頻度を選択します。
   - FPort: ダウンリンクメッセージをデバイスに送信するために使用できる周波数ポートを選択します。
   - Payload (ペイロード): データレートに応じて、ペイロードの最大サイズを指定します。デフォルトのデータレートを使用すると、ペイロードの最大サイズを US915 RfRegion では 33 バイト、EU868 RfRegion では 51 バイトに設定できます。より大きなデータレートを使用すると、最大サイズが 242 バイトのペイロードまで転送できます。

   ダウンリンクメッセージをスケジュールするには、[Schedule] (スケジュール) を選択します。

API を使用してダウンリンクメッセージをスケジュールする

API を使用してダウンリンクメッセージをスケジュールするには、StartMulticastGroupSession API オペレーションまたは start-multicast-group-session CLI コマンドを使用します。

次の API オペレーションまたは CLI コマンドを使用して、マルチキャストグループに関する情報を取得したり、マルチキャストグループを削除できます。

- GetMulticastGroupSession 、、または get-multicast-group-session
- DeleteMulticastGroupSession 、、または delete-multicast-group-session

セッションの開始後にマルチキャストグループにデータを送信するには、SendDataToMulticastGroup API オペレーションまたは send-data-to-multicast-group CLI コマンドを使用します。

次のステップ

ダウンリンクメッセージがデバイスに送信されるように設定すると、セッションの開始時にメッセージが送信されます。マルチキャストグループ内のデバイスでは、メッセージが受信されたかどうかを確認できません。

追加のダウンリンクメッセージを設定する

マルチキャストグループ内のデバイスに送信する追加のダウンリンクメッセージを設定することもできます。

- コンソールから追加のダウンリンクメッセージを設定するには、次の操作を実行します。
  1. AWS IoT コンソールの [Multicast groups] (マルチキャストグループ) ページに移動して、作成したマルチキャストグループを選択します。
  2. マルチキャストグループの詳細ページで、[Schedule downlink message] (ダウンリンクメッセージのスケジュール) を選択し、その後 [Configure additional downlink message] (追加のダウンリンクメッセージを設定) を選択します。
  3. 最初にダウンリンクメッセージに設定したのと同様に [Data rate] (データレート)、[Frequency] (頻度)、[FPort]、および [Payload] (ペイロード) のパラメータを指定します。
- API または CLI を使用して追加のダウンリンクメッセージを設定するには、追加のダウンリンクメッセージごとに SendDataToMulticastGroup API オペレーションまたは send-data-to-multicast-group CLI コマンドを呼び出します。

セッションスケジュールの更新

マルチキャストセッションに新しい開始日時が使用されるよう、セッションスケジュールを更新することもできます。新しいセッションスケジュールは、以前にスケジュールされたセッションに上書きされます。

> **Note**
>
> マルチキャストセッションは、必要な場合にのみ更新してください。更新により、デバイスのグループが長時間起動状態になり、バッテリーを消費する可能性があります。

- コンソールからセッションスケジュールを更新するには、次の操作を実行します。
  1. AWS IoT コンソールの [Multicast groups] (マルチキャストグループ) ページに移動して、作成したマルチキャストグループを選択します。
  2. マルチキャストグループの詳細ページで、[Schedule downlink message] (ダウンリンクメッセージのスケジュール) を選択し、その後 [Update session schedule] (セッションスケジュールの更新) を選択します。
  3. 最初のダウンリンクメッセージで指定したのと同様に、[Start date] (開始日)、[Start time] (開始時間)、および [Session timeout] (セッションタイムアウト) のパラメータを指定します。
- API または CLI からセッションスケジュールを更新するには、`StartMulticastGroupSession` API オペレーションまたは `start-multicast-group-session` CLI コマンドを使用します。

AWS IoT Core for LoRaWAN デバイス用の Firmware Updates Over-The-Air (FUOTA)

Firmware Updates Over-The-Air (FUOTA) を使用して、AWS IoT Core for LoRaWAN デバイスにファームウェアの更新をデプロイします。

FUOTA を使用すると、ファームウェアの更新を個々のデバイスまたはデバイスのグループに送信できます。マルチキャストグループを作成して、ファームウェアの更新を複数のデバイスに送信することもできます。まず、デバイスをマルチキャストグループに追加し、その後ファームウェアの更新イメージをそれらのすべてのデバイスに送信します。ファームウェアのイメージにデジタル署名して、イメージを受信するデバイスがそれが正しいソースから来ていることを確認できるようにすることをお勧めします。

AWS IoT Core for LoRaWAN の FUOTA では、次を行うことができます。

- 新しいファームウェアイメージまたはデルタイメージを単一のデバイスやデバイスのグループにデプロイします。
- 新しいファームウェアがデバイスに導入された後、そのファームウェアの信頼性と完全性を検証します。
- デプロイの進行状況をモニタリングし、失敗した場合に問題をデバッグします。

FUOTA およびマルチキャストグループに対する AWS IoT Core for LoRaWAN のサポートは、LoRa Alliance の次の仕様に基づいています。

- LoRaWAN Remote Multicast Setup Specification、TS005-2.0.0
- LoRaWAN Fragmented Data Block Transportation Specification、TS004-2.0.0
- LoRaWAN Application Layer Clock Synchronization Specification、TS003-2.0.0

> **Note**
>
> AWS IoT Core for LoRaWAN では、LoRa Alliance 仕様に従い、クロック同期が自動的に実行されます。ClockSync シグナリングを使用してサーバー側の時刻をリクエストするデバイスに返信するには `AppTimeReq` 関数を使用します。

次の動画では、AWS IoT Core for LoRaWAN FUOTA タスクの作成方法について説明し、タスクにデバイスを追加し、FUOTA タスクをスケジュールするプロセスについて説明します。

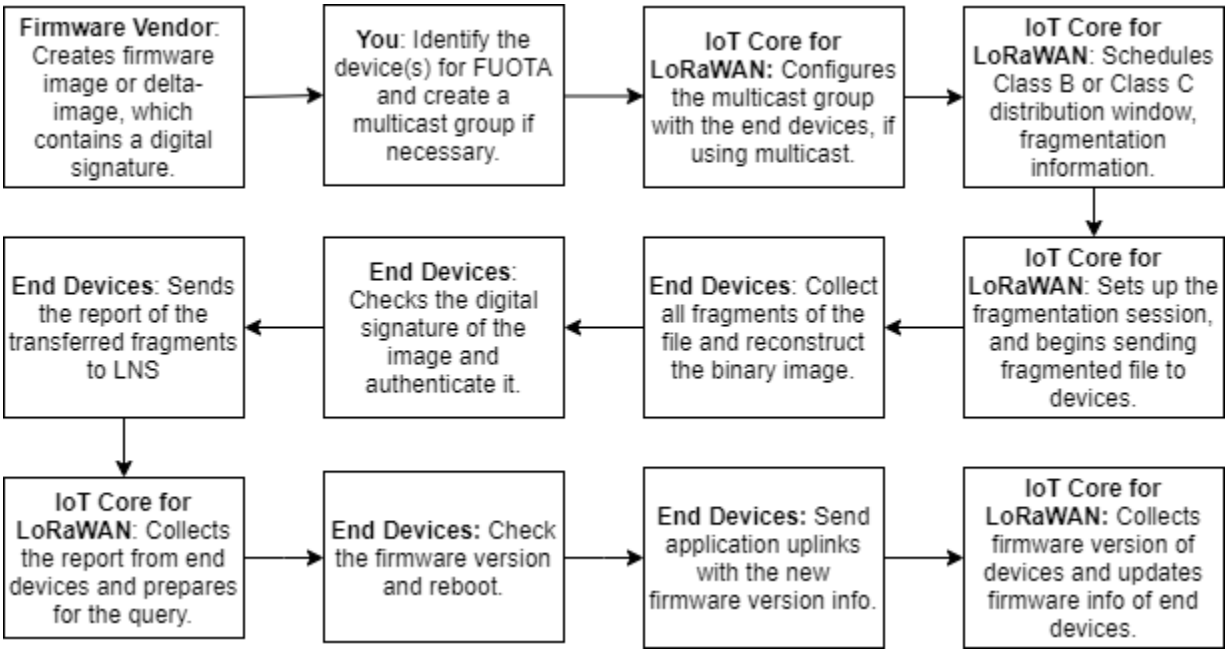
次のトピックでは、FUOTA の更新方法を示します。

- FUOTA プロセスの概要
- FUOTA タスクを作成およびファームウェアイメージの指定
- は デバイスおよびマルチキャストグループを FUOTA タスクに追加して FUOTA セッションをスケジュールする
- FUOTA タスクとタスクに追加されたデバイスのステータスを監視およびトラブルシューティングします。

FUOTA プロセスの概要

次の図は、エンドデバイスに対して AWS IoT Core for LoRaWAN によりFUOTA プロセスがどのように実行されるのかを示しています。FUOTA セッションに個々のデバイスを追加する場合は、マル

チキャストグループの作成および設定の手順をスキップできます。FUOTA セッションにデバイスを直接追加できます。その後、AWS IoT Core for LoRaWAN によりファームウェアの更新プロセスが開始されます。



デバイス用に FUOTA を実行するには、最初にデジタル署名されたファームウェアイメージを作成し、FUOTA タスクに追加するデバイスおよびマルチキャストグループを設定します。FUOTA セッションを開始すると、エンドデバイスによりすべてのフラグメントが収集され、フラグメントからイメージが再構築され、ステータスが AWS IoT Core for LoRaWAN に報告されます。その後、新しいファームウェアイメージが適用されます。

FUOTA プロセスのさまざまな手順を次に示します。

1. デジタル署名したファームウェアイメージまたはデルタイメージを作成する

   AWS IoT Core for LoRaWAN で LoRaWAN デバイス用の FUOTA を実行するには、ファームウェアの更新を無線で送信する際に、ファームウェアイメージまたはデルタイメージにデジタル署名しておくことをお勧めします。その後、画像を受信するデバイスでは、それが正しいソースから来ていることを確認できます。

   ファームウェアイメージのサイズは 1 メガバイトを超えてはいけません。ファームウェアのサイズが大きいほど、更新プロセスの完了に時間がかかります。データ転送を高速化したい場合、または新しいイメージが 1 メガバイトを超える場合は、デルタイメージを使用します。デルタイメージは、新しいファームウェアイメージと以前のイメージの間のデルタとなる新しいイメージの一部です。

> **Note**
>
> AWS IoT Core for LoRaWAN では、デジタル署名作成ツールおよびファームウェアバージョンの管理システムは提供されていません。サードパーティー製のツールを使用して、ファームウェアイメージのデジタル署名を生成できます。ARM Mbed GitHub リポジトリに埋め込まれているようなデジタル署名ツールを使用することをお勧めします。これには、デルタイメージを生成するためのツールと、デバイスがそのイメージを使用するためのツールも含まれています。

2. FUOTA のデバイスの特定および設定

   FUOTA のデバイスを特定したら、ファームウェアの更新を個々のデバイスまたは複数のデバイスに送信します。

   - ファームウェアの更新を複数のデバイスに送信するには、マルチキャストグループを作成し、エンドデバイスでマルチキャストグループを設定します。詳細については、「マルチキャストグループを作成してダウンリンクのペイロードを複数のデバイスに送信する」を参照してください。
   - 個々のデバイスにファームウェアの更新を送信するには、それらのデバイスを FUOTA セッションに追加し、ファームウェアの更新を実行します。

3. 配信ウィンドウをスケジュールしてフラグメンテーションセッションをセットアップする

   マルチキャストグループを作成した場合、クラス B またはクラス C の配信ウィンドウを指定して、デバイスが AWS IoT Core for LoRaWAN からフラグメントを受信できるタイミングを決定できます。クラス B またはクラス C モードに切り替える前に、デバイスがクラス A で動作している可能性があります。また、セッションの開始時刻を指定する必要があります。

   クラス B またはクラス C のデバイスは、指定された配信ウィンドウで起動し、ダウンリンクパケットの受信を開始します。クラス C モードで動作するデバイスは、クラス B のデバイスよりも多くの電力を消費します。詳細については、「デバイスクラス」を参照してください。

4. エンドデバイスによる AWS IoT Core for LoRaWAN へのステータスの報告およびファームウェアイメージの更新

   フラグメンテーションセッションの設定後、エンドデバイスおよび AWS IoT Core for LoRaWAN では次の手順が実行され、デバイスのファームウェアが更新されます。

1. LoRaWan デバイスのデータレートが低いため、FUOTA プロセスを開始するために AWS IoT Core for LoRaWAN でフラグメンテーションセッションが設定され、ファームウェアイメージがフラグメント化されます。その後、これらのフラグメントがエンドデバイスに送信されます。
2. AWS IoT Core for LoRaWAN によりイメージのフラグメントが送信されたら、LoRaWan エンドデバイスでは次のタスクが実行されます。
   a. フラグメントが収集され、これらのフラグメントからバイナリイメージが再構築されます。
   b. 再構築されたイメージのデジタル署名の確認、イメージの認証が行われ、それが正しいソースから来ていることが確認されます。
   c. AWS IoT Core for LoRaWAN からのファームウェアのバージョンと現在のバージョンが比較されます。
   d. 転送済みの、フラグメント化されたイメージのステータスが AWS IoT Core for LoRaWAN に報告され、新しいファームウェアイメージが適用されます。

> **Note**
>
> 場合によっては、ファームウェアイメージのデジタル署名が確認される前に、エンドデバイスにより転送済みのフラグメント化されたイメージのステータスが AWS IoT Core for LoRaWAN に報告されます。

FUOTA プロセスを学習したので、FUOTA タスクを作成し、タスクにデバイスを追加してファームウェアを更新できます。詳細については、「FUOTA タスクを作成およびファームウェアイメージの指定」を参照してください。

FUOTA タスクを作成およびファームウェアイメージの指定

LoRaWAN デバイスのファームウェアを更新するには、まず FUOTA タスクを作成し、アップデートに使用するデジタル署名されたファームウェアイメージを指定します。その後、デバイスおよびマルチキャストグループをタスクに追加し、FUOTA セッションをスケジュールできます。セッションが開始されると、AWS IoT Core for LoRaWAN によりフラグメンテーションセッションが設定されます。また、エンドデバイスによりフラグメントが収集され、イメージが再構築され、新しいファームウェアが適用されます。FUOTA プロセスの詳細については、「FUOTA プロセスの概要」を参照してください。

FUOTA タスクを作成し、S3 バケットに格納するファームウェアイメージまたはデルタイメージをアップロードする方法を次に示します。

前提条件

FUOTA を実行する前に、エンドデバイスがイメージの適用時にイメージの信頼性を確認できるよう、ファームウェアイメージにデジタル署名する必要があります。サードパーティー製のツールを使用して、ファームウェアイメージのデジタル署名を生成できます。ARM Mbed GitHub リポジトリに埋め込まれているようなデジタル署名ツールを使用することをお勧めします。これには、デルタイメージを生成するためのツールと、デバイスがそのイメージを使用するためのツールも含まれています。

コンソールを使用して FUOTA タスクを作成しファームウェアイメージをアップロードする

コンソールを使用して FUOTA タスクを作成し、ファームウェアイメージをアップロードするには、コンソールの [FUOTA tasks] (FUOTA タスク) タブに移動して、[Create FUOTA task] (FUOTA タスクの作成) を選択します。

1. FUOTA タスクの作成

   FUOTA タスクを作成するには、タスクのプロパティとタグを指定します。

   1. FUOTA タスクのプロパティの指定

      FUOTA タスクのプロパティを指定するには、FUOTA タスクについて次の情報を入力します。

      - Name (名前): FUOTA タスクの一意の名前を入力します。名前には、文字、数字、ハイフン、またはアンダースコアのみを使用できます。スペースを含めることはできません。
      - Description (説明): マルチキャストグループの説明をオプションで提供できます。説明フィールドは最大 2,048 文字です。
      - RFRegion: FUOTA タスクの周波数帯域を設定します。周波数帯域は、ワイヤレスデバイスまたはマルチキャストグループのプロビジョニングに使用したものと一致する必要があります。

   2. FUOTA タスクのタグ

      オプションで、FUOTA タスクの Tags (タグ) として、任意のキーと値の組み合わせを指定できます。イメージの作成を続行するには、[Next] (次へ) を選択します。

2. ファームウェアイメージのアップロード

FUOTA タスクに追加するデバイスにおけるファームウェアの更新に使用するファームウェアのイメージファイルを選択します。ファームウェアのイメージファイルは S3 バケットに保存されます。お客様に代わってファームウェアイメージにアクセスするアクセス許可を AWS IoT Core for LoRaWAN に与えることができます。ファームウェアイメージにデジタル署名して、ファームウェアの更新時にその信頼性が確認できるようにすることをお勧めします。

1. ファームウェアのイメージファイルを選択する

   ファームウェアの新しいイメージファイルを S3 バケットにアップロードするか、既に S3 バケットにアップロードされている既存のイメージを選択できます。

   > **Note**
   >
   > ファームウェアのイメージファイルのサイズは 1 メガバイトを超えてはいけません。ファームウェアのサイズが大きいほど、更新プロセスの完了に時間がかかります。

   - 既存のイメージを使用するには、[Select an existing firmware image] (既存のファームウェアイメージを選択する) を選択し、[Browse S3] (S3 をブラウズする) を選択します。その後、使用するファームウェアのイメージファイルを選択します。

     AWS IoT Core for LoRaWAN により、S3 バケット内にあるファームウェアのイメージファイルへのパスである S3 URL が設定されます。パスの形式は `s3://bucket_name/file_name` です。Amazon Simple Storage Service コンソールでファイルを表示するには、[View] (表示) を選択します。
   - 新しいファームウェアイメージをアップロードするには。

     a. [Upload a new firmware image] (新しいファームウェアイメージをアップロード) を選択し、ファームウェアイメージをアップロードします。イメージファイルは 1 メガバイトを超えてはいけません。

     b. S3 バケットを作成し、ファームウェアイメージファイルを保存するための Bucket name (バケット名) を入力するには、[Create S3 bucketS3] (S3 バケットの作成) を選択します。

2. バケットへのアクセス許可

新しいサービスロールを作成するか、既存のロールを選択して、AWS IoT Core for LoRaWAN がお客様に代わって S3 バケット内にあるファームウェアのイメージファイルにアクセスすることを許可できます。[Next] を選択します。

新しいロールを作成するには、ロール名を入力するか、空白のままにするとランダムな名前が自動的に生成されます。S3 バケットへのアクセスを許可するポリシーのアクセス許可を表示するには、[View policy permissions] (ポリシーのアクセス許可を表示する) を選択します。

S3 バケットを使用してイメージを保存し、AWS IoT Core for LoRaWAN にアクセスを許可する方法の詳細については、「S3 バケットにファームウェアファイルをアップロードし、IAM ロールを追加する」を参照してください。

3. 確認と作成

FUOTA タスクを作成するには、指定した FUOTA タスクと設定の詳細を確認し、[Create task] (タスクの作成) を選択します。

API を使用して FUOTA タスクを作成しファームウェアイメージをアップロードする

API を使用して FUOTA タスクを作成し、ファームウェアのイメージファイルを指定するには、`CreateFuotaTask` API オペレーションまたは `create-fuota-task` CLI コマンドを使用します。`create-fuota-task` コマンドへの入力として `input.json` ファイルを指定できます。API または CLI を使用する際、入力として指定するファームウェアのイメージファイルは、既に S3 バケットにアップロードされている必要があります。また、AWS IoT Core for LoRaWAN が S3 バケット内のファームウェアイメージにアクセスできるようにする IAM ロールを指定します。

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

各パラメータの意味は次のとおりです。

input.json の内容

```
{
   "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"
}
```

FUOTA タスクを作成したら、次の API オペレーションまたは CLI コマンドを使用して、FUOTA タスクに関する情報を更新、削除、取得できます。

- UpdateFuotaTask 、、または update-fuota-task
- GetFuotaTask 、、または get-fuota-task
- ListFuotaTasks 、、または list-fuota-tasks
- DeleteFuotaTask 、、または delete-fuota-task

次のステップ

FUOTA タスクを作成し、ファームウェアイメージを指定したので、ファームウェアの更新のためタスクにデバイスを追加できます。個々のデバイスまたはマルチキャストグループをタスクに追加できます。詳細については、「 は デバイスおよびマルチキャストグループを FUOTA タスクに追加して FUOTA セッションをスケジュールする」を参照してください。

は デバイスおよびマルチキャストグループを FUOTA タスクに追加して FUOTA セッションをスケジュールする

FUOTA タスクを作成したら、ファームウェアを更新するデバイスをタスクに追加できます。デバイスが FUOTA タスクに正常に追加されたら、FUOTA セッションをスケジュールしてデバイスのファームウェアを更新できます。

- デバイスの数が少ない場合、それらのデバイスを FUOTA タスクに直接追加できます。
- ファームウェアを更新するデバイスの数が多い場合、これらのデバイスをマルチキャストグループに追加して、マルチキャストグループを FUOTA タスクに追加できます。マルチキャストグループの作成と使用の詳細については、「マルチキャストグループを作成してダウンリンクのペイロードを複数のデバイスに送信する」を参照してください。

> **Note**
>
> FUOTA タスクには、個々のデバイスまたはマルチキャストグループを追加できます。デバイスとマルチキャストグループの両方をタスクに追加することはできません。

デバイスまたはマルチキャストグループを追加したら、ファームウェアの更新セッションを開始できます。AWS IoT Core for LoRaWAN ではファームウェアイメージが収集され、イメージがフラグメント化されます。その後、暗号化された形式でフラグメントが保存されます。エンドデバイスによりフラグメントが収集され、新しいファームウェアイメージが適用されます。ファームウェアの更新にかかる時間は、イメージのサイズとフラグメントの方法によって異なります。ファームウェアの更新が完了すると、AWS IoT Core for LoRaWAN で保存したファームウェアイメージの暗号化されたフラグメントは削除されます。削除後も S3 バケット内でファームウェアイメージを見つけることができます。

前提条件

FUOTA タスクにデバイスまたはマルチキャストグループを追加する前に、次の操作を実行してください。

- FUOTA タスクを作成し、ファームウェアイメージを指定しておく必要があります。詳細については、「FUOTA タスクを作成およびファームウェアイメージの指定」を参照してください。
- デバイスのファームウェアを更新するワイヤレスデバイスをプロビジョニングします。デバイスのオンボーディングの詳細については、「デバイスを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。
- 複数のデバイスのファームウェアを更新するには、それらをマルチキャストグループに追加します。詳細については、「マルチキャストグループを作成してダウンリンクのペイロードを複数のデバイスに送信する」を参照してください。
- デバイスを AWS IoT Core for LoRaWAN にオンボードする際、FUOTA の設定パラメータ `FPorts` を指定します。LoRaWAN v1.0.x デバイスを使用している場合は、`GenAppKey` も指定する必要があります。FUOTA の設定パラメータの詳細については、「マルチキャストおよび FUOTA 設定用のデバイスを準備する」を参照してください。

コンソールを使用して FUOTA タスクにデバイスを追加し FUOTA セッションをスケジュールする

コンソールを使用してデバイスまたはマルチキャストグループを追加し、FUOTA セッションをスケジュールするには、コンソールの [FUOTA tasks] (Fuota タスク) タブに移動します。次に、デバイスを追加する FUOTA タスクを選択し、ファームウェアの更新を実行します。

デバイスおよびマルチキャストグループの追加

1. FUOTA タスクには、個々のデバイスまたはマルチキャストグループを追加できます。ただし、個々のデバイスとマルチキャストグループの両方を同じ FUOTA タスクに追加することはできません。コンソールを使用してデバイスを追加するには、次の操作を実行します。

1. [FUOTA task details] (FUOTA タスクの詳細) で、[Add device] (デバイスの追加) を選択します。
2. タスクに追加するデバイスの RFRegion または周波数帯域を選択します。この値は、FUOTA タスク用に選択した RFRegion と一致する必要があります。
3. 個々のデバイスとマルチキャストグループのどちらにタスクに追加するかを選択します。
   - 個々のデバイスを追加するには、[Add individual devices] (個々のデバイスを追加する) をクリックし、FUOTA タスクに追加する各デバイスのデバイス ID を入力します。
   - マルチキャストグループを追加するには、[Add multicast groups] (マルチキャストグループを追加する) をクリックし、タスクにマルチキャストグループを追加します。デバイスのプロファイルまたはタグを使用して、タスクに追加するマルチキャストグループをフィルタリングできます。デバイスプロファイルでフィルタリングする場合、[Supports Class B] (クラス B をサポート) または [Supports Class C] (クラス C をサポート) が有効なプロファイルを持つデバイスのマルチキャストグループを選択できます。

2. FUOTA セッションのスケジュール

   デバイスまたはマルチキャストグループが正常に追加されたら、FUOTA セッションをスケジュールできます。セッションをスケジュールするには、次の操作を実行します。

   1. デバイスのファームウェアを更新する FUOTA タスクを選択し、[Schedule FUOTA session] (FUOTA セッションのスケジュール) を選択します。
   2. FUOTA セッションの[Start date] (開始日) および[Start time] (開始時刻) を指定します。開始時刻が現在の時刻から 30 分以上後であることを確認してください。

API を使用して FUOTA タスクにデバイスを追加し FUOTA セッションをスケジュールする

AWS IoT Wireless API または CLI を使用して、ワイヤレスデバイスまたはマルチキャストグループを FUOTA タスクに追加できます。その後、FUOTA セッションをスケジュールできます。

1. デバイスおよびマルチキャストグループの追加

   ワイヤレスデバイスまたはマルチキャストグループを FUOTA タスクに関連付けることができます。

   - FUOTA タスクに個々のデバイスを関連付けるには、`AssociateWirelessDeviceWithFuotaTask` API オペレーションまたは

associate-wireless-device-with-fuota-task CLI コマンドを使用し、入力として WirelessDeviceID を指定します。

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

- マルチキャストグループを FUOTA タスクに関連付けるには、AssociateMulticastGroupWithFuotaTask API オペレーションまたは associate-multicast-group-with-fuota-task CLI コマンドを使用し、入力として MulticastGroupID を指定します。

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

ワイヤレスデバイスまたはマルチキャストグループを FUOTA タスクに関連付けた後、次の API オペレーションまたは CLI コマンドを使用して、デバイスまたはマルチキャストグループを一覧表示したり、タスクから関連付けを解除できます。

- DisassociateWirelessDeviceFromFuotaTask 、、または disassociate-wireless-device-from-fuota-task
- DisassociateMulticastGroupFromFuotaTask 、、または disassociate-multicast-group-from-fuota-task
- ListWirelessDevices 、、または list-wireless-devices
- ListMulticastGroups 、、または list-multicast-groups-by-fuota-task

> **Note**
>
> API:
>
> - MulticastGroupID では、ListWirelessDevices がフィルタとして使用される際に、一般的なワイヤレスデバイスおよびマルチキャストグループに関連付けられたデバイスを一覧表示できます。この API では、FuotaTaskID がフィルタとして使用される際に、FUOTA タスクに関連付けられているワイヤレスデバイスが一覧表示されます。

- `ListMulticastGroups` では、`FuotaTaskID` がフィルタとして使用される際に、一般的なマルチキャストグループおよび FUOTA タスクに関連付けられたマルチキャストグループを一覧表示できます。

2. FUOTA セッションのスケジュール

   デバイスまたはマルチキャストグループが FUOTA タスクに正常に追加されたら、FUOTA セッションを開始してデバイスのファームウェアを更新できます。開始時刻は、現在の時刻から 30 分以上後である必要があります。API または CLI を使用して FUOTA セッションをスケジュールするには、`StartFuotaTask` API オペレーションまたは `start-fuota-task` CLI コマンドを使用します。

   FUOTA セッションの開始後は、タスクにデバイスまたはマルチキャストグループを追加できなくなります。`GetFuotaTask` API オペレーションまたは `get-fuota-task` CLI コマンドを使用すると、FUOTA セッションのステータスに関する情報を得ることができます。

FUOTA タスクとタスクに追加されたデバイスのステータスを監視およびトラブルシューティングします。

ワイヤレスデバイスをプロビジョニングし、使用する可能性のあるマルチキャストグループを作成したら、次の手順を実行して FUOTA セッションを開始できます。

FUOTA タスクのステータス

FUOTA タスクには、AWS Management Console で次のステータスメッセージのいずれかが表示されます。

- 保留中

  このステータスは、FUOTA タスクを作成したが、まだファームウェアの更新セッションがないことを示します。タスクが作成されると、このステータスメッセージが表示されます。この間は、FUOTA タスクを更新できます。また、デバイスまたはマルチキャストグループをタスクと関連付けたり、関連付けを解除できます。ステータスが [Pending] (保留中) から変更された後は、タスクに他のデバイスを追加することはできません。

- FUOTA セッションの待機中

  デバイスが FUOTA タスクに正常に追加されると、タスクにファームウェアの更新セッションがスケジュールされている場合、このステータスメッセージが表示されます。この間、デバイスを更

新したり FUOTA セッションに追加することはできません。FUOTA セッションをキャンセルすると、グループのステータスは [Pending] (保留中) に変わります。

- FUOTA セッション中

  FUOTA セッションが開始されると、このステータスメッセージが表示されます。フラグメンテーションセッションが開始され、エンドデバイスによりフラグメントが収集され、ファームウェアイメージが再構築されます。また、新しいファームウェアバージョンが元のバージョンと比較され、新しいイメージが適用されます。

- FUOTA の完了

  新しいファームウェアイメージが適用されたことがエンドデバイスにより AWS IoT Core for LoRaWAN に報告された後、またはセッションのタイムアウト時に FUOTA セッションは完了としてマークされ、このステータスが表示されます。

  また、次のいずれかの場合もこのステータスが表示されるので、ファームウェアの更新がデバイスに正しく適用されたかどうかを確認してください。

  - FUOTA タスクのステータスが [FUOTA session waiting] (FUOTA セッションの待機中) で、S3 バケット内のイメージファイルへのリンクが間違っている、AWS IoT Core for LoRaWAN にバケット内のファイルにアクセスするための十分なアクセス許可がないなどの S3 バケットのエラーがある場合。
  - FUOTA タスクのステータスが [FUOTA session waiting] (FUOTA セッションの待機中) で、FUOTA セッションを開始するリクエストがあるが、FUOTA タスクのデバイスまたはマルチキャストグループからのレスポンスが受信されない場合。
  - FUOTA タスクのステータスが [In FUOTA session] (FUOTA セッション中) で、デバイスまたはマルチキャストグループが一定期間フラグメントを送信していないため、セッションがタイムアウトになっている場合。

- 削除の待機中

  他の状態にある FUOTA タスクを削除すると、このステータスが表示されます。削除は永続的で、元に戻すことができません。この操作には時間がかかる場合があり、FUOTA タスクが削除されるまでタスクのステータスは [Delete waiting] (削除の待機中) です。FUOTA タスクがこの状態に入った後は、他のステータスに移行することはできません。

FUOTA タスクでのデバイスのステータス

FUOTA タスクのデバイスには、AWS Management Console で次のステータスメッセージのいずれかが表示されます。各ステータスメッセージにカーソルを合わせると、そのメッセージが示す内容の詳細を確認できます。

- 初期

  FUOTA セッションの開始時、AWS IoT Core for LoRaWAN によりデバイスにファームウェアの更新でサポートされているパッケージがあるかどうかがが確認されます。デバイスにサポートされているパッケージがある場合、デバイスの FUOTA セッションが開始します。ファームウェアイメージがフラグメント化され、フラグメントがデバイスに送信されます。このステータスが表示された場合、デバイスの FUOTA セッションはまだ開始していません。

- パッケージはサポートされていません

  デバイスにサポートされている FUOTA パッケージがない場合は、このステータスが表示されます。ファームウェアの更新パッケージがサポートされていない場合、デバイスの FUOTA セッションを開始できません。このエラーを解決するには、FUOTA を使用してデバイスのファームウェアがファームウェアの更新を受信できるかどうかを確認します。

- フラグメンテーションアルゴリズムはサポートされていません

  FUOTA セッションの開始時に、AWS IoT Core for LoRaWAN によりデバイスのフラグメンテーションのセッションが設定されます。このステータスが表示された場合、使用するフラグメンテーションアルゴリズムの種類をデバイスのファームウェアの更新に適用できないことを意味します。このエラーは、使用するデバイスにサポートされている FUOTA パッケージがないために発生します。このエラーを解決するには、FUOTA を使用してデバイスのファームウェアがファームウェアの更新を受信できるかどうかを確認します。

- メモリが足りない

  AWS IoT Core for LoRaWAN によりイメージのフラグメントが送信された後、エンドデバイスによりそれらが収集されます。その後これらのフラグメントからバイナリイメージが再構築されます。このステータスは、デバイスにファームウェアイメージの受信フラグメントを組み合わせるのに十分なメモリがない場合に表示されます。そのために、ファームウェアの更新セッションが途中で終了する可能性があります。このエラーを解決するには、デバイスのハードウェアがこの更新を受信できるかどうかを確認します。使用しているデバイスでこの更新を受信できない場合は、デルタイメージを使用してファームウェアを更新します。

- フラグメンテーションのインデックスはサポートされていません

フラグメンテーションのインデックスは、同時に実行できる 4 つのフラグメンテーションのセッションのうち 1 つを識別します。デバイスが指定されたフラグメンテーションのインデックス値をサポートしていない場合、このステータスが表示されます。この問題を解決するには、次のいずれかの操作を実行します。

- デバイス用に新しい FUOTA タスクを開始します。
- エラーが解決しない場合は、ユニキャストモードからマルチキャストモードに切り替えます。
- それでもエラーが解決しない場合は、デバイスのファームウェアを確認してください。

- メモリエラー

  このステータスは、AWS IoT Core for LoRaWAN から受信フラグメントを受信した際に、デバイスにメモリエラーが発生したことを示します。このエラーが発生した場合、デバイスはこの更新プログラムを受信できない可能性があります。このエラーを解決するには、デバイスのハードウェアがこの更新を受信できるかどうかを確認します。必要に応じて、デルタイメージを使用してデバイスのファームウェアを更新します。

- 不適切な記述子

  デバイスが、指定された記述子をサポートしていません。記述子は、フラグメンテーションのセッション中に転送されるファイルを記述するフィールドです。このエラーが表示された場合は、AWS Support Center (センター) にお問い合わせください。

- セッションカウントのリプレイ

  このステータスは、デバイスがこのセッションカウントを以前に使用したことを示します。このエラーを解決するには、デバイス用に新しい FUOTA タスクを開始します。

- フラグメントが見つかりません

  デバイスにより AWS IoT Core for LoRaWAN からイメージのフラグメントが収集される際、コード化され独立したフラグメントから新しいファームウェアイメージが再構築されます。デバイスがすべてのフラグメントを受信していない場合、新しいイメージを再構築できず、このステータスが表示されます。このエラーを解決するには、デバイス用に新しい FUOTA タスクを開始します。

- MIC エラー

  収集されたフラグメントから新しいファームウェアイメージがデバイスにより再構築されると、MIC (Message Integrity Check) が実行され、イメージの信頼性と、それが正しいソースから取得されているかどうかが確認されます。フラグメントの再構成後、デバイスにより MIC の不一致が検出されると、このステータスが表示されます。このエラーを解決するには、デバイス用に新しい FUOTA タスクを開始します。

- Successful

  デバイスの FUOTA セッションが成功しました。

  > **Note**
  >
  > このステータスメッセージは、デバイスによりフラグメントからイメージが再構築され、それが確認できたことを示します。しかし、デバイスにより AWS IoT Core for LoRaWAN にステータスが報告されたときに、デバイスファームウェアが更新されていない可能性があります。デバイスのファームウェアが更新されているかどうかを確認してください。

次のステップ

FUOTA タスクとそのデバイスのさまざまなステータス、および問題のトラブルシューティングの方法について学習しました。これらの各ステータスの詳細については、「LoRaWAN Fragmented Data Block Transportation Specification, TS004-1.0.0」を参照してください。

ネットワークアナライザを使用したワイヤレスリソースフリートのリアルタイムでのモニタリング

ネットワークアナライザは、デフォルトの WebSocket 接続を使用して、ワイヤレス接続リソースのリアルタイムトレースメッセージログを受信します。ネットワークアナライザを使用すると、モニタリングするリソースを追加し、トレースメッセージングセッションをアクティブ化し、リアルタイムでトレースメッセージの受信を開始できます。

リソースをモニタリングするには、Amazon CloudWatch を使用することもできます。CloudWatch を使用するには、IAM ロールをセットアップしてログ記録を設定し、ログエントリがコンソールに表示されるのを待ちます。ネットワークアナライザを使用すると、接続のセットアップとトレースメッセージの受信開始にかかる時間が大幅に短縮され、リソースのフリートに関するジャストインタイムのログ情報を取得できます。CloudWatch を使用したモニタリングについては、「Amazon CloudWatch Logs を使用した AWS IoT Wireless リソースのモニタリング」を参照してください。

セットアップ時間を短縮し、トレースメッセージの情報を使用することで、リソースをより効果的にモニタリングし、有意義な洞察を得て、エラーのトラブルシューティングを行うことができます。LoRaWAN デバイスと LoRaWAN ゲートウェイの両方をモニタリングできます。例えば、LoRaWAN デバイスの 1 つのオンボーディング時に、参加エラーをすばやく特定できます。エラーをデバッグするには、提供されたトレースメッセージログの情報を使用します。

ネットワークアナライザの使用方法

リソースフリートをモニタリングし、トレースメッセージの受信を開始するには、次の手順を実行します。

1. ネットワークアナライザの設定を作成し、リソースを追加する

   トレースメッセージングをアクティブ化する前に、ネットワークアナライザ設定を作成し、設定にリソースを追加します。まず、ログレベルとワイヤレスデバイスのフレーム情報を含む設定を指定します。次に、ワイヤレスゲートウェイとワイヤレスデバイス ID を使用して、モニタリングするワイヤレスリソースを追加します。

2. WebSockets を使用してトレースメッセージをストリーミングする

   IAM ロールの認証情報を使用して署名済みのリクエスト URL を生成し、WebSocket プロトコルを使用してネットワークアナライザートレースメッセージをストリーミングできます。

3. トレースメッセージングセッションをアクティブ化し、トレースメッセージをモニタリングする

   トレースメッセージの受信を開始するには、トレースメッセージングセッションをアクティブにします。追加コストが発生しないようにするには、ネットワークアナライザのトレースメッセージングセッションを非アクティブ化するか、閉じます。

次の動画では、AWS IoT Core for LoRaWAN ネットワークアナライザの仕組みについて説明し、ネットワークアナライザを使用してリソースを追加し、結合アクティビティをトレースするプロセスについて説明します。

次のトピックでは、設定を作成し、リソースを追加し、トレースメッセージングセッションをアクティブ化する方法を示します。

トピック

- ネットワークアナライザに必要な IAM ロールを追加する
- ネットワークアナライザの設定を作成し、リソースを追加する
- WebSockets を使用してネットワークアナライザのトレースメッセージをストリーミングする
- ネットワークアナライザのトレースメッセージログをリアルタイムで表示およびモニタリングする
- ネットワークアナライザを使用してマルチキャストグループと FUOTA タスクのデバッグとトラブルシューティングを行う

ネットワークアナライザに必要な IAM ロールを追加する

ネットワークアナライザを使用する場合は、API オペレーションを使用する権限をユーザーに付与する必要があります。UpdateNetworkAnalyzerConfiguration そして GetNetworkAnalyzerConfiguration をクリックして、ネットワークアナライザリソースにアクセスします。以下に、権限を付与するために使用する IAM ポリシーを示します。

ネットワークアナライザの IAM ポリシー

以下のいずれかを実行してみてください。

- フルアクセスワイヤレスポリシー

  ロールにポリシー AWSIoTWirelessFullAccess をアタッチして、AWS IoT Core for LoRaWAN にフルアクセスポリシーを付与します。詳細については、「AWSIoTWirelessFullAccess ポリシーの概要」を参照してください。

- API を取得および更新するためのスコープ付き IAM ポリシー

  IAM コンソールの [Create policy] (ポリシーの作成) ページで、[Visual editor] (ビジュアルエディタ) タブを開き、以下の IAM ポリシーを作成します。

  1. [Service] (サービス) には [IoTWireless] を選択します。
  2. [Access level] (アクセスレベル) で [Read] (読み取り) を展開して [GetNetworkAnalyzerConfiguration] を選択し、[Wrtite] (書き込み) を展開して [UpdateNetworkAnalyzerConfiguration] を選択してください。
  3. [Next:Tags] (次へ: タグ) を選択し、[IoTWirelsesNetworkAnalyzerPolicy] など、ポリシーの [Name] (名前) を入力します。[Create policy] (ポリシーの作成) を選択します。

  以下は、ユーザーが作成したポリシー [IoTWirelessNetworkAnalyzerPolicy] を示しています。ポリシー作成の詳細については、「IAM ポリシーの作成」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

特定のリソースにアクセスするためのスコープ付きポリシー

よりきめ細かなアクセスコントロールを設定するには、ワイヤレスゲートウェイとデバイスを [Resource] (リソース) フィールドに追加する必要があります。次のポリシーでは、ワイルドカード ARN を使用して、すべてのゲートウェイとデバイスへのアクセスを許可します。`WirelessGatewayId` および `WirelessDeviceId` を使用して、特定のゲートウェイおよびデバイスへのアクセスを制御できます。

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

ネットワークアナライザを使用するものの、ワイヤレスゲートウェイまたはデバイスを使用しない権限をユーザーに付与するには、次のポリシーを使用します。指定しない限り、リソースを使用する権限は暗黙的に拒否されます。

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

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

次のステップ

ポリシーを作成したので、ネットワークアナライザの設定にリソースを追加し、それらのリソースのトレースメッセージング情報を受け取ることができるようになりました。詳細については、「ネットワークアナライザの設定を作成し、リソースを追加する」を参照してください。

ネットワークアナライザの設定を作成し、リソースを追加する

トレースメッセージをストリーミングする前に、ネットワークアナライザ設定を作成し、モニタリングするリソースをこの設定に追加します。設定を作成すると、次のことができます。

- 設定名とオプションの説明を指定します。
- ログメッセージのフレーム情報や詳細レベルなどの設定をカスタマイズします。
- モニタリングするリソースを追加します。リソースは、ワイヤレスデバイスまたはワイヤレスゲートウェイ、もしくはその両方にすることができます。

指定した構成設定によって、設定に追加するリソースについて受信するトレースメッセージング情報が決まります。モニタリングするユースケースに応じて、複数の設定を作成することもできます。

以下は、設定を作成してリソースを追加する方法を示しています。

トピック

- ネットワークアナライザの設定を作成する
- リソースを追加し、ネットワークアナライザの設定を更新する

ネットワークアナライザの設定を作成する

ワイヤレスゲートウェイまたはワイヤレスデバイスをモニタリングする前に、ネットワークアナライザ設定を作成する必要があります。設定を作成するときは、設定名を指定するだけです。作成した後でも、設定をカスタマイズし、モニタリングするリソースを設定に追加できます。設定によって、それらのリソースについて受信するトレースメッセージング情報が決まります。

モニタリングするリソースとそれらに対して受信する情報のレベルに応じて、複数の設定を作成できます。例えば、AWS アカウント の一連のゲートウェイについてのエラー情報のみを表示する設定を作成できます。モニタリングするワイヤレスデバイスに関するすべての情報を表示する設定を作成することもできます。

次のセクションでは、さまざまな設定と、設定の作成方法について説明します。

構成設定

ネットワークアナライザ設定を作成または更新するときに、次のパラメータをカスタマイズしてログストリーム情報をフィルタリングすることもできます。

- フレーム情報

  この設定は、トレースメッセージについてのワイヤレスデバイスリソースのフレーム情報です。フレーム情報は、ネットワークサーバーとエンドデバイス間の通信のデバッグに使用できます。このエージェントは、デフォルトでは有効になっています。

- ログレベル

  情報ログまたはエラーログを表示したり、ログ記録をオフにしたりできます。

  - 情報

    ログレベルが [Info] (情報) のログは冗長性が高く、エラーログストリームと情報ログストリームの両方を含みます。情報ログは、デバイスまたはゲートウェイの状態の変更を表示するために使用できます。

    > **Note**
    >
    > 冗長性が高いログストリームを収集すると、追加コストが発生する可能性があります。料金の詳細については、「AWS IoT Core の料金」を参照してください。

  - エラー

ログレベルが Error のログは冗長性が低く、エラー情報のみを表示します。これらのログは、アプリケーションにデバイス接続エラーなどのエラーがあるときに使用できます。ログストリームの情報を使用して、フリート内のリソースのエラーを特定し、トラブルシューティングできます。

コンソールを使用して設定を作成する

AWS IoT コンソールまたは AWS IoT Wireless API を使用して、ネットワークアナライザ設定を作成し、オプションパラメータをカスタマイズできます。また、複数の設定を作成し、後で使用しなくなった設定を削除することもできます。

ネットワークアナライザの設定を作成する

1. AWS IoT コンソールのネットワークアナライザハブを開いて、[Create configuration] (設定の作成) を選択します。
2. 設定を指定します。
   - 名前、説明、タグ

     英字、数字、ハイフン、またはアンダースコアのみを含む、一意の [Configuration name] (設定名) を指定します。オプションの [Description] (説明) フィールドを使用して設定に関する情報を指定し、[Tags] (タグ) フィールドを使用して設定に関するメタデータのキーバリューペアを追加します。リソースの命名と説明の詳細については、AWS IoT Wireless リソースについて説明する を参照してください。
   - 構成設定

     フレーム情報を無効にするかどうかを選択し、[Select log levels] (ログレベルの選択) を使用して、トレースメッセージログに使用するログレベルを選択します。[Next] を選択します。
3. 設定にリソースを追加します。今すぐリソースを追加するか、[Create] (作成) を選択して、後でリソースを追加することもできます。後でリソースを追加するには、[Create] (作成) を選択します。

   [Network Analyzer hub page] (ネットワークアナライザハブページ) には、作成した設定とその構成が表示されます。新しい設定の詳細を表示するには、設定名を選択します。

ネットワークアナライザの設定を削除する

モニタリングするリソースと、そのリソースに対して受け取るトレースメッセージング情報のレベルに応じて、複数のネットワークアナライザ設定を作成することができます。

コンソールから設定を削除するには

1. AWS IoT コンソールのネットワークアナライザハブに移動し、削除する設定を選択します。
2. [アクション] を選択し、[削除] を選択します。

API を使用して設定を作成する

API を使用してネットワークアナライザーの設定を作成するには、CreateNetworkAnalyzerConfiguration API オペレーションまたは create-network-analyzer-configuration CLI コマンドを使用します。

設定を作成するときは、設定名を指定するだけです。この API オペレーションを使用して、構成設定を指定し、設定の作成時にリソースを追加することもできます。または、UpdateNetworkAnalyzerConfiguration API オペレーションまたは update-network-analyzer-configuration CLI を使用して、後で指定することもできます。

- 設定を作成する

  設定の作成時に、名前を指定する必要があります。例えば、次のコマンドでは、名前とオプションの説明のみを指定して、設定を作成します。デフォルトでは、フレーム情報がアクティブ化されており、ログレベルは INFO を使用する設定になっています。

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_Network_Analyzer_Config \
    --description "My first network analyzer configuration"
```

  このコマンドを実行すると、ネットワークアナライザ設定の ARN と ID が表示されます。

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

- リソースを使用して設定を作成する

構成設定をカスタマイズするには、trace-content パラメータを使用します。リソースを追加するには、WirelessDevices および WirelessGateways パラメータを使用して、設定に追加するゲートウェイ、デバイス、またはその両方を指定します。例えば、次のコマンドは、構成設定をカスタマイズし、WirelessGatewayID と WirelessDeviceID で指定されたワイヤレスリソースを設定に追加します。

```
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"
```

次の例は、コマンドを実行したときの出力を示しています。

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

ネットワークアナライザ設定を一覧表示する

モニタリングするリソースと、そのリソースに対して受け取るトレースメッセージング情報の詳細のレベルに応じて、複数のネットワークアナライザ設定を作成することができます。これらの設定を作成した後、ListNetworkAnalyzerConfigurations API オペレーションまたは list-network-analyzer-configuration CLI コマンドを使用して、これらの設定の一覧を取得できます。

```
aws iotwireless list-network-analyzer-configurations
```

このコマンドを実行すると、AWS アカウント のすべてのネットワークアナライザ設定が表示されます。また、max-results パラメータを使用して、表示する設定の数を指定することができます。以下は、このコマンドを実行したときの出力を示しています。

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

ネットワークアナライザの設定を削除する

DeleteNetworkAnalyzerConfiguration API オペレーションまたは delete-network-analyzer-configuration CLI コマンドを使用して、使用しなくなった設定を削除できます。

```
aws iotwireless delete-network-analyzer-configuration \
    --configuration-name My_NetworkAnalyzer_Config
```

このコマンドを実行しても、出力は生成されません。使用可能な設定を確認するには、ListNetworkAnalyzerConfigurations API オペレーションを使用できます。

次のステップ

ネットワークアナライザ設定を作成したので、設定にリソースを追加したり、構成設定を更新したりできます。詳細については、「リソースを追加し、ネットワークアナライザの設定を更新する」を参照してください。

リソースを追加し、ネットワークアナライザの設定を更新する

トレースメッセージングをアクティブ化する前に、設定にリソースを追加する必要があります。使用できるデフォルトのネットワークアナライザ設定は 1 つだけです。AWS IoT Core for LoRaWAN によって、NetworkAnalyzerConfig_Default という名前がこの設定に割り当てられます。このフィールドは編集できません。この設定は、コンソールからネットワークアナライザを使用する場合、自動的に AWS アカウント に追加されます。

このデフォルト設定に、モニタリングするリソースを追加できます。リソースは、LoRaWAN デバイスと LoRaWAN ゲートウェイのいずれかまたは両方にすることができます。設定に個々のリソースを追加するには、ワイヤレスゲートウェイとワイヤレスデバイス ID を使用します。

設定

設定をするには、まずデフォルト設定にリソースを追加し、トレースメッセージングをアクティブにします。トレースメッセージログを受信したら、次のパラメータをカスタマイズしてデフォルト設定を更新し、ログストリームをフィルタリングすることもできます。

- フレーム情報

  この設定は、トレースメッセージのワイヤレスデバイスリソースのフレーム情報です。フレーム情報はデフォルトで有効になっており、ネットワークサーバーとエンドデバイス間の通信のデバッグに使用できます。

- ログレベル

  情報ログまたはエラーログを表示したり、ログ記録をオフにしたりできます。

  - 情報

    ログレベルが Info のログは冗長性が高く、情報を提供しエラーを含むログストリームを含んでいます。情報ログは、デバイスまたはゲートウェイの状態の変更を表示するために使用できます。

    > **Note**
    >
    > 冗長性が高いログストリームを収集すると、追加コストが発生する可能性があります。料金の詳細については、「AWS IoT Core の料金」を参照してください。

  - エラー

    ログレベルが Error のログは冗長性が低く、エラー情報のみを表示します。これらのログは、アプリケーションにデバイス接続エラーなどのエラーがあるときに使用できます。ログストリームの情報を使用して、フリート内のリソースのエラーを特定し、トラブルシューティングできます。

前提条件

リソースを追加する前に、モニタリングするゲートウェイとデバイスを AWS IoT Core for LoRaWAN にオンボーディングしておく必要があります。詳細については、「AWS IoT Core for LoRaWAN へのゲートウェイとデバイスの接続」を参照してください。

コンソールを使用して、リソースを追加し、ネットワークアナライザ設定を更新する

リソースを追加し、オプションパラメータをカスタマイズするには、AWS IoT コンソールまたは AWS IoT Wireless API を使用します。リソースに加えて、設定を編集し、更新された設定を保存することもできます。

設定にリソースを追加するには (コンソール)

1. AWS IoT コンソールの [Network Analyzer] (ネットワークアナライザ) ハブを開き、ネットワークアナライザの設定 NetworkAnalyzerConfig_Default を選択します。
2. [Add resources] (リソースを追加) を選択します。
3. ワイヤレスゲートウェイとワイヤレスデバイス ID を使用して、モニタリングするリソースを追加します。最大 250 のワイヤレスゲートウェイまたはワイヤレスデバイスを追加できます。リソースを追加するには、次の手順に従います。
   a. [View gateways] (ゲートウェイを表示) または [View devices] (デバイスを表示) タブを使用して、AWS アカウント に追加したゲートウェイとデバイスのリストを表示します。
   b. モニタリングするデバイスまたはゲートウェイの `WirelessDeviceID` または `WirelessGatewayID` をコピーし、対応するリソースの識別子の値を入力します。
   c. リソースの追加を続行するには、[Add gateway] (ゲートウェイを追加) または [Add device] (デバイスを追加) を選択し、ワイヤレスゲートウェイまたはデバイスを追加します。追加していたリソースをモニタリングする必要がなくなった場合は、[Remove resource] (リソースを削除) を選択します。
4. すべてのリソースを追加したら、[Add] (追加) をクリックします。

   追加したゲートウェイとデバイスの数が[Network Analyzer] (ネットワークアナライザ) ハブページに表示されます。トレースメッセージングセッションをアクティブにするまで、ゲートウェイとデバイスをさらに続けて追加できます。セッションをアクティブ化した後、リソースを追加するには、セッションを非アクティブ化する必要があります。

ネットワークアナライザの設定を編集するには (コンソール)

ネットワークアナライザの設定を編集して、トレースメッセージログのフレーム情報を無効にするかどうかとそのログレベルを選択することもできます。

1. AWS IoT コンソールの [Network Analyzer] (ネットワークアナライザ) ハブを開き、ネットワークアナライザの設定 NetworkAnalyzerConfig_Default を選択します。
2. [編集] を選択します。

3. フレーム情報を無効にするかどうかを選択し、[Select log levels](ログレベルの選択) を使用して、トレースメッセージログに使用するログレベルを選択します。[Save] を選択します。

   ネットワークアナライザの設定の詳細ページで指定した設定が表示されます。

API を使用してリソースを追加し、ネットワークアナライザの設定を更新する

AWS IoT Wireless API オペレーションまたはAWS IoT Wireless CLI コマンドを使用して、リソースを追加し、ネットワークアナライザの設定を更新します。

- リソースを追加し、ネットワークアナライザの設定を更新するには、UpdateNetworkAnalyzerConfiguration API または update-network-analyzer-configuration CLI を使用します。
  - リソースを追加する

    追加するワイヤレスデバイスについては、`WirelessDevicesToAdd` を使用してデバイスの `WirelessDeviceID` を文字列の配列として入力します。追加するワイヤレスゲートウェイについては、`WirelessGatewaysToAdd` を使用してゲートウェイの `WirelessGatewayID` を文字列の配列として入力します。
  - 設定を編集する

    ネットワークアナライザの設定を編集するには、`TraceContent` パラメータを使用して、`WirelessDeviceFrameInfo` を `ENABLED` にするか `DISABLED` にするか、および `LogLevel` パラメータを `INFO`、`ERROR`、`DISABLED` のいずれにするかを指定します。

```
{
   "TraceContent": {
      "LogLevel": "string",
      "WirelessDeviceFrameInfo": "string"
   },
   "WirelessDevicesToAdd": [ "string" ],
   "WirelessDevicesToRemove": [ "string" ],
   "WirelessGatewaysToAdd": [ "string" ],
   "WirelessGatewaysToRemove": [ "string" ]
}
```

- 追加した設定とリソースに関する情報を取得するには、GetNetworkAnalyzerConfiguration API オペレーション、または get-network-analyzer-configuration コマンドを使用します。ネットワークアナライザ設定 `NetworkAnalyzerConfig_Default` の名前を入力として指定します。

次のステップ

リソースを追加し、オプションの設定を指定したので、WebSocket プロトコルを使用して、AWS IoT Core for LoRaWAN との接続を確立し、ネットワークアナライザを使用できるようになります。その後、トレースメッセージをアクティブにして、リソースのトレースメッセージの受信を開始できます。詳細については、「WebSockets を使用してネットワークアナライザのトレースメッセージをストリーミングする」を参照してください。

WebSockets を使用してネットワークアナライザのトレースメッセージをストリーミングする

WebSocket プロトコルを使用すると、ネットワークアナライザトレースメッセージをリアルタイムでストリーミングできます。リクエストを送信すると、サービスは JSON ストラクチャで応答します。トレースメッセージングをアクティブ化したら、メッセージログを使用してリソースに関する情報を取得し、エラーのトラブルシューティングを行うことができます。詳細については、「WebSocket protocol」を参照してください。

WebSockets を使用してネットワークアナライザのトレースメッセージをストリーミングする方法を次に示します。

トピック

- WebSocket ライブラリで署名済みリクエストを生成する
- WebSocket メッセージとステータスコード

WebSocket ライブラリで署名済みリクエストを生成する

WebSocket ライブラリを使用してサービスにリクエストを送信できるように、署名済みリクエストを生成する方法を次に示します。

IAM ロールに WebSocket リクエストのポリシーを追加する

WebSocket プロトコルを使用してネットワークアナライザを呼び出すには、リクエストを行う AWS Identity and Access Management (IAM) ロールに以下のポリシーをアタッチする必要があります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotwireless:StartNetworkAnalyzerStream",
```

```
            "Resource": "*"
        }
    ]
}
```

署名付き URL を作成する

アプリケーションとネットワークアナライザとの間の通信を設定するために必要な情報を含む、WebSocket リクエストの URL を作成します。リクエストのアイデンティティを確認するために、WebSocket ストリーミングでは、署名のリクエストに Amazon 署名バージョン 4 のプロセスが使用されます。署名バージョン 4 の詳細については、Amazon Web Services 全般のリファレンスの「AWS API リクエストの署名」を参照してください。

ネットワークアナライザを呼び出すには、StartNetworkAnalyzerStream リクエスト URL を使用します。リクエストは、前述の IAM ロールの認証情報を使用して署名されます。URL は以下の形式です。読みやすさのために改行が追加されています。

```
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
```

署名バージョン 4 パラメータには、次の値を使用します。

- X-Amz-Algorithm – 署名プロセスに使用しているアルゴリズム。唯一の有効な値は AWS4-HMAC-SHA256 です。
- X-Amz-Credential – アクセスキー ID と認証情報スコープのコンポーネントを連結して形成され、スラッシュ (/) で区切られる文字列。認証情報スコープには、YYYYMMDD 形式の日付、AWS リージョン、サービス名、および終了文字列 (aws4_request) が含まれます。
- X-Amz-Date – 署名が作成された日時。Amazon Web Services 全般リファレンスの「署名バージョン 4 の日付の処理」の手順に従って、日付と時刻を生成します。
- X-Amz-Expires – 認証情報が有効期限切れになるまでの時間 (秒単位)。最大値は 300 秒 (5 分) です。
- X-Amz-Security-Token – (オプション) 一時的な認証情報用の署名バージョン 4 のトークン。このパラメータを指定する場合は、正規リクエストに含めます。詳細については、AWS Identity and

Access Management ユーザーガイドの「一時的なセキュリティ認証情報のリクエスト」を参照してください。

- X-Amz-Signature – リクエストに対して生成した署名バージョン 4 の署名。
- X-Amz-SignedHeaders – リクエストの署名を作成するときに署名されるヘッダー。唯一の有効な値は host です。

リクエスト URL を構築し、署名バージョン 4 署名を作成する

リクエストの URL を作成し、署名バージョン 4 の署名を作成するには、次のステップを実行します。例は擬似コードで示しています。

タスク 1: 正規リクエストを作成する

リクエストからの情報を含む文字列を標準化された形式で作成します。これにより、AWS は、リクエストを受け取ると、タスク 3: 署名を計算する でお客様が計算したものと同じ署名を計算できるようになります。詳細については、Amazon Web Services 全般のリファレンスの「署名バージョン 4 の正規リクエストを作成する」を参照してください。

1. アプリケーションでリクエストの変数を定義します。

```
# 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. 正規 URI (ユニフォームリソース識別子) を作成します。正規 URI はドメインとクエリ文字列との間の URI の一部です。

```
canonical_uri = "/start-network-analyzer-stream"
```

3. 正規ヘッダーと署名付きヘッダーを作成します。正規ヘッダーの末尾の \n に注意してください。

   - 小文字のヘッダー名とそれに続くコロンを追加します。
   - このヘッダーの値のカンマ区切りリストを追加します。複数の値を持つヘッダーの値はソートしないようにします。
   - 改行 (\n) を追加します。

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. アルゴリズムをハッシュアルゴリズムに一致させます。SHA-256 を使用する必要があります。

```
algorithm = "AWS4-HMAC-SHA256"
```

5. 派生キーの範囲をリクエストが実行された日付、 リージョン、サービスに絞るための認証情報スコープを作成します。

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. 正規クエリ文字列を作成します。クエリ文字列値は、URI エンコードされ、名前順にソートされている必要があります。

   - パラメータ名を文字コードポイントで昇順にソートします。名前が重複しているパラメータは、値でソートする必要があります。たとえば、大文字 F で始まるパラメータ名は、小文字 b で始まるパラメータ名より前に置きます。
   - RFC 3986 が定義する非予約文字である A-Z、a-z、0-9、ハイフン (-)、アンダースコア (_)、ピリオド (.)、およびチルド (~) を、URI でエンコードしないようにします。
   - 他のすべての文字についても、%XY によるパーセントエンコードが必要です。X および Y には 16 進数文字 (0～9 および大文字の A ～ F) が入ります。例えば、スペース文字は %20 として (一部のエンコードスキームのように '+' を使用せずに) エンコードする必要があり、拡張 UTF-8 文字は %XY%ZA%BC 形式でエンコードする必要があります。
   - パラメータ値の等号 (=) 文字を二重エンコードします。

```
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. ペイロードのハッシュを作成します。GET リクエストの場合、ペイロードは空の文字列です。

```
payload_hash = HashSHA256(("").Encode("utf-8")).HexDigest()
```

8. すべての要素を組み合わせて正規リクエストを作成します。

```
canonical_request = method + '\n'
   + canonical_uri + '\n'
   + canonical_querystring + '\n'
   + canonical_headers + '\n'
   + signed_headers + '\n'
   + payload_hash
```

タスク 2: 署名する文字列を作成します。

署名する文字列には、リクエストについてのメタ情報が含まれています。次のステップでリクエストの署名を計算するときに、文字列を使用してサインインします。詳細については、Amazon Web Services 全般のリファレンスの「署名バージョン 4 の署名文字列を作成する」を参照してください。

```
string_to_sign=algorithm + "\n"
   + amz_date + "\n"
   + credential_scope + "\n"
   + HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

タスク 3: 署名を計算する

AWS シークレットアクセスキーから署名キーを取得します。保護レベルを高めるために、派生キーは日付、サービス、AWS リージョンに固有になっています。取得したキーを使用して、リクエストに署名します。詳細については、Amazon Web Services 全般のリファレンスの「AWS 署名バージョン 4 の署名を計算する」を参照してください。

コードでは、GetSignatureKey 関数を実装して署名キーを取得したものとします。詳細と関数の例については、「Amazon Web Services 全般リファレンス」の「署名バージョン 4 の署名キーを取得する方法の例」を参照してください。

関数 HMAC(key, data) は、バイナリ形式で結果を返す HMAC-SHA 256 関数を表します。

```
#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
```

タスク 4: 署名情報をリクエストに追加し、リクエスト URL を作成する

署名を計算したら、クエリ文字列に追加します。詳細については、Amazon Web Services 全般リファレンスの「HTTP リクエストに署名を追加する」を参照してください。

```
#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
```

次のステップ

WebSocket ライブラリでリクエスト URL を使用して、サービスへのリクエストを実行し、メッセージを確認できるようになりました。詳細については、「WebSocket メッセージとステータスコード」を参照してください。

WebSocket メッセージとステータスコード

署名済みリクエストを作成したら、WebSocket ライブラリ、またはプログラミング言語に適したライブラリでリクエスト URL を使用して、サービスへのリクエストを行うことができます。この署名付きリクエストの生成方法の詳細については、「WebSocket ライブラリで署名済みリクエストを生成する」を参照してください。

WebSocket メッセージ

WebSocket プロトコルを使用して双方向接続を確立できます。メッセージは、クライアントからサーバーに、およびサーバーからクライアントに送信できます。ただし、ネットワークアナライザは、サーバーからクライアントに送信されるメッセージのみをサポートします。クライアントからの

メッセージの受信は想定外であり、クライアントからメッセージを受信すると、サーバーは自動的に WebSocket 接続を閉じます。

リクエストが受信され、トレースメッセージングセッションが開始されると、サーバーはペイロードである JSON ストラクチャで応答します。ペイロードの詳細と、AWS Management Console からトレースメッセージングをアクティブ化する方法については、「ネットワークアナライザのトレースメッセージログをリアルタイムで表示およびモニタリングする」を参照してください。

WebSocket ステータスコード

サーバーからクライアントへの通信の WebSocket ステータスコードを次に示します。WebSocket ステータスコードは、「RFC Standard of Normal closure of connections」に従っています。

サポートされているステータスコードを次に示します。

- 1000

  このステータスコードは通常のクロージャを示します。これは WebSocket 接続が確立され、リクエストへの対応が実行されたことを意味します。このステータスは、セッションがアイドル状態になり、接続がタイムアウトになったときに確認できます。
- 1002

  このステータスコードは、プロトコルエラーが原因でエンドポイントが接続を終了していることを示します。
- 1003

  このステータスコードは、受信できない形式のデータを受信したため、エンドポイントが接続を終了したエラーステータスを示します。エンドポイントはテキストデータのみをサポートしており、サポートされていない形式を使用しているクライアントからバイナリメッセージまたはメッセージを受信すると、このステータスコードが表示されることがあります。
- 1008

  このステータスコードは、ポリシーに違反するメッセージを受信したため、エンドポイントが接続を終了したエラーステータスを示します。このステータスは汎用であり、1003 や 1009 などの他のステータスコードが適用されない場合に表示されます。このステータスは、ポリシーを非表示にする必要がある場合や、有効期限が切れた署名などの認証に失敗した場合にも表示されます。
- 1011

  このステータスコードは、予期しない状態または内部エラーが発生して、リクエストへの対応を実行できなかったため、サーバーが接続を終了しているエラーステータスを示します。

次のステップ

署名済みリクエストを生成する方法と、WebSocket 接続を使用してサーバーからのメッセージを確認する方法を学んだので、トレースメッセージングをアクティブにして、ワイヤレスゲートウェイとワイヤレスデバイスリソースのメッセージログの受信を開始できます。詳細については、「ネットワークアナライザのトレースメッセージログをリアルタイムで表示およびモニタリングする」を参照してください。

ネットワークアナライザのトレースメッセージログをリアルタイムで表示およびモニタリングする

ネットワークアナライザの設定にリソースを追加した場合は、トレースメッセージングをアクティブにして、リソースのトレースメッセージの受信を開始できます。AWS Management Console、AWS IoT Wireless API、AWS CLI のいずれかを使用できます。

前提条件

ネットワークアナライザを使用してトレースメッセージングをアクティブ化する前に、次の準備が必要です。

- デフォルトのネットワークアナライザ設定に、モニタリングするリソースが追加されている。詳細については、「リソースを追加し、ネットワークアナライザの設定を更新する」を参照してください。
- `StartNetworkAnalyzerStream` リクエスト URL を使用して、署名済みリクエストが生成されている。リクエストは、このリクエストを行う AWS Identity and Access Management ロールの認証情報を使用して署名されます。詳細については、「署名付き URL を作成する」を参照してください。

コンソールを使用してトレースメッセージングをアクティブ化する

トレースメッセージングをアクティブにするには

1. AWS IoT コンソールの [Network Analyzer] (ネットワークアナライザ) ハブを開き、ネットワークアナライザの設定 NetworkAnalyzerConfig_Default を選択します。
2. ネットワークアナライザの設定の詳細ページで、[Activate trace messaging] (トレースメッセージングをアクティブ化) を選択し、[Activate] (アクティブ化) を選択します。

   トレースメッセージの受信が開始され、最新のトレースメッセージからコンソールに表示されます。

> **Note**
>
> メッセージングセッションの開始後、トレースメッセージの受信には、セッションを非アクティブ化するか、トレースセッションを終了するまで、追加コストが発生する可能性があります。料金の詳細については、「AWS IoT Core の料金」を参照してください。

トレースメッセージの表示とモニタリング

トレースメッセージングをアクティブにすると、WebSocket 接続が確立され、トレースメッセージが最新のものからリアルタイムで表示されます。プリファレンスをカスタマイズして、各ページに表示するトレースメッセージの数を指定し、各メッセージの関連するフィールドのみを表示させることができます。例えば、トレースメッセージログをカスタマイズして、ログレベルが `ERROR` に設定されているワイヤレスゲートウェイリソースのログのみを表示することができます。そうすると、ゲートウェイでエラーをすばやく特定してデバッグできます。トレースメッセージには、次に示す情報が含まれます。

- メッセージ番号: 最後に受信したメッセージが最初になるように付番された一意の番号。
- リソース ID: リソースのワイヤレスゲートウェイまたはワイヤレスデバイス ID。
- タイムスタンプ: メッセージが受信された時刻。
- メッセージ ID: AWS IoT Core for LoRaWAN が受信した各メッセージに割り当てた識別子。
- FPort: WebSocket 接続を使用してデバイスと通信するための周波数ポート。
- DevEui: ワイヤレスデバイスの拡張一意識別子 (EUI)。
- リソース: モニタリング対象リソースがワイヤレスデバイスかワイヤレスゲートウェイか。
- イベント: ワイヤレスデバイスのログメッセージのイベント。Join、Rejoin、Uplink_Data、Downlink_Data、または Registration。
- ログレベル: デバイスの `INFO` または `ERROR` ログストリームに関する情報。

ネットワークアナライザ JSON ログメッセージ

一度に 1 つのトレースメッセージを選択して、そのメッセージの JSON ペイロードを表示することもできます。トレースメッセージログで選択したメッセージに応じて、2 つの部分 (CustomerLog と LoRaFrame) が含まれていることを示す情報が JSON ペイロードに表示されます。

CustomerLog

JSON の CustomerLog 部分には、メッセージを受信したリソースのタイプと識別子、ログレベル、およびメッセージの内容が表示されます。次の例は、CustomerLog ログメッセージの例を示しています。JSON の `message` フィールドを使用して、エラーとその解決方法に関する詳細情報を取得できます。

LoRaFrame

JSON の LoRaFrame 部分には、メッセージ ID があり、デバイスの物理ペイロードとワイヤレスメタデータに関する情報が含まれています。

次の例に、トレースメッセージの構造を示します。

```
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;
  },
};
```

レビューと以降のステップ

このセクションでは、トレースメッセージを表示し、この情報を使用してエラーをデバッグする方法を学習しました。すべてのメッセージを表示したら、次の操作を実行できます。

- トレースメッセージングを無効化する

  追加コストが発生しないようにするには、トレースメッセージングセッションを非アクティブにします。セッションを非アクティブにすると、WebSocket 接続が切断され、追加のトレースメッセージが受信されなくなります。コンソールで既存のメッセージを引き続き表示することができます。

- 設定のフレーム情報を編集する

  ネットワークアナライザの設定を編集して、フレーム情報を非アクティブ化するかどうかを選択し、メッセージのログレベルを選択できます。設定を更新する前に、トレースメッセージングセッションを非アクティブ化することを検討してください。これらの編集を行うには、AWS IoTコンソールの [Network Analyzer details] (ネットワークアナライザの詳細) ページを選択し、[Edit] (編集) を選択します。その後、新しい設定で設定を更新し、トレースメッセージングをアクティブにして、更新されたメッセージを表示できます。

- 設定にリソースを追加する

  また、ネットワークアナライザの設定にリソースを追加し、リアルタイムでモニタリングすることもできます。のワイヤレスゲートウェイとワイヤレスデバイスリソースを、合わせて最大 250 個追加できます。リソースを追加するには、AWS IoT コンソールの [Network Analyzer details] (ネットワークアナライザの詳細) ページで、[Resources] (リソース) タブを選択し、[Add resources] (リソースを追加) を選択します。その後、新しいリソースで設定を更新し、トレースメッセージングをアクティブにして、追加のリソースの更新されたメッセージを表示できます。

設定の編集とリソースの追加によるネットワークアナライザ設定の更新の詳細については、「リソースを追加し、ネットワークアナライザの設定を更新する」を参照してください。

ネットワークアナライザを使用してマルチキャストグループと FUOTA タスクのデバッグとトラブルシューティングを行う

モニタリングできるワイヤレスリソースには、LoRaWAN デバイス、LoRaWAN ゲートウェイ、マルチキャストグループが含まれます。ネットワークアナライザーを使用して、FUOTA タスクの問題のデバッグとトラブルシューティングを行うこともできます。FUOTA タスクが進行中の場合、セット

アップ、データ送信、ステータスクエリに関連するメッセージをモニタリングおよび追跡することもできます。

FUOTA タスクを監視するには、タスクにマルチキャストグループが含まれている場合は、マルチキャストグループとグループ内のデバイスの両方をネットワークアナライザ設定に追加する必要があります。また、フレーム情報とマルチキャストフレーム情報を有効にして、FUOTA タスクの進行中にマルチキャストグループやデバイスと交換されるユニキャスト/マルチキャストアップリンク/ダウンリンクメッセージを追跡する必要があります。

マルチキャストグループをモニタリングするには、それらをネットワークアナライザの設定に追加し、マルチキャストフレーム情報を使用して、これらのグループに送信されるマルチキャストダウンリンクメッセージのトラブルシューティングを行います。ユニキャスト通信が使用されているグループに参加しようとしているデバイスをトラブルシューティングするには、これらのデバイスもネットワークアナライザの設定に含める必要があります。グループ内のデバイスとのユニキャスト通信のみをモニタリングするには、ワイヤレスデバイスのフレーム情報を有効にします。このアプローチにより、マルチキャストグループとそのグループに参加するデバイスの両方を包括的にモニタリングおよび診断できます。

次のセクションには、ネットワークアナライザを使用してマルチキャストグループと FUOTA タスクのデバッグとトラブルシューティングを行う方法が説明されています。

トピック

- デバイスのみを含む FUOTA タスクのデバッグ
- マルチキャストグループによる FUOTA タスクのデバッグ
- マルチキャストグループに参加しようとしているデバイスをデバッグする
- マルチキャストグループセッションをデバッグする

デバイスのみを含む FUOTA タスクのデバッグ

ネットワークアナライザーを使用して、タスクに LoRaWAN デバイスのみが追加された FUOTA タスクをデバッグできます。FUOTA タスクにデバイスを追加する方法については、「 は デバイスおよびマルチキャストグループを FUOTA タスクに追加して FUOTA セッションをスケジュールする」を参照してください。FUOTA タスクをデバッグするには、次のステップを実行します。

1. ワイヤレスデバイスのフレーム情報を有効にしてネットワークアナライザ設定を作成します。これにより、タスクの進行中にデバイスと交換される FUOTA アップリンクおよびダウンリンクメッセージをモニタリングできます。

2. ワイヤレスデバイス識別子を使用して、FUOTA タスク内のデバイスをネットワークアナライザー設定に追加します。
3. トレースメッセージングをアクティブにして、ネットワークアナライザーの設定のデバイスによるトレースメッセージの受信を開始できます。

トレースメッセージ情報の applicationCommandType 列では、データ転送と断片化設定に関連するユニキャストダウンリンクメッセージの受信を開始します。

> **Note**
>
> トレースメッセージテーブルに applicationCommandType 列が表示されない場合は、この列を表に表示するように設定を調整できます。

ワイヤレスメタデータ > アプリケーション情報の JSON ログメッセージ
で、applicationCommandType およびその他の詳細なメッセージを確認することもできます。

マルチキャストグループによる FUOTA タスクのデバッグ

ネットワークアナライザーを使用して、グループにマルチキャストグループと LoRaWAN デバイスが追加された FUOTA タスクをデバッグできます。FUOTA タスクにデバイスを追加する方法については、「 は デバイスおよびマルチキャストグループを FUOTA タスクに追加して FUOTA セッションをスケジュールする」を参照してください。FUOTA タスクをデバッグするには、次のステップを実行します。

1. ワイヤレスデバイスとマルチキャストグループのフレーム情報とマルチキャストフレーム情報の設定をアクティブにして、ネットワークアナライザー設定を作成します。
2. FUOTA タスク内のマルチキャストグループを、マルチキャストグループ識別子を使用してネットワークアナライザー設定に追加します。マルチキャストフレーム情報を有効にすると、FUOTA タスクの進行中にグループに送信されるファームウェアデータメッセージと FUOTA ステータスクエリメッセージをデバッグできます。
3. ワイヤレスデバイス識別子を使用して、マルチキャストグループ内のデバイスをネットワークアナライザー設定に追加します。フレーム情報を有効にすると、FUOTA タスクの進行中にデバイスと直接交換されるアップリンクメッセージとダウンリンクメッセージをモニタリングできます。
4. トレースメッセージングをアクティブにして、ネットワークアナライザーの設定のデバイスとマルチキャストグループによるトレースメッセージの受信を開始できます。

その後、トレースメッセージテーブルの applicationCommandType 列と JSON ログメッセージの詳細を使用して、トレースメッセージを表示して、それをデバッグできます (「デバイスのみを含む FUOTA タスクのデバッグ」を参照)。

マルチキャストグループに参加しようとしているデバイスをデバッグする

ネットワークアナライザーを使って、マルチキャストグループに参加しようとしているデバイスをデバッグできます。マルチキャストグループにデバイスを追加する方法については、「マルチキャストグループを作成しグループにデバイスを追加する」を参照してください。マルチキャストグループをデバッグするには、次のステップを実行します。

1. ワイヤレスデバイスのフレーム情報をアクティブにして、ネットワークアナライザー設定を作成します。
2. ワイヤレスデバイス識別子を使用して、モニタリングするデバイスをネットワークアナライザー設定に追加します。
3. トレースメッセージングをアクティブにして、ネットワークアナライザーの設定のデバイスによるトレースメッセージの受信を開始できます。
4. グループ内のデバイスに対してトレースメッセージが有効になったら、マルチキャストグループへのデバイスの関連付けを開始します。

マルチキャストグループセッションをデバッグする

ネットワークアナライザーを使用してマルチキャストグループセッションをデバッグできます。詳細については、「マルチキャストグループ内のデバイスに送信するダウンリンクメッセージをスケジュールする」を参照してください。マルチキャストグループセッションをデバッグするには、次のステップを実行します。

1. マルチキャストグループのマルチキャストフレーム情報をアクティブにして、ネットワークアナライザー設定を作成します。
2. マルチキャストグループ識別子を使用することにより、モニタリングするマルチキャストグループをネットワークアナライザー設定に追加します。
3. マルチキャストセッションを開始する前に、トレースメッセージングをアクティブにして、マルチキャストグループセッションのトレースメッセージの受信を開始できます。
4. マルチキャストグループセッションを開始し、トレースメッセージテーブルと JSON ログメッセージに表示されるメッセージを確認してステータスをモニタリングします。

トレースメッセージテーブルでは、MulticastAddr が DevAddr 列に表示されます。JSON ログメッセージでは、WirelessMetadata > ApplicationInfo と移動することで、MulticastGroupId およびその他の詳細なメッセージを確認することができます。

AWS IoT Core for LoRaWAN とインターフェース VPC エンドポイント (AWS PrivateLink)

AWS IoT Core for LoRaWAN への接続には、パブリックインターネット経由で接続するのではなく、Virtual Private Cloud (VPC) 内のインターフェイス VPC エンドポイント (AWS PrivateLink) 経由で直接接続することができます。VPC インターフェイスエンドポイントを使用すると、AWS ネットワーク内で VPC と AWS IoT Core for LoRaWAN 間の通信が完全かつ安全に実施されます。

AWS IoT Core for LoRaWAN は、AWS PrivateLink を活用する Amazon Virtual Private Cloud インターフェイスエンドポイントをサポートします。各 VPC エンドポイントは、VPC サブネット内の 1 つ以上の Elastic Network Interface とプライベート IP アドレスで表されます。詳細については、「Amazon VPC ユーザーガイド」の「インターフェイス VPC エンドポイント (AWS PrivateLink)」を参照してください。

VPC とエンドポイントの詳細については、「Amazon VPC とは何か」を参照してください。

AWS PrivateLink の詳細については、「AWS PrivateLink および VPC エンドポイント」を参照してください。

AWS IoT Wireless VPC エンドポイントに関する考慮事項

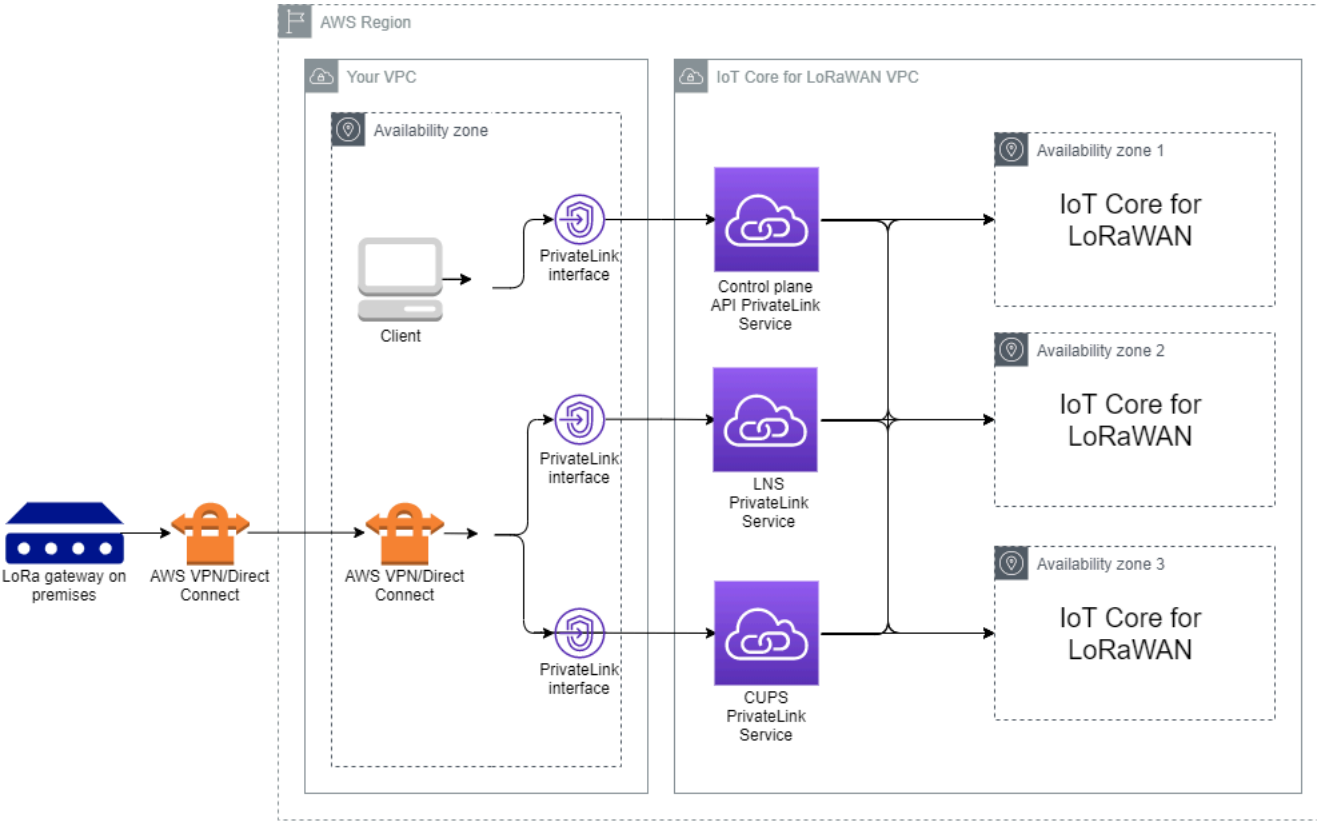
AWS IoT Wireless 用のインターフェイス VPC エンドポイントを設定する前に、Amazon VPC ユーザーガイドの「インターフェイスエンドポイントのプロパティと制限」を確認してください。

AWS IoT Wireless は、VPC からのすべての API アクションの呼び出しをサポートしています。AWS IoT Wireless では、VPC エンドポイントポリシーがサポートされません。デフォルトで、エンドポイント経由での AWS IoT Wireless への完全なアクセスが許可されます。詳細については、「Amazon VPC ユーザーガイド」の「VPC エンドポイントでサービスへのアクセスを制御する」を参照してください。

AWS IoT Core for LoRaWAN privatelink アーキテクチャ

以下の図は、AWS IoT Core for LoRaWAN の privatelink アーキテクチャを示しています。このアーキテクチャは、自身の VPC、AWS IoT Core for LoRaWAN VPC、およびオンプレミス環境の間で

AWS PrivateLink インターフェイスエンドポイントを共有するために Transit Gateway と Route 53 リゾルバーを使用します。より詳しいアーキテクチャ図は、VPC インターフェイスエンドポイントへの接続をセットアップするときに提供されます。



AWS IoT Core for LoRaWAN のエンドポイント

AWS IoT Core for LoRaWAN には、3 つのパブリックエンドポイントがあります。各パブリックエンドポイントには、対応する VPC インターフェイスエンドポイントがあります。パブリックエンドポイントは、コントロールプレーンとデータプレーンエンドポイントに分類できます。これらのエンドポイントについては、「AWS IoT Core for LoRaWAN API エンドポイント」を参照してください。

- Control Plane API エンドポイント

  コントロールプレーン API エンドポイントを使用して AWS IoT Wireless API とやり取りすることができます。これらのエンドポイントは、AWS PrivateLink を使用することによって、Amazon VPC 内でホストされているクライアントからアクセスできます。

- Data Plane API エンドポイント

  データプレーン API エンドポイントは、AWS IoT Core for LoRaWAN の LoRaWAN Network Server (LNS) エンドポイントおよび Configuration and Update Server (CUPS) エンドポイントとや

り取りするために使用できる LNS エンドポイント と CUPS エンドポイントです。これらのエンドポイントは、AWS VPN または AWS Direct Connect を使用することによって、オンプレミスの LoRa ゲートウェイからアクセスできます。これらのエンドポイントは、ゲートウェイを AWS IoT Core for LoRaWAN にオンボードしているときに取得できます。詳細については、「ゲートウェイを AWS IoT Core for LoRaWAN に追加する」を参照してください。

トピック

- AWS IoT Core for LoRaWAN コントロールプレーン API エンドポイントをオンボードする
- AWS IoT Core for LoRaWAN データプレーン API エンドポイントをオンボードする

AWS IoT Core for LoRaWAN コントロールプレーン API エンドポイントをオンボードする

AWS IoT Core for LoRaWAN コントロールプレーン API エンドポイントを使用して AWS IoT Wireless API とやり取りすることができます。例えば、このエンドポイントを使用して SendDataToWirelessDevice API を実行し、AWS IoT から LoRaWAN デバイスにデータを送信できます。詳細については、「AWS IoT Core for LoRaWAN の Control Plane API エンドポイント」を参照してください。

AWS PrivateLink を活用するコントロールプレーンエンドポイントにアクセスするには、Amazon VPC 内でホストされているクライアントが使用できます。これらのエンドポイントを使用して、パブリックインターネット経由で接続するのではなく、Virtual Private Cloud (VPC) 内のインターフェイスエンドポイント経由で AWS IoT Wireless API に接続します。

コントロールプレーンをオンボードするには:

- Amazon VPC とサブネットを作成する
- サブネット内でAmazon EC2 インスタンスを起動します。
- Amazon VPC インターフェイスエンドポイントを作成します
- SMTP インターフェイスエンドポイントへの接続をテストします

Amazon VPC とサブネットを作成する

インターフェイスエンドポイントに接続する前に、VPC とサブネットを作成する必要があります。次に、サブネットで EC2 インスタンスを起動します。これは、インターフェイスエンドポイントに接続するために使用できます。

VPC を作成するには：

1. Amazon VPC コンソールの [VPC] ページに移動して、[Create VPC] (VPC を作成) をクリックします。
2. [Create VPC] (VPC を作成) ページで以下を実行します。
   - [VPC Name tag - optional] (名前タグ – オプション) に名前を入力します (例、**`VPC-A`**)。
   - [IPv4 CIDR block] (IPv4 CIDR ブロック) に VPC の IPv4 アドレス範囲を入力します (例、**`10.100.0.0/16`**)。
3. 他のフィールドはデフォルト値のままにしておき、[Create VPC] (VPC を作成) を選びます。

サブネットを作成するには：

1. Amazon VPC コンソールの [Subnets] (サブネット) ページに移動し、[Create subnet] (サブネットを作成) をクリックします。
2. [Create subnet group] (サブネットグループを作成) ページで：
   - [VPC ID] には、先ほど作成した VPC を選択します (例、`VPC-A`)。
   - [Subnet name] (サブネット名) に名前を入力します (例、**`Private subnet`**)。
   - サブネットの [Availability Zone] (アベイラビリティーゾーン) を選択します。
   - [IPv4 CIDR block] (IPv4 CIDR ブロック) にサブネットの IP アドレスブロックを CIDR 形式で入力します (例、**`10.100.0.0/24`**)。
3. [Create subnet] (サブネットを作成) をクリックしてサブネットを作成し、VPC に追加します。

詳細については、[Work with VPC and subnets](VPC とサブネットの使用)を参照してください。

サブネット内でAmazon EC2 インスタンスを起動します。

EC2 インスタンスを起動するには：

1. Amazon EC2 コンソールに移動して、[Launch Instance] (インスタンスを起動) をクリックします。
2. AMI には [Amazon Linux 2 AMI (HVM), SSD Volume Type] を選択して、次に [t2 micro] インスタンスタイプを選択します。インスタンスの詳細を設定するために、[Next] (次のステップ) を選択します。
3. [Configure Instance Details] (インスタンスの詳細の設定) ページで：
   - [Network] (ネットワーク) には、先ほど作成した VPC を選択します (例、`VPC-A`)。

- Subnet] (サブネット) には、先ほど作成したサブネットを選択します (例、**`Private subnet`**)。
- [IAM role] (IAM ロール) にはロール AWSIoTWirelessFullAccess を選択して、AWS IoT Core for LoRaWAN に完全なアクセスポリシーを付与します。詳細については、「AWSIoTWirelessFullAccess ポリシーの概要」を参照してください。
- [Assume Private IP] (プライベート IP の継承) には、10.100.0.42 などの IP アドレスを使用します。

4. [Next: Add Storage] (次のステップ: ストレージの追加)] を選んでから、[Next: Add Tags] (次のステップ: タグの追加) を選びます。オプションで、EC2 インスタンスに関連付ける任意のタグを追加できます。[Next: Configure Security Group] (次に：セキュリティグループを設定) を選択します。
5. [Configure Security Group] (セキュリティグループの設定) ページで、以下を許可するセキュリティグループを設定します。
   - `10.200.0.0/16` に設定した [Source] (ソース) に対して [All TCP] (すべての TCP) を開放します
   - `10.200.0.0/16` に設定した [Source] (ソース) に対して [All ICMP - IPV4] (すべての ICMP-IPV4) を開放します
6. [Review and Launch] (確認と作成) をクリックしてインスタンスの詳細を確認し、EC2 インスタンスを起動します。

詳細については、「Amazon EC2 Linux インスタンスの開始方法」を参照してください。

Amazon VPC インターフェイスエンドポイントを作成します

VPC 用の VPC エンドポイントが作成できます。作成後は、EC2 API がこのエンドポイントにアクセスすることができます。エンドポイントを作成するには：

1. VPC の [Endopoints] (エンドポイント) コンソールに移動して、[Create Endpoint] (エンドポイントの作成) をクリックします。
2. [Create Endpoint] (エンドポイントの作成) ページで、以下の情報を指定します。
   - [Service category] (サービスカテゴリ) には [AWS のサービスs] を選択します。
   - [Service Name] (サービス名) については、キーワード **`iotwireless`** を入力して検索します。表示された `iotwireless` サービスのリストで、お使いのリージョンのコントロールプレーン API エンドポイントを選択します。エンドポイントは `com.amazonaws.`*`region`*`.iotwireless.api` の形式です。

- [VPC] と [Subnets] (サブネット) には、エンドポイントを作成する VPC と、エンドポイントネットワークを作成するアベイラビリティーゾーン (AZ) を選択します。

  > **Note**
  >
  > iotwireless サービスは、一部のアベイラビリティーゾーンをサポートしていない場合があります。

- [Enable DNS name] (DNS 名を有効にする) には、[Enable for this endpoint] (このエンドポイントで有効にする) を選択します。

  このオプションを選択すると、後で接続をテストするために使用する API がプライベートリンクのエンドポイントを通過するように、DNS が自動的に解決され、Amazon Route 53 Public Data Planeにルートが設定されます。

- [Security group] (セキュリティグループ) には、エンドポイントネットワークインターフェイスに関連付けるセキュリティグループを選択します。
- オプションで、タグを追加または削除できます。タグとは名前と値のペアで、エンドポイントに関連付けるために使用します。

3. [Create Endpoint] (エンドポイントの作成) をクリックして、VPC エンドポイントを作成します。

SMTP インターフェイスエンドポイントへの接続をテストします

SSH を使用して Amazon EC2 インスタンスにアクセスしたから、AWS CLIを使用してプライベートリンクインターフェイスエンドポイントに接続する事ができます。

インターフェイスエンドポイントに接続する前に、「Linux での AWS CLI バージョン 2 のインストール、更新、アンインストール」で説明されている手順に従って、最新の AWS CLI バージョンをダウンロードしてください。

以下の例は、CLI を使用してインターフェイスエンドポイントへの接続をテストする方法を示しています。

```
aws iotwireless create-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com  \
    --name='test-privatelink'
```

以下の例は、コマンドを実行する例を示しています。

```
Response:
{
 "Arn": "arn:aws:iotwireless:region:acct_number:ServiceProfile/1a2345ba-4c5d-67b0-ab67-e0c8342f2857",
 "Id": "1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
}
```

同様に、次のコマンドを実行して、サービスプロファイル情報を取得したり、すべてのサービスプロファイルを一覧表示したりできます。

```
aws iotwireless get-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
    --id="1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
```

以下は、list-device-profiles コマンドの例です。

```
aws iotwireless list-device-profiles \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
```

AWS IoT Core for LoRaWAN データプレーン API エンドポイントをオンボードする

AWS IoT Core for LoRaWAN データプレーンのエンドポイントは、次のエンドポイントで構成されます。これらのエンドポイントは、ゲートウェイを AWS IoT Core for LoRaWAN に追加するときに取得します。詳細については、「ゲートウェイを AWS IoT Core for LoRaWAN に追加する」を参照してください。

- LoRaWAN Network Server (LNS)エンドポイント

  LNS エンドポイントは*account-specific-prefix*.lns.lorawan.*region*.amazonaws.com形式です。このエンドポイントを使用して、LoRaアップリンクおよびダウンリンクメッセージを交換するための接続を確立できます。

- 設定および更新サーバー(CUPS)エンドポイント

  CUPS エンドポイントは*account-specific-prefix*.cups.lorawan.*region*.amazonaws.com形式です。このエンドポイントは、認証情報管理、リモート設定、ゲートウェイのファームウェア更新に使用できます。

詳細については、「CUPS および LNS プロトコルの使用」を参照してください。

AWS アカウント とリージョンの Data Plane API エンドポイントを検索するには、ここに記載されている get-service-endpoint CLI コマンド、または `GetServiceEndpoint` REST API を使用します。詳細については、「AWS IoT Core for LoRaWAN の Data Plane API エンドポイント」を参照してください。

オンプレミスの LoRaWAN ゲートウェイを接続して、AWS IoT Core for LoRaWAN エンドポイントと通信できます。この接続を確立するには、まず VPN 接続を使用して、オンプレミスのゲートウェイを VPC 内の AWS アカウントに接続します。その後、AWS IoT Core for LoRaWAN VPC 内で、privatelink を活用するデータプレーンインターフェイスエンドポイントと通信できるようになります。

以下は、これらのエンドポイントをオンボードする方法を説明しています。

- VPC インターフェイスエンドポイントとプライベートホストゾーンを作成します
- VPNを使用して、LoRaゲートウェイをAWS アカウントに接続します

VPC インターフェイスエンドポイントとプライベートホストゾーンを作成します

AWS IoT Core for LoRaWAN には、Configuration and Update Server (CUPS) エンドポイントと LoRaWAN Network Server (LNS) エンドポイントという 2 つのデータプレーンエンドポイントがあります。両方のエンドポイントへのプライベートリンク接続を確立するセットアッププロセスは同じであるため、例として、LNS エンドポイントを使用できます。

データプレーンエンドポイントの場合、LoRa ゲートウェイはまず Amazon VPC 内の AWS アカウント に接続し、次に AWS IoT Core for LoRaWAN VPC 内の VPC エンドポイントに接続します。

エンドポイントに接続する場合、DNS 名は 1 つの VPC 内で解決できますが、複数の VPC 間で解決することはできません。エンドポイントの作成時にプライベート DNS を無効にするには、[Enable DNS name] (DNS 名を有効にする) 設定を無効にします。Route 53 が VPC の DNS クエリに応答する方法に関する情報を提供するには、プライベートホストゾーンを使用できます。VPC をオンプレミス環境と共有するには、Route 53 リゾルバーを使用してハイブリッド DNS を円滑に稼働させることができます。

このチュートリアルの手順を完了するには、次のステップを実行します。

- Amazon VPC とサブネットを作成します
- Amazon VPC インターフェイスエンドポイントを作成します

Amazon VPC とサブネットを作成します

コントロールプレーンエンドポイントのオンボーディング時に作成した Amazon VPC とサブネットを再利用できます。詳細については、Amazon VPC とサブネットを作成する を参照してください。

Amazon VPC インターフェイスエンドポイントを作成します

VPC 用に VPC エンドポイントを作成できます。これは、コントロールプレーンエンドポイント用にエンドポイントを作成する方法と似ています。

1. VPC の [Endopoints] (エンドポイント) コンソールに移動して、[Create Endpoint] (エンドポイントの作成) をクリックします。
2. [Create Endpoint] (エンドポイントの作成) ページで、以下の情報を指定します。
   - [Service category] (サービスカテゴリ) には [AWS のサービスs] を選択します。
   - Service Name] (サービス名) については、キーワード **lns**. を入力して検索します。表示された `lns` サービスのリストで、お使いのリージョンの LNS データプレーン API エンドポイントを選択します。エンドポイントは `com.amazonaws.region.lorawan.lns` の形式です。

     > **Note**
     >
     > CUPS エンドポイントでこの手順を実行している場合は、cupsを検索します。エンドポイントは `com.amazonaws.region.lorawan.cups` の形式です。

   - [VPC] と [Subnets] (サブネット) には、エンドポイントを作成する VPC と、エンドポイントネットワークを作成するアベイラビリティーゾーン (AZ) を選択します。

     > **Note**
     >
     > `iotwireless`サービスは、一部のアベイラビリティーゾーンをサポートしていない場合があります。

   - [Enable DNS name] (DNS 名を有効にする) で、[Enable for this endpoint] (このエンドポイントで有効にする) が選択されていないことを確認してください。

このオプションを選択しない事で、VPC エンドポイントのプライベート DNS を無効にして、代わりにプライベートホストゾーンを使用する事ができます。

- [Security group] (セキュリティグループ) には、エンドポイントネットワークインターフェイスに関連付けるセキュリティグループを選択します。
- オプションで、タグを追加または削除できます。タグとは名前と値のペアで、エンドポイントに関連付けるために使用します。

3. [Create Endpoint] (エンドポイントの作成) をクリックして、VPC エンドポイントを作成します。

プライベートホストゾーンを設定します

プライベートリンクエンドポイントを作成した後、詳細タブに、DNS 名のリストが表示されます。これらの DNS 名のいずれかを使用して、プライベートホストゾーンを設定できます。DNS名は `vpce-xxxx.lns.lorawan.region.vpce.amazonaws.com` 形式になります。

プライベートホストゾーンを作成します

プライベートホストゾーンを作成するには

1. Route 53 の [Hosted zones] (ホストゾーン) コンソールに移動して、[Create hosted zone] (ホストゾーンの作成) をクリックします。
2. [Create hosted zone] (ホストゾーンの作成) ページで、以下の情報を指定します。
   - [Domain name] (ドメイン名) には、LNS エンドポイントの完全なサービス名である **`lns.lorawan.region.amazonaws.com`** を入力します。

     > **Note**
     >
     > CUPS エンドポイントでこの手順を実行している場合は、**`cups.lorawan.region.amazonaws.com`**を入力します。

   - [Type] (タイプ) には、[Private Hosted Zone] (プライベートホストゾーン) を選択します。
   - 必要に応じて、タグを追加または削除してホストゾーンに関連付けることができます。
3. プライベートホストゾーンを作成するには、ホストゾーンの作成を選んでください。

詳細については、「プライベートホストゾーンの作成」を参照してください。

プライベートホストゾーンを作成したら、そのドメインにトラフィックをルーティングする方法を DNS に指示するレコードを作成できます。

レコードを作成する

プライベートホストゾーンを作成したら、そのドメインにトラフィックをルーティングする方法を DNS に指示するレコードを作成できます。レコードを作成するには：

1. 表示されるホストゾーンのリストで、前に作成したプライベートホストゾーンを選び、レコードを作成するを選びます。
2. ウィザードを使用してレコードを作成します。コンソールに [Quick create] (クイック作成) 方式が表示された場合は、[Switch to wizard] (ウィザードに切り替える) をクリックします。
3. [Routing policy] (ルーティングポリシー) に [Simple Routing] (シンプルルーティング) を選択し、[Next] (次へ) を選びます。
4. [Configure records] (レコードを設定)ページで、[Define simple record] (シンプルなレコードを定義) を選択します。
5. [Define simple record] (シンプルなレコードを定義) ページで
   - [Record name] (レコード名) には AWS アカウント番号のエイリアスを入力します。この値は、ゲートウェイのオンボーディング時に取得、または `GetServiceEndpoint` REST API を使用して取得します。
   - [Record type] (レコードタイプ) では、値を `A - Routes traffic to an IPv4 address and some AWS resources` のままにしておきます。
   - [Value/Route traffic to (値/トラフィックのルーティング先)] には、[Alias to VPC endpoint (VPC エンドポイントへのエイリアス)] を選択します。次に、お使いの [Region] (地域) を選択し、表示されたエンドポイントのリストから、「Amazon VPC インターフェイスエンドポイントを作成します」の説明に従って先ほど作成したエンドポイントを選択します。
6. [Define simple record] (シンプルなレコードを定義) をクリックしてレコードを作成します。

Route 53 インバウンドリゾルバを設定します

VPC エンドポイントをオンプレミス環境と共有するには、Route 53 リゾルバーを使用してハイブリッド DNS を円滑に稼働させる事ができます。インバウンドリゾルバーは、オンプレミスネットワークからデータプレーンエンドポイントへのトラフィックを、パブリックインターネットを経由せずにルーティングすることを可能にします。サービスのプライベート IP アドレスの値を返すには、VPC エンドポイントと同じ VPC 内に Route 53 リゾルバーを作成します。

インバウンドリゾルバーを作成する場合、VPC と以前にアベイラビリティーゾーン (AZ) で作成したサブネットのみを指定する必要があります。Route 53 リゾルバはこの情報を使用して、各サブネットにトラフィックをルーティングするための IP アドレスを自動的に割り当てます。

インバウンドリゾルバーを作成するには：

1. Route 53 の [Inbound endpoints] (インバウンドエンドポイント) コンソールに移動して、[Create inbound endpoint] (インバウンドエンドポイントの作成) をクリックします。

   > **Note**
   >
   > エンドポイントとプライベートホストゾーンの作成時に使用したものと同じ AWS リージョンを使用していることを確認してください。

2. [Create inbound endpoint] (インバウンドエンドポイントの作成) ページで、以下の情報を指定します。
   - [Endpoint name] (エンドポイント名) に名前を入力します (例、**`VPC_A_Test`**)。
   - [VPC in the region] (該当リージョンの VPC) には、VPC エンドポイントの作成時に使用したものと同じ VPC を選択します。
   - [Security group for this endpoint] (このエンドポイントのセキュリティグループ) を設定して、オンプレミスネットワークからの受信トラフィックを許可します。
   - IP アドレスのために、[Use an IP address that is selected automatically.] (自動的に選択された IP アドレスを使用します。)を選びます。
3. [Submit] (送信) をクリックして、インバウンドリゾルバーを作成します。

この例では、トラフィックをルーティングするためのインバウンド Route 53 リゾルバーに IP アドレス `10.100.0.145` と `10.100.192.10` が割り当てられたとしましょう。

次のステップ

DNS エントリのトラフィックをルーティングするプライベートホストゾーンとインバウンドリゾルバーを作成しました。Site-to-Site VPN エンドポイントまたはClient VPN エンドポイントのいずれかが使用できるようになりました。詳細については、「VPNを使用して、LoRaゲートウェイをAWS アカウントに接続します」を参照してください。

VPNを使用して、LoRaゲートウェイをAWS アカウントに接続します

オンプレミスのゲートウェイを AWS アカウントに接続するには、Site-to-Site VPN 接続またはクライアント VPN エンドポイントのどちらかを使用できます。

オンプレミスゲートウェイに接続する前に、VPC エンドポイントを作成し、ゲートウェイからのトラフィックがパブリックインターネットを経由しないように、プライベートホストゾーンとインバウンドリゾルバーを設定しておく必要があります。詳細については、「VPC インターフェイスエンドポイントとプライベートホストゾーンを作成します」を参照してください。

Site-to-Site VPN エンドポイント

ゲートウェイハードウェアがない、または別の AWS アカウントを使用して VPN 接続をテストしたい場合は、Site-to-Site VPN 接続を使用できます。Site-to-Site VPN を使用して、同じ AWS アカウントから、または異なる AWS リージョンで使用している別の AWS アカウントから VPC エンドポイントに接続できます。

> **Note**
>
> ゲートウェイハードウェアがあり、VPN 接続を設定したい場合は、代わりにClient VPN を使用することをお勧めします。手順については、クライアント VPN エンドポイント を参照してください。

Site-to-Site VPN を設定するには

1. サイト内で、接続のセットアップ元になる別の VPC を作成します。VPC-Aには、先程作成した VPC を再利用できます。別の VPC を作成するには (例、VPC-B)、以前に作成した VPC の CIDR ブロックと重複しない CIDR ブロックを使用します。

   VPC のセットアップについては、AWS setup Site-to-Site VPN connection で説明されている手順に従ってください。

   > **Note**
   >
   > このドキュメントで説明されているSite-to-Site VPN VPN 方式では、VPN 接続に OpenSwan を使用しており、これは 1 つのVPNトンネルしかサポートしません。VPN 用に別の商用ソフトウェアを使用している場合は、サイト間の 2 つのトンネルを設定できる場合があります。

2. VPN 接続をセットアップしたら、AWS アカウントからのインバウンドリゾルバーの IP アドレスを追加して、`/etc/resolv.conf` ファイルを更新します。この IP アドレスは、ネームサーバーに使用します。この情報の入手方法については、「Route 53 インバウンドリゾルバを設定します」を参照してください。この例には、Route 53 リゾルバーを作成したときに割り当てられた IP アドレス `10.100.0.145` を使用できます。

```
options timeout:2 attempts:5
; generated by /usr/sbin/dhclient-script
search region.compute.internal
nameserver 10.100.0.145
```

3. `nslookup` コマンドを使用して、VPN 接続がパブリックインターネットを経由する代わりに AWS PrivateLink エンドポイントを使用しているかどうかをテストできるようになりました。以下の例は、コマンドを実行する例を示しています。

```
nslookup account-specific-prefix.lns.lorawan.region.amazonaws.com
```

以下は、コマンド実行後の出力例です。これには、AWS PrivateLink LNS エンドポイントに対して接続が確立されたことを示すプライベート IP アドレスが表示されています。

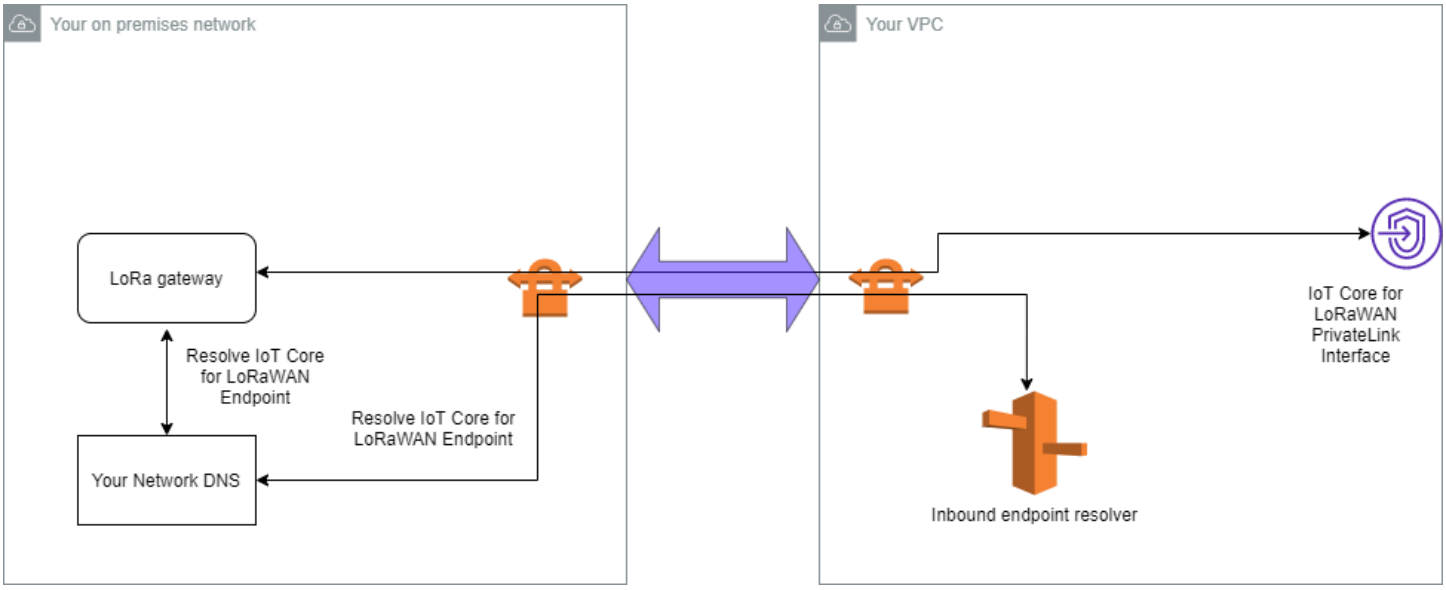
```
Server: 10.100.0.145
Address: 10.100.0.145

Non-authoritative answer:
Name: https://xxxxx.lns.lorawan.region.amazonaws.com
Address: 10.100.0.204
```

Site-to-Site VPN 接続の使用については、「Site-to-Site VPN の仕組み」を参照してください。

クライアント VPN エンドポイント

AWS Client VPN は、AWS リソースとオンプレミスネットワーク内のリソースに安全にアクセスするための、クライアントベースのマネージド VPN サービスです。次に、クライアント VPN サービスのアーキテクチャを示します。



Client VPN エンドポイントへの VPN 接続を確立するには

1. 「AWS Client VPN の開始方法」で説明されている手順に従って、クライアント VPN エンドポイントを作成します。
2. オンプレミスネットワーク (例、Wi-Fi ルーター) に、そのルーターのアクセス URL (例、192.168.1.1) を使用してログインし、そしてルート名とパスワードを見つけます。
3. ゲートウェイのドキュメントにある指示に従って LoRaWAN ゲートウェイをセットアップしてから、ゲートウェイを AWS IoT Core for LoRaWAN に追加します。ゲートウェイを追加する方法については、「ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。
4. ゲートウェイのファームウェアが、最新の物であるかどうかを確認します。ファームウェアが古くなっている場合は、オンプレミスネットワークで提供されている指示に従って、ゲートウェイのファームウェアを更新できます。詳細については、「AWS IoT Core for LoRaWAN で CUPS サービスを使用してゲートウェイファームウェアを更新する」を参照してください。
5. OpenVPN が有効になったかどうかを確認します。有効になっている場合は、次の手順に進み、オンプレミスネットワーク内で OpenVPN クライアントを設定します。有効になっていない場合は、「Guide to install OpenVPN for OpenWrt」 (OpenWrt 用に OpenVPN をインストールするガイド) の手順に従ってください。

> **Note**
>
> この例では、OpenVPN を使用します。クライアント VPN 接続のセットアップには、AWS VPN または AWS Direct Connect などの他の VPN クライアントを使用できます。

6. クライアント設定からの情報と、Luci を使用した OpenVPN クライアントの使用方法に基づいて、OpenVPN クライアントを設定します。
7. オンプレミスネットワークに SSH 接続し、AWS アカウント内のインバウンドリゾルバーの IP アドレス (`10.100.0.145`) を追加して `/etc/resolv.conf` ファイルを更新します。
8. ゲートウェイトラフィックがAWS PrivateLinkを使用してエンドポイントに接続する場合は、ゲートウェイの最初の DNS エントリをインバウンドリゾルバーの IP アドレスに置き換えます。

Site-to-Site VPN 接続の使用については、「クライアント VPN の開始方法」を参照してください。

LNS および CUPS VPC エンドポイントに接続する

以下に、LNS および CUPS VPC エンドポイントへの接続をテストする方法を示します。

CUPS エンドポイントのテスト

LoRa ゲートウェイから CUPS エンドポイントへの AWS PrivateLink 接続をテストするには、以下のコマンドを実行します。

```
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
```

LNS エンドポイントのテスト

LNSエンドポイントをテストするには、まずワイヤレスゲートウェイで動作するLoRaWANデバイスをプロビジョニングします。その後、デバイスを追加し、joinプロシージャーを使用して、アップリンクメッセージの送信を開始できます。

AWS IoT Core for Amazon Sidewalk

AWS IoT Core for Amazon Sidewalk では、Sidewalk エンドデバイスを AWS クラウド に接続したり、他の AWS のサービス を使用したりするために利用できるクラウドサービスを提供しています。

Amazon Sidewalk は安全な共有ネットワークで、コミュニティ内のデバイスを接続して、接続を維持することができます。Amazon Sidewalk は、Sidewalk エンドデバイスと Sidewalk ゲートウェイの間、および Sidewalk ゲートウェイと Sidewalk クラウドの間でデータを転送します。

AWS IoT Core for Amazon Sidewalk へのアクセス

コンソールまたは AWS IoT Wireless API オペレーションを使用して、Sidewalk エンドデバイスを AWS IoT にオンボードできます。デバイスがオンボーディングされると、AWS IoT Core にメッセージが送信されます。その後、Amazon Sidewalk エンドデバイスのデータを使用する AWS クラウド上でビジネスアプリケーションの開発を開始できます。

コンソールを使用する場合

Sidewalk エンドデバイスをオンボードするには、AWS Management Console にサインインして AWS IoT コンソールの [デバイス] ページに移動します。デバイスがオンボーディングされると、IoT コンソールのこのページでデバイスを表示および管理できます。

API または CLI の使用

AWS IoT Wireless API オペレーションを使用して、Sidewalk デバイスと LoRaWAN デバイスの両方をオンボードできます。AWS IoT Core が構築されている AWS IoT Wireless API は、AWS SDK でサポートされています。詳細については、AWS SDK およびツールキットを参照してください。

AWS CLI を使用して、Sidewalk エンドデバイスをオンボードおよび管理するためのコマンドを実行します。詳細については、「AWS IoT Wireless CLI リファレンス」を参照してください。

AWS IoT Core for Amazon Sidewalk のリージョンとエンドポイント

Amazon Sidewalk は `us-east-1` AWS リージョン でのみご利用いただけます。AWS IoT Core for Amazon Sidewalk では、このリージョンのコントロールプレーンとデータプレーン API エンドポイントのサポートを提供しています。データプレーン API エンドポイントは、AWS アカウント に固有

です。詳細については、AWS 全般のリファレンスガイドの「AWS IoT Wireless サービスエンドポイント」を参照してください。

AWS IoT Core for Amazon Sidewalk には、デバイスと AWS クラウド 間で送信されるデバイスデータと、AWS IoT Wireless API オペレーションの最大 TPS に適用されるクォータがあります。詳細については、「AWS 全般のリファレンス」の「AWS IoT Wireless クォータ」を参照してください。

AWS IoT Core for Amazon Sidewalk の料金

AWS にサインアップすると、AWS 無料利用枠を利用して、無料で AWS IoT Core for Amazon Sidewalk の使用を開始することができます。

全般的な製品の概要と料金の詳細については、「AWS IoT Core の料金」を参照してください。

AWS IoT Core for Amazon Sidewalk の概要

AWS IoT Core for Amazon Sidewalk を使用すると、Amazon Sidewalk エンドデバイスを AWS IoT にオンボードして管理およびモニタリングできます。また、デバイスデータを他の AWS のサービスに送信する送信先も管理します。

AWS IoT Core for Amazon Sidewalk の機能

AWS IoT Core for Amazon Sidewalk を使用すると、次のことができます。

- Sidewalk エンドデバイスを、AWS IoT コンソール、 AWS IoT Core for Amazon Sidewalk の API オペレーション、または AWS CLI コマンドを使用して AWS IoT にオンボードします。
- AWS クラウド が提供する機能を活用します。
- AWS IoT ルールを使用して受信ペイロードメッセージを処理し、他の AWS のサービス とやり取りする送信先を作成します。
- イベント通知を有効にすると、Sidewalk エンドデバイスがプロビジョニングまたは登録されたとき、ダウンリンクメッセージが正常にデバイスに配信されたかどうかなどのイベントに関するメッセージを受信できます。
- Sidewalk エンドデバイスをリアルタイムでログに記録してモニタリングし、有用なインサイトを得て、エラーを特定しトラブルシューティングします。
- Sidewalk エンドデバイスを AWS IoT モノと関連付けると、デバイスの表現をクラウドに簡単に保存できます。AWS IoT におけるモノを使用すると、機能の検索や管理、他の AWS IoT Core 機能へのアクセスを簡単に行えるようになります。

以下のトピックは、Amazon Sidewalk と AWS IoT Core for Amazon Sidewalk について学習するのに役立ちます。

トピック

Amazon Sidewalk の概要

Amazon Sidewalk は、互換性のある Amazon Echo デバイスや Ring デバイスなどの Amazon Sidewalk Bridges を使用して IoT デバイスにクラウド接続を提供する安全なコミュニティネットワークです。Amazon Sidewalk は、短距離通信には Bluetooth LE、長距離通信には 周波数 900 MHz の LoRa および FSK 無線プロトコルを使用して、自宅やそれ以外の場所での低帯域幅で長距離の接続を可能にします。

Amazon Sidewalk を有効にすると、このネットワークはコミュニティ内の他の Sidewalk エンドデバイスをサポートすることができ、環境のセンシングなどのアプリケーションに使用することができます。Amazon Sidewalk は、デバイスの接続と維持に役立ちます。

Amazon Sidewalk の機能

Amazon Sidewalk の機能を次に示します。

- Amazon Sidewalk は、Ring デバイスと一部の Echo デバイスを含む Sidewalk ゲートウェイを使用して低帯域幅のネットワークを構築します。ゲートウェイを使用すると、インターネット帯域幅の一部を共有し、それを使用してエンドデバイスをネットワークに接続できます。
- Amazon Sidewalk は、複数のレイヤーから成る暗号化とセキュリティを備えた安全なネットワーキングメカニズムを提供します。
- Amazon Sidewalk には、Sidewalk への参加を有効または無効にする簡単なメカニズムが用意されています。

Amazon Sidewalk のコンセプト

Amazon Sidewalk の主要なコンセプトは次のとおりです。

Sidewalk ゲートウェイ

Sidewalk ゲートウェイ、または Amazon Sidewalk Bridges は、Sidewalk エンドデバイスとクラウド間でデータをルーティングします。ゲートウェイとは、SubG-CSS (非同期、LDR)、SubG-FSK (同期、HDR)、または Sidewalk 通信用の Bluetooth LE をサポートする Echo デバイスや Ring Floodlight Cam などのAmazonデバイスです。Sidewalk ゲートウェイは、インターネット帯域幅の一部を Sidewalk コミュニティと共有して、Sidewalk 対応デバイスのグループに接続できるようにします。

Sidewalk エンドデバイス

Sidewalk エンドデバイスは、Sidewalk ゲートウェイに接続することで Amazon Sidewalk をローミングします。エンドデバイスは、Sidewalk 対応ライトやドアロックなど、低帯域幅、低消費電力のスマート製品です。

> **Note**
>
> 一部の Sidewalk ゲートウェイはエンドデバイスとしても機能します。

Sidewalk Network Server

Amazon が運営する Sidewalk Network Server は、Sidewalk ネットワークの時刻同期を保ちながら、受信パケットを検証し、アップリンクとダウンリンクのメッセージを目的の送信先にルーティングします。

Amazon Sidewalk の詳細情報

Amazon Sidewalk の詳細については、以下のウェブページを参照してください。

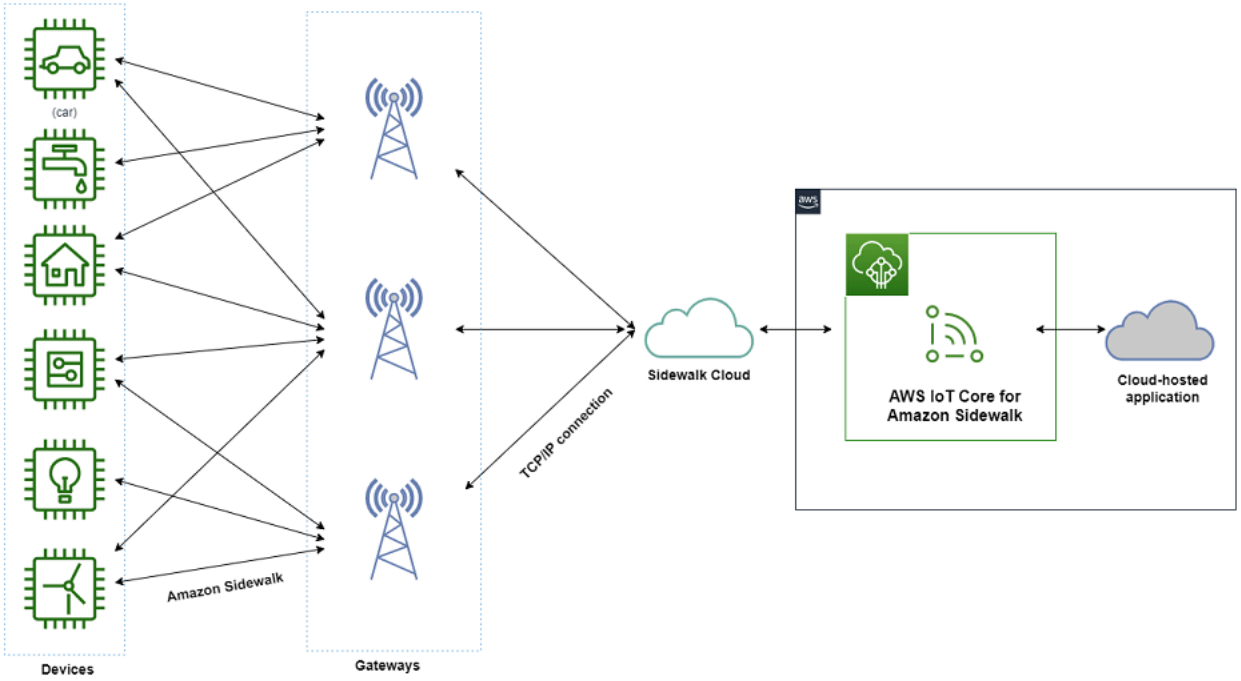
- Amazon Sidewalk
- Amazon Sidewalk ドキュメント
- AWS IoT Core for Amazon Sidewalk

AWS IoT Core for Amazon Sidewalk の仕組み

AWS IoT Core for Amazon Sidewalk を使用すると、Amazon Sidewalk エンドデバイスを AWS IoT にオンボードして管理およびモニタリングできます。また、デバイスデータを他の AWS のサービスに送信する送信先も管理します

AWS IoT Core for Amazon Sidewalk では、Sidewalk エンドデバイスを AWS クラウド に接続したり、他の AWS のサービス を使用したりするために利用できるクラウドサービスを提供しています。また、AWS IoT Core for Amazon Sidewalk を使用して Sidewalk デバイスを管理し、デバイス上のアプリケーションをモニタリングおよび構築することができます。

Sidewalk エンドデバイスは、Sidewalk ゲートウェイを介して AWS IoT Core と通信します。AWS IoT Core for Amazon Sidewalk は、AWS IoT Core が Sidewalk エンドデバイスやゲートウェイを管理し通信するために必要なサービスおよびデバイスポリシーを管理します。また、デバイスデータを他の AWS のサービス に送信する送信先も管理します。



AWS IoT Core for Amazon Sidewalk の使用を開始する

AWS IoT コンソール、AWS IoT Core for Amazon Sidewalk API、または AWS CLI を使用して Sidewalk エンドデバイスを作成してオンボードし、Sidewalk ネットワークに接続できます。Amazon Sidewalk の開始方法およびエンドデバイスの AWS IoT へのオンボーディングについては、以下のトピックを参照してください。

- AWS IoT Core for Amazon Sidewalk の開始方法

  このトピックでは、Sidewalk エンドデバイスをオンボーディングするための前提条件を説明し、センサーモニタリングアプリケーションを使用するワークフローを説明し、AWS CLI コマンドを使用してデバイスをオンボードする方法の概要を説明します。

- AWS IoT Core for Amazon Sidewalk への接続

このセクションでは、オンボーディングワークフローの導入のさまざまなステップについて説明し、コンソールと API オペレーションを使用してエンドデバイスをオンボーディングする方法について説明します。また、デバイスを接続して、デバイスと AWS IoT Core for Amazon Sidewalk の間で交換されるメッセージを表示することもできます。

- AWS IoT Core for Amazon Sidewalk を使用したデバイスの一括プロビジョニング

  このセクションでは、AWS IoT Core for Amazon Sidewalk を使用して Sidewalk エンドデバイスを一括プロビジョニングするためのステップバイステップのチュートリアルについて説明します。一括プロビジョニングのワークフローと、多数の Sidewalk デバイスをオンボーディングする方法を学びます。

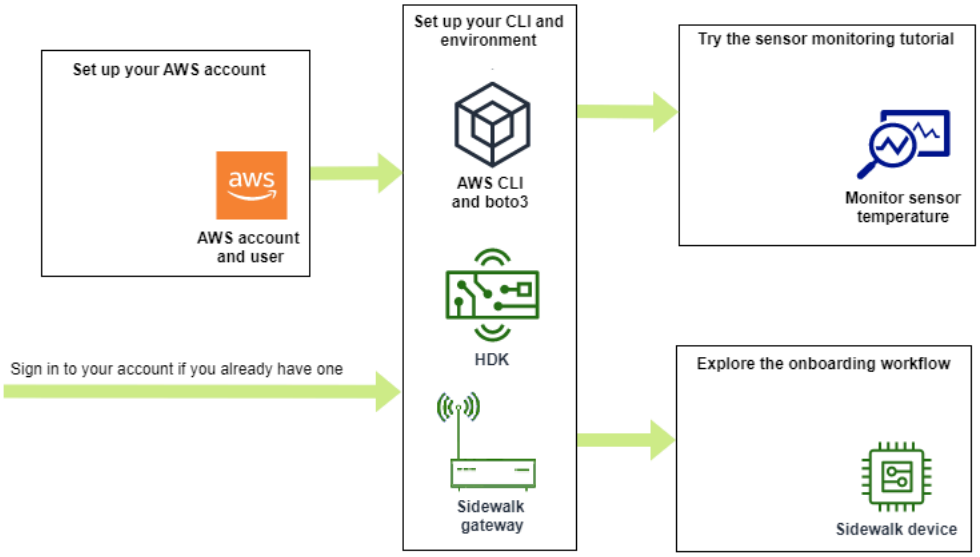
AWS IoT Core for Amazon Sidewalk の詳細情報

AWS IoT Core for Amazon Sidewalk の詳細については、以下のウェブページを参照してください。

- Amazon Sidewalk
- Amazon Sidewalk ドキュメント
- AWS IoT Core for Amazon Sidewalk

AWS IoT Core for Amazon Sidewalk の開始方法

このセクションでは、Sidewalk のエンドデバイスの AWS IoT Core for Amazon Sidewalk への接続を開始する方法について説明します。エンドデバイスを Amazon Sidewalk に接続し、それらの間でメッセージを渡す方法について説明します。また、Sidewalk のサンプルアプリケーションや、AWS IoT Core for Amazon Sidewalk を使用してセンサーモニタリングを実行する方法の概要についても学びます。サンプルアプリケーションには、センサー温度の変化を表示およびモニタリングするためのダッシュボードが用意されています。



次のトピックは、AWS IoT Core for Amazon Sidewalk の使用を開始する際に役立ちます。

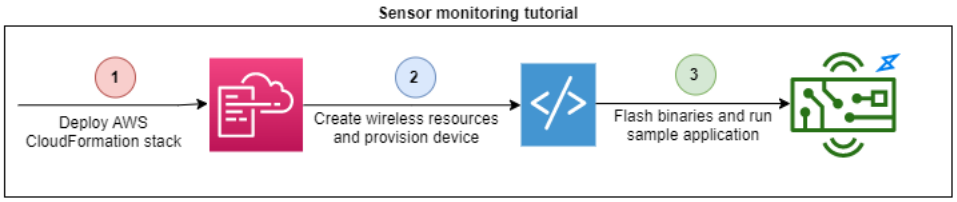
トピック

- センサーモニタリングのチュートリアルを試してみる
- Sidewalk デバイスのオンボーディングの概要

センサーモニタリングのチュートリアルを試してみる

このセクションでは、センサーの温度をモニタリングする方法を示す GitHub の Amazon Sidewalk サンプルアプリケーションの概要を説明します。このチュートリアルでは、必要なワイヤレスリソースをプログラムで作成し、エンドデバイスをプロビジョニングしてバイナリをフラッシュし、エンドデバイスをアプリケーションに接続するスクリプトを使用します。AWS CLI と Python コマンドを使用するスクリプトは、AWS CloudFormation スタックとワイヤレスリソースを作成し、バイナリをフラッシュしてアプリケーションを Hardware Development Kit (HDK) にデプロイします。

次の図は、サンプルアプリケーションを実行し、Sidewalk のエンドデバイスをアプリケーションに接続する場合の手順を示しています。このチュートリアルの前提条件や設定を含む詳細な手順については、GitHub の「README ドキュメント」を参照してください。



Sidewalk デバイスのオンボーディングの概要

このセクションでは、Sidewalk エンドデバイスを AWS IoT Core for Amazon Sidewalk にオンボードする方法について説明します。デバイスをオンボードするには、まず Sidewalk デバイスを追加し、次にデバイスをプロビジョニングして登録し、ハードウェアをクラウドアプリケーションに接続します。このチュートリアルを実行する前に、Python および AWS CLI のインストール を確認して完了してください。

以下の手順は、Sidewalk のエンドデバイスを AWS IoT Core for Amazon Sidewalk にオンボードして接続する方法を示しています。AWS CLI を使用してデバイスをオンボードする場合は、このセクションに記載されているサンプルコマンドを参照してください。AWS IoT コンソールを使用してデバイスをオンボードする方法については、「AWS IoT Core for Amazon Sidewalk への接続」を参照してください。

> **Important**
>
> オンボーディングワークフロー全体を実行するには、エンドデバイスをプロビジョニングして登録し、Hardware Development Kit (HDK) を接続します。詳細については、「Amazon Sidewalk ドキュメント」の「エンドデバイスのプロビジョニングと登録」を参照してください。

トピック

- ステップ 1: AWS IoT Core for Amazon Sidewalk に Sidewalk デバイスを追加する
- ステップ 2: Sidewalk エンドデバイスの送信先の作成
- ステップ 3: エンドデバイスのプロビジョニングと登録
- ステップ 4: Sidewalk デバイスへの接続とメッセージの交換

ステップ 1: AWS IoT Core for Amazon Sidewalk に Sidewalk デバイスを追加する

Sidewalk エンドデバイスを AWS IoT Core for Amazon Sidewalk に追加するために実行する手順の概要を次に示します。デバイスプロファイルと作成したワイヤレスデバイスについて取得した情報を保存します。この情報を使用して、エンドデバイスのプロビジョニングと登録を行います。これらのステップの詳細については、AWS IoT Core for Amazon Sidewalk にデバイスを追加する を参照してください。

1. デバイスプロファイルの作成

   Sidewalk デバイスの共有設定を含むデバイスプロファイルを作成します。プロファイルを作成するときは、プロファイルの *name* を英数字の文字列で指定します。プロファイルを作成するには、AWS IoT コンソールの [プロファイル] ハブの Sidewalk] タブに移動して [プロファイルを作成] を選択するか、この例に示すように CreateDeviceProfile API オペレーションまたは create-device-profile CLI コマンドを使用します。

   ```
   // Add your device profile using a name and the sidewalk object.
   aws iotwireless create-device-profile --name sidewalk_profile --sidewalk {}
   ```

2. Sidewalk のエンドデバイスの作成

   AWS IoT Core for Amazon Sidewalk で Sidewalk のエンドデバイスを作成します。送信先名と、前の手順で取得したデバイスプロファイルの ID を指定します。デバイスを追加するには、AWS IoT コンソールの [デバイス] ハブの Sidewalk タブに移動して [デバイスをプロビジョニング] を選択するか、この例に示すように CreateWirelessDevice API オペレーションまたは create-wireless-device CLI コマンドを使用します。

   > **Note**
   >
   > 送信先には、AWS アカウント と AWS リージョン に固有の名前を指定してください。AWS IoT Core for Amazon Sidewalk に送信先を追加するときには、同じ送信先名を使用します。

   ```
   // 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. デバイスプロファイルとワイヤレスデバイス情報を取得する

   デバイスプロファイルとワイヤレスデバイス情報を JSON として取得します。JSON には、デバイスの詳細、デバイス証明書、プライベートキー、DeviceTypeId、および Sidewalk 製造シリアル番号 (SMSN) に関する情報が含まれます。

   - AWS IoT コンソールを使用している場合は、[デバイス] ハブの [Sidewalk] タブを使用して、Sidewalk エンドデバイス用の結合された JSON ファイルをダウンロードできます。

- API オペレーションを使用している場合は、API オペレーション GetDeviceProfile や GetWirelessDevice から取得したレスポンスを、*device_profile.json* や *wireless_device.json* などの個別の JSON ファイルとして保存します。

```
// 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
```

ステップ 2: Sidewalk エンドデバイスの送信先の作成

AWS IoT Core for Amazon Sidewalk に送信先を追加するために実行する手順の概要は次のとおりです。AWS Management Console、または AWS IoT Wireless API オペレーション、もしくは AWS CLI を使用して、次の手順を実行して AWS IoT ルールと送信先を作成します。その後、ハードウェアプラットフォームに接続し、メッセージを表示および交換できます。このセクションの AWS CLI の例で使用されている IAM ロールと AWS IoT ルールのサンプルについては、「送信先の IAM ロールと IoT ルールを作成する」を参照してください。

1. IAM ロールの作成

   データを AWS IoT ルールに送信するための AWS IoT Core for Amazon Sidewalk アクセス許可を付与する IAM ロールを作成します。このロールを作成するには、CreateRole API オペレーションまたは create-role CLI コマンドを使用します。ロールには *SidewalkRole* という名前を付けることができます。

```
aws iam create-role --role-name lambda-ex \
    --assume-role-policy-document file://lambda-trust-policy.json
```

2. 送信先のルールを作成する

   デバイスのデータを処理し、メッセージを公開するトピックを指定する AWS IoT ルールを作成します。ハードウェアプラットフォームに接続すると、このトピックに関するメッセージが表示されます。AWS IoT Core API オペレーション CreateTopicRule、または AWS CLI コマンド create-topic-rule を使用して、送信先のルールを作成します。

```
aws iot create-topic-rule --rule-name Sidewalkrule \
```

```
    --topic-rule-payload file://myrule.json
```

3. 送信先を作成する

   Sidewalk デバイスを、他の AWS のサービス で使用できるように処理する IoT ルールに関連付ける送信先を作成します。AWS IoT コンソールの [送信先] ハブ、CreateDestination API オペレーション、または create-destination CLI コマンドを使用して送信先を追加できます。

```
aws iotwireless create-destination --name SidewalkDestination \
    --expression-type RuleName --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

ステップ 3: エンドデバイスのプロビジョニングと登録

Python コマンドを使用して、エンドデバイスをプロビジョニングして登録できます。プロビジョニングスクリプトは、取得したデバイスの JSON データを使用して製造用バイナリイメージを生成し、ハードウェアボードにフラッシュします。次に、ハードウェアプラットフォームに接続するためのエンドデバイスを登録します。詳細については、「Amazon Sidewalk ドキュメント」の「エンドデバイスのプロビジョニングと登録」を参照してください。

> **Note**
>
> Sidewalk エンドデバイスを登録する際には、ゲートウェイが Amazon Sidewalk にオプトインしていて、ゲートウェイとデバイスがお互いの通信範囲内にある必要があります。

ステップ 4: Sidewalk デバイスへの接続とメッセージの交換

エンドデバイスを登録したら、エンドデバイスを接続してメッセージやデバイスデータの交換を開始できます。

1. Sidewalk エンドデバイスの接続

   HDK をコンピュータに接続し、ベンダーのマニュアルに記載されている指示に従って HDK に接続します。詳細については、「Amazon Sidewalk ドキュメント」の「エンドデバイスのプロビジョニングと登録」を参照してください。

2. メッセージの表示と交換

   MQTT クライアントを使用して、ルールで指定されたトピックをサブスクライブし、受信したメッセージを表示します。SendDataToWirelessDevice API オペレーションまたは send-data-to-wireless-device CLI コマンドを使用して、デバイスにダウンリンクメッセージを送信し、接続ステータスを確認することもできます。

   (オプション) メッセージ配信ステータスイベントを有効にして、ダウンリンクメッセージが正常に受信されたかどうかを確認できます。

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

AWS IoT Core for Amazon Sidewalk への接続

このセクションでは、Sidewalk エンドデバイスをオンボーディングし、デバイスを Sidewalk ネットワークに接続する方法について説明します。Sidewalk デバイスのオンボーディングの概要 で説明したように、オンボーディングチュートリアルで実行する手順について説明されています。AWS IoT コンソールと AWS IoT Core for Amazon Sidewalk の API オペレーションを使用してデバイスをオンボーディングする方法を学びます。また、これらのオペレーションを実行する AWS CLI コマンドについても学びます。

前提条件

エンドデバイスと送信先を AWS IoT Core for Amazon Sidewalk に追加するには、AWS アカウントをセットアップする必要があります。AWS IoT Wireless API または AWS CLI コマンドを使用してこれらのオペレーションを実行するには、AWS CLI もセットアップする必要があります。前提条件および設定の詳細については、「Python および AWS CLI のインストール」を参照してください。

> **Note**
>
> エンドデバイスのプロビジョニングと登録、Hardware Development Kit (HDK) への接続に関するオンボーディングワークフロー全体を実行するには、Sidewalk ゲートウェイと HDK もセットアップする必要があります。詳細については、「Amazon Sidewalk ドキュメント」の

「Hardware Development Kit (HDK) のセットアップ」と「Sidewalk ゲートウェイのセットアップ」を参照してください。

Sidewalk リソースについて説明する

作業を開始してリソースを作成する前に、Sidewalk のエンドデバイス、デバイスプロファイル、および送信先の命名規則を検討してください。AWS IoT Core for Amazon Sidewalk は、作成したリソースに一意の識別子を割り当てます。ただし、わかりやすい名前を付けたり、説明を追加したり、識別や管理に役立つオプションのタグを追加したりできます。

> **Note**
>
> 送信先の名前は、作成後に変更することはできません。自分の AWS アカウント と AWS リージョン で固有の名前を使用してください。

詳細については、「AWS IoT Wireless リソースについて説明する」を参照してください。

トピック

- AWS IoT Core for Amazon Sidewalk にデバイスを追加する
- Sidewalk エンドデバイスの送信先を追加する
- Sidewalk デバイスを接続してアップリンクメタデータ形式を表示する

AWS IoT Core for Amazon Sidewalk にデバイスを追加する

ワイヤレスデバイスを作成する前に、まずデバイスプロファイルを作成します。デバイスプロファイルは、Sidewalk デバイスのデバイス機能やその他のパラメータを定義します。1 つのデバイスプロファイルを複数のデバイスに関連付けることができます。

デバイスプロファイルを作成した後、プロファイルに関する情報を取得すると、`DeviceTypeId` が返されます。エンドデバイスをプロビジョニングするときは、この ID、デバイス証明書、アプリケーションサーバーの公開キー、および SMSN を使用します。

デバイスを作成して追加する方法

1. Sidewalk エンドデバイス用のデバイスプロファイルを作成します。Sidewalk デバイスに使用するプロファイル名を英数字文字列として指定します。プロファイルは、関連付けるデバイスを識別するのに役立ちます。
   - (コンソール) Sidewalk デバイスを追加するときに、新しいプロファイルを作成することもできます。これにより、AWS IoT Core for Amazon Sidewalk にデバイスを簡単に追加してプロファイルと関連付けることができます。
   - (API) プロファイル名と Sidewalk オブジェクト、`sidewalk {}` を指定して `CreateDeviceProfile` API オペレーションを使用します。API レスポンスには、プロファイル ID と ARN (Amazon リソースネーム) が含まれます。
2. AWS IoT Core for Amazon Sidewalk にワイヤレスデバイスを追加します。送信先の名前を指定して、上記の手順で作成したデバイスプロファイルを選択します。
   - (コンソール) Sidewalk デバイスを追加するときに、送信先名を入力し、作成したプロファイルを選択します。
   - (API) `CreateWirelessDevice` API オペレーションを使用します。送信先名と、以前に取得したデバイスプロファイルの ID を指定します。

   ワイヤレスデバイスのパラメータ

パラメータ	説明	メモ
送信先名	他の AWS のサービス ユーザーが使用するデバイスのデータを処理するための AWS IoT ルールを説明する送信先の名前。	送信先をまだ作成していない場合は、任意の文字列値を指定できます。AWS IoT Core for Amazon Sidewalk は、デバイスの作成時に空の送信先を作成し、送信先を追加するときに更新できます。
デバイスプロファイル	以前に作成したデバイスプロファイル。	–

3. エンドデバイスのプロビジョニングに必要な情報を含む JSON ファイルを取得します。
   - (コンソール) 作成した Sidewalk デバイスの詳細ページからこのファイルをダウンロードします。
   - (API) `GetDeviceProfile` および `GetWirelessDevice` API オペレーションを使用して、デバイスプロファイルとワイヤレスデバイスに関する情報を取得します。API レスポンス情報

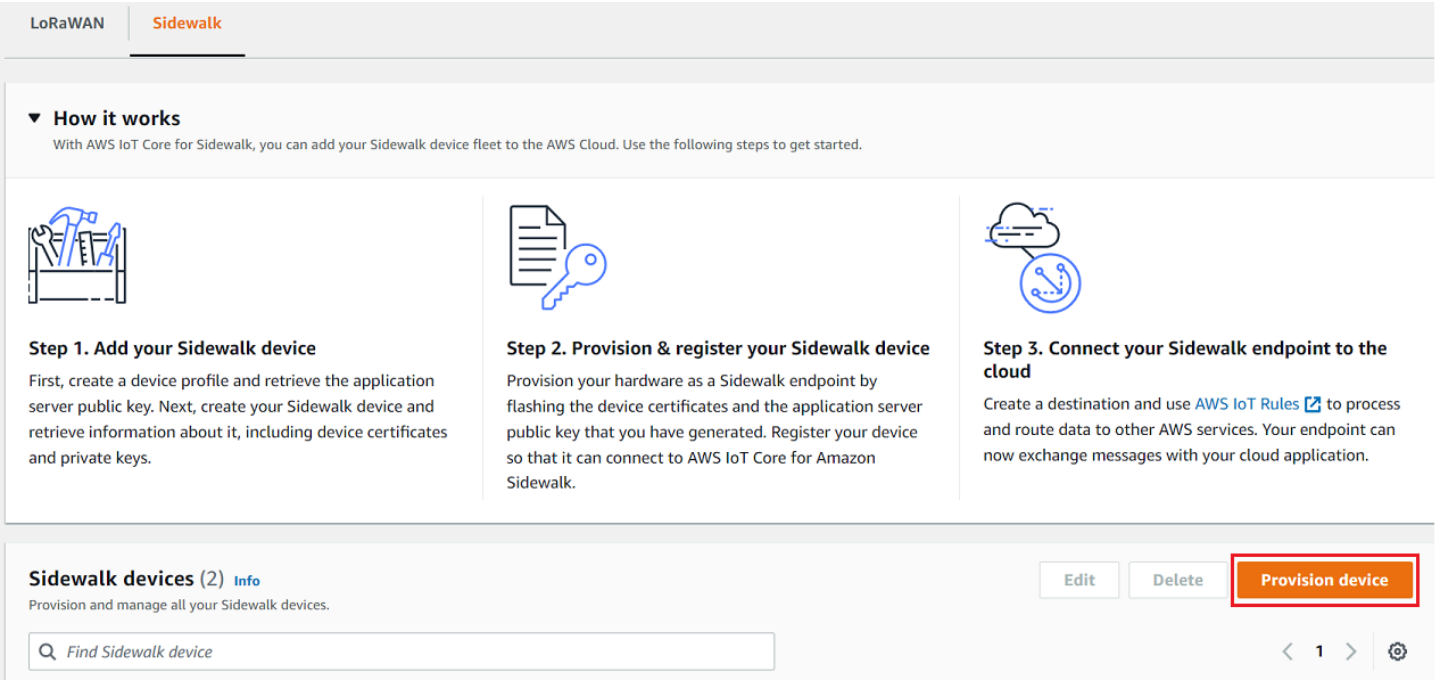
を、*device_profile.json* および *wireless_device.json* などの JSON ファイルとして保存します。

デバイスプロファイルと Sidewalk エンドデバイスを追加します

このセクションでは、デバイスプロファイルを作成する方法を説明します。また、AWS IoT コンソールと AWS CLI を使用して Sidewalk エンドデバイスを AWS IoT Core for Amazon Sidewalk に追加する方法についても説明します。

Sidewalk デバイスを追加する (コンソール)

AWS IoT コンソールを使用して Sidewalk デバイスを追加するには、[デバイスハブの Sidewalk タブ] に移動し、[デバイスのプロビジョニング] を選択して、次の手順を実行します。



1. デバイスの詳細を指定する

   Sidewalk デバイスの設定情報を指定します。新しいデバイスプロファイルを作成するか、Sidewalk デバイス用の既存のプロファイルを選択することもできます。

   a. デバイス名とオプションの説明を指定します。説明は最大 2,048 文字とすることができます。これらのフィールドは、デバイスの作成後に編集できます。

   b. Sidewalk デバイスに関連付けるデバイスプロファイルを選択します。既存のデバイスプロファイルがある場合は、プロファイルを選択できます。新しいプロファイルを作成する場合は、[プロファイルの作成] を選択し、プロファイルの名前を入力します。

> **Note**
>
> デバイスプロファイルにタグを付けるには、プロファイルを作成した後、[プロファイルハブ] に移動し、プロファイルを編集してこの情報を追加します。

c. デバイスから他の AWS のサービス にメッセージをルーティングする送信先の名前を指定します。まだ送信先を作成していない場合は、[送信先ハブ] に移動して送信先を作成してください。その後、Sidewalk デバイスの送信先を選択できます。詳細については、「Sidewalk エンドデバイスの送信先を追加する」を参照してください。

d. Sidewalk デバイスの追加を続行するには、[次へ] を選択します。

2. Sidewalk デバイスを AWS IoT モノと関連付ける (オプション)

   オプションで Sidewalk デバイスを AWS IoT モノに関連付けることができます。IoT モノは、AWS IoT デバイスレジストリのエントリです。モノを使用すると、デバイスの検索と管理を簡単に行えるようになります。モノをデバイスに関連付けると、デバイスから他の AWS IoT Core 機能にアクセスできます。

   デバイスをモノに関連付けるには、[モノの自動登録] を選択します。

   a. Sidewalk デバイスに関連付けたい IoT モノの固有の名前を入力します。モノの名前は大文字と小文字が区別され、AWS アカウント と AWS リージョン で一意でなければなりません。

   b. モノの種類や、モノのリストから絞り込むために使用できる検索可能な属性など、IoT モノに追加の設定があれば提供してください。

   c. [次へ] を選択し、Sidewalk デバイスに関する情報を確認してから、[作成] を選択します。

Sidewalk デバイスを追加する (CLI)

Sidewalk デバイスを追加し、Sidewalk デバイスのプロビジョニングに使用する JSON ファイルをダウンロードするには、次の API オペレーションを実行します。

トピック

- ステップ 1: デバイスプロファイルを作成する
- ステップ 2: Sidewalk デバイスを追加する

ステップ 1: デバイスプロファイルを作成する

AWS アカウント でデバイスプロファイルを作成するには、CreateDeviceProfile API オペレーションまたは create-device-profile CLI コマンドを使用します。デバイスプロファイルを作成するときは、名前を指定し、任意のタグを名前と値のペアとして指定します。

例えば、次のコマンドは Sidewalk デバイス用のデバイスプロファイルを作成します。

```
aws iotwireless create-device-profile \
    --name sidewalk_profile --sidewalk {}
```

このコマンドを実行すると、Amazon リソースネーム (ARN) とデバイスプロファイルの ID が出力として返されます。

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

ステップ 2: Sidewalk デバイスを追加する

Sidewalk デバイスを AWS IoT Core for Amazon Sidewalk のアカウントに追加するには、CreateWirelessDevice API オペレーションまたは create-wireless-device CLI コマンドを使用します。デバイスを作成するときは、Sidewalk デバイスのオプションの名前と説明に加えて、次のパラメータを指定します。

> **Note**
>
> Sidewalk デバイスを AWS IoT モノと関連付ける場合は、AssociateWirelessDeviceWithThing API オペレーションまたは associate-wireless-device-with-thing CLI コマンドを使用してください。

以下のコマンドは、Sidewalk デバイスの作成例を示しています。

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

以下は、device.json ファイルの内容を示しています。

device.json の内容

```
{
  "Type": "Sidewalk",
  "Name": "SidewalkDevice",
  "DestinationName": "SidewalkDestination",
  "Sidewalk": {
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
    }
}
```

このコマンドを実行すると、デバイス ID と Amazon リソースネーム (ARN) が出力として返されます。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/23456789-abcd-0123-bcde-fabc012345678",
    "Id": "23456789-abcd-0123-bcde-fabc012345678"
}
```

プロビジョニング用のデバイス JSON ファイルを取得する

Sidewalk デバイスを AWS IoT Core for Amazon Sidewalk に追加したら、エンドデバイスのプロビジョニングに必要な情報を含む JSON ファイルをダウンロードします。この情報は、AWS IoT コンソールまたは AWS CLI を使用して取得できます。デバイスのプロビジョニング方法の詳細については、「Amazon Sidewalk ドキュメント」の「エンドデバイスのプロビジョニングと登録」を参照してください。

JSON ファイルを取得する (コンソール)

Sidewalk デバイスをプロビジョニングするための JSON ファイルを取得するには:

1. Sidewalk デバイスハブに移動します。
2. AWS IoT Core for Amazon Sidewalk に追加したデバイスを選択して詳細を表示します。
3. 追加したデバイスの詳細ページで [デバイスの JSON ファイルをダウンロード] を選択して JSON ファイルを取得します。

   エンドデバイスのプロビジョニングに必要な情報を含む `certificate.json` ファイルをダウンロードします。以下は JSON ファイルのサンプルです。これには、デバイス証明書、プライベートキー、Sidewalk 製造シリアル番号 (SMSN)、および `DeviceTypeID` が含まれています。

```
{
  "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"
}
```

Sidewalk デバイスの詳細ページには、次の情報も表示されます。

- デバイス ID、Amazon リソースネーム (ARN)、および送信先の AWS IoT Amazon リソースネーム (ARN)、および送信先の情報。
- デバイスプロファイルと送信先の詳細。
- デバイスから最後のアップリンクメッセージを受信した時刻。
- デバイスがプロビジョニング済みか登録済みかを示すステータス。

JSON ファイルを取得する (CLI)

AWS IoT Core for Amazon Sidewalk の API または AWS CLI を使用して Sidewalk エンドデバイスをプロビジョニングするための JSON ファイルを取得するには、デバイスプロファイルとワイヤレスデバイスに関する情報を取得した API レスポンスを JSON ファイル (*wireless_device.json* および *device_profile.json* など) として一時的に保存します。Sidewalk デバイスのプロビジョニングに使用します。

JSON ファイルを取得する方法は次のとおりです。

トピック

- ステップ 1: デバイスプロファイル情報を JSON ファイルとして取得する
- ステップ 2: Sidewalk デバイス情報を JSON ファイルとして取得する

ステップ 1: デバイスプロファイル情報を JSON ファイルとして取得する

GetDeviceProfile API オペレーションまたは get-device-profile CLI コマンドを使用して、AWS IoT Core for Amazon Sidewalk のアカウントに追加したデバイスプロファイルに関する情報を取得します。デバイスプロファイルに関する情報を取得するには、プロファイル ID を指定します。

その後、API は指定された識別子とデバイス ID に一致するデバイスプロファイルに関する情報を返します。このレスポンス情報をファイルとして保存し、*device_profile.json* のような名前を付けます。

CLI コマンドの例を以下に示します。

```
aws iotwireless get-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" > device_profile.json
```

このコマンドを実行すると、デバイスプロファイルのパラメータ、アプリケーションサーバの公開キー、および DeviceTypeID が返されます。以下は、API からのサンプルレスポンス情報を含む JSON ファイルを示しています。API レスポンスのパラメータの詳細については、「GetDeviceProfile」を参照してください。

GetDeviceProfile API レスポンス (***device_profile.json*** の内容)

```
{
    "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
    }
}
```

ステップ 2: Sidewalk デバイス情報を JSON ファイルとして取得する

GetWirelessDevice API オペレーションまたは get-wireless-device CLI コマンドを使用して、AWS IoT Core for Amazon Sidewalk のアカウントに追加した Sidewalk デバイスに関する情報を取得します。エンドデバイスに関する情報を取得するには、デバイスを追加したときに取得したワイヤレスデバイスの識別子を指定します。

その後、API は指定された識別子とデバイス ID に一致するデバイスに関する情報を返します。このレスポンス情報を JSON ファイルとして保存します。*wireless_device.json* など、ファイルにはわかりやすい名前を使用します。

以下の例は、CLI を使用するコマンドを実行する例を示しています。

```
aws iotwireless get-wireless-device --identifier-type WirelessDeviceId \
    --identifier "23456789-abcd-0123-bcde-fabc012345678" > wireless_device.json
```

このコマンドを実行すると、デバイスの詳細、デバイス証明書、プライベートキー、Sidewalk の製造シリアル番号 (SMSN) が返されます。次は、このコマンドを実行したときの出力の例を示しています。API レスポンスのパラメータの詳細については、「GetWirelessDevice」を参照してください。

GetWirelessDevice API レスポンス (***wireless_device.json*** の内容)

```
{
    "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"
    },

    ...

}
```

次のステップ

JSON ファイル、*wireless_device.json*、*device_profile.json* を一時的に保存し、次のステップでそれらを使用してハードウェアプラットフォームに接続するためのエンドデバイスのプロビジョニングと登録を行います。詳細については、「Amazon Sidewalk ドキュメント」の「エンドデバイスのプロビジョニングと登録」を参照してください。

Sidewalk エンドデバイスの送信先を追加する

AWS IoT ルールを使用してデータやデバイスメッセージを処理し、他のサービスにルーティングします。また、デバイスから受信したバイナリメッセージを処理するルールを定義して、メッセージを他の形式に変換し、他のサービスで使用しやすくすることもできます。送信先は、デバイスのメッセージデータを処理して他の AWS のサービス に送信するルールに Sidewalk エンドデバイスを関連付けます。

送信先を作成して使用する方法

1. AWS IoT ルールと送信先の IAM ロールを作成します。この AWS IoT ルールは、デバイスのデータを処理し、他の AWS のサービス やアプリケーションが使用できるようにルーティングするルールを指定します。IAM ロールは、ルールへのアクセス許可を付与します。
2. `CreateDestination` API オペレーションを使用して Sidewalk デバイスの送信先を作成します。送信先名、ルール名、ロール名、および任意のパラメータを指定します。API は送信先の一意の識別子を返します。この識別子は、エンドデバイスを AWS IoT Core for Amazon Sidewalk に追加するときに指定できます。

以下に、送信先を作成する方法と、送信先の AWS IoT ルールと IAM ロールを示します。

トピック

- Sidewalk デバイスの送信先を作成する
- 送信先の IAM ロールと IoT ルールを作成する

Sidewalk デバイスの送信先を作成する

AWS IoT Core for Amazon Sidewalk のアカウントに送信先を追加するには、送信先ハブを使用するか、`CreateDestination` を使用します。送信先を作成するときは、以下を指定してください。

- Sidewalk エンドデバイスに使用する送信先の一意な名前。

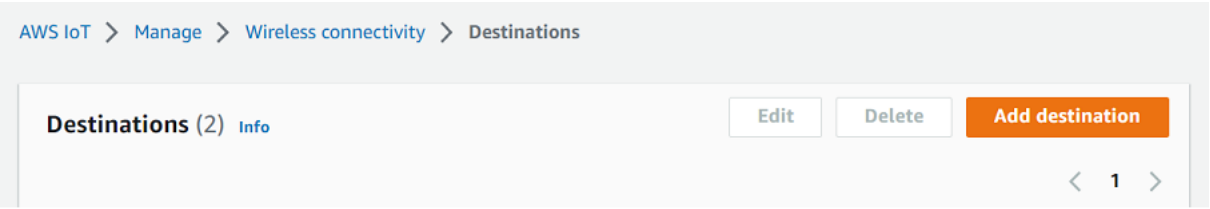
> **Note**
>
> 既に送信先名を使用してデバイスを追加している場合は、送信先を作成するときにその名前を使用する必要があります。詳細については、「ステップ 2: Sidewalk デバイスを追加する」を参照してください。

- デバイスのデータを処理する AWS IoT ルールの名前と、メッセージが公開されるトピック。
- ルールにアクセスするための許可をデバイスのデータに付与する IAM ロール。

以下のセクションに、送信先の AWS IoT ルールと IAM ロールを作成する方法示します。

送信先を作成する (コンソール)

AWS IoT コンソールを使用して送信先を作成するには、[送信先ハブ] に移動し、[送信先を追加] を選択します。



デバイスのデータを処理するには、送信先を作成するときに以下のフィールドを指定して、[送信先を追加] を選択します。

- 送信先の詳細

  [Destination name] (送信先名) と、必要に応じて送信先の説明を入力します。

- ルール名

  デバイスが送信したメッセージを評価し、デバイスのデータを処理するために設定された AWS IoT ルールです。ルール名は送信先にマップされます。送信先には、受信したメッセージを処理す

るためのルールが必要です。メッセージは、AWS IoT ルールの呼び出し、または AWS IoT メッセージブローカーへのパブリッシュのいずれかによって処理されるように選択できます。

- [Enter a rule name] (ルール名を入力) を選択する場合は、名前を入力し、次に [Copy] (コピー) をクリックして、AWS IoT ルールを作成するときに入力するルール名をコピーします。[Create rule] (ルールの作成) を選択して今すぐルールを作成するか、AWS IoT コンソールの [Rules] (ルール) ハブを開き、その名前のルールを作成します。

  ルールを入力し、[Advanced] (アドバンスト) 設定を使用してトピック名を指定することもできます。トピック名はルールの呼び出し中に提供され、ルール内の `topic` 式を使用してアクセスします。AWS IoT のルールの詳細については、「AWS IoT のルール」を参照してください。
- [Publish to AWS IoT message broker] (IoT メッセージブローカーに発行) を選択する場合は、トピック名を入力します。その後、MQTT トピック名をコピーできます。また、複数のサブスクライバーがこのトピックにサブスクライブして、そのトピックに発行されたメッセージを受信できます。詳細については、「MQTT のトピック」を参照してください。

送信先の AWS IoT ルールの詳細については、「LoRaWAN デバイスメッセージを処理するルールを作成する」を参照してください。

- ロール名

  [Rule name] (ルール名) で名前を付けたルールにアクセスするための許可をデバイスのデータに付与する IAM ロール。コンソールでは、新しいサービスロールを作成する、または既存のサービスロールを選択することができます。新しいサービスロールを作成する場合は、ロール名 (例、**`SidewalkDestinationRole`**) を入力するか、あるいは、空白のままにして AWS IoT Core for LoRaWAN で新しいロール名を作成することできます。その後、適切なアクセス許可を持つ IAM ロールが AWS IoT Core for LoRaWAN で自動的に作成されます。

送信先を作成する (CLI)

送信先を作成するには、`CreateDestination` API オペレーションまたは `create-destination` CLI コマンドを使用します。例えば、次のコマンドは Sidewalk エンドデバイス用の送信先を作成します。

```
aws iotwireless create-destination --name SidewalkDestination \
    --expression-type RuleName --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

このコマンドを実行すると、Amazon リソースネーム (ARN) や送信先の名前を含む、送信先の詳細が返されます。

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/SidewalkDestination",
    "Name": "SidewalkDestination"
}
```

送信先の作成に関する詳細については、「LoRaWAN デバイスメッセージを処理するルールを作成する」を参照してください。

送信先の IAM ロールと IoT ルールを作成する

AWS IoT ルールは、デバイスメッセージを他のサービスに送信します。AWS IoT ルールでは、Sidewalk エンドデバイスから受信したバイナリメッセージを処理して、他のサービスで使用しやすくすることもできます。AWS IoT Core for Amazon Sidewalk の送信先は、デバイスのメッセージデータを処理して他のサービスに送信するルールにワイヤレスデバイスを関連付けます。このルールは、AWS IoT Core for Amazon Sidewalk がデバイスのデータを受信するとすぐにそのデータに対してアクションを実行します。データを同じサービスに送信するすべてのデバイスについて、すべての送信先を作成して、すべての送信先を作成できます。ルールにデータを送信するためのアクセス許可を付与する IAM ロールも作成する必要があります。

送信先の IAM ロールを作成する

データを AWS IoT ルールに送信するための AWS IoT Core for Amazon Sidewalk アクセス許可を付与する IAM ロールを作成します。このロールを作成するには、`CreateRole` API オペレーションまたは `create-role` CLI コマンドを使用します。ロールには *SidewalkRole* という名前を付けることができます。

```
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"}]}'
```

また、JSON ファイルを使用してロールの信頼ポリシーを定義することもできます。

```
aws iam create-role --role-name SidewalkRole \
    --assume-role-policy-document file://trust-policy.json
```

以下は、JSON ファイルの内容を示しています。

trust-policy.json の内容

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

送信先のルールを作成する

ルールを作成するには、AWS IoT Core API オペレーション、CreateTopicRule、または AWS CLI コマンド、create-topic-rule を使用します。トピックルールは、送信先が Sidewalk エンドデバイスから受信したデータを他の AWS のサービス にルーティングするために使用されます。例えば、Lambda 関数にメッセージを送信するルールアクションを作成できます。デバイスからアプリケーションデータを受け取り、base64 を使用してペイロードデータをデコードして他のアプリケーションで使用できるように Lambda 関数を定義できます。

以下の手順は、Lambda 関数を作成する方法と、この関数にメッセージを送信するトピックルールを作成する方法を示しています。

1. 実行ロールとポリシーを作成する

   AWS リソースにアクセスするためのアクセス許可を関数に付与する IAM ロールを作成します。また、JSON ファイルを使用してロールの信頼ポリシーを定義することもできます。

   ```
   aws iam create-role --role-name lambda-ex \
       --assume-role-policy-document file://lambda-trust-policy.json
   ```

   以下は、JSON ファイルの内容を示しています。

   lambda-trust-policy.json の内容

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Lambda 関数を作成およびテストする

   ペイロードデータを base64 でデコードする AWS Lambda 関数を作成するには、次の手順を実行します。

   a. ペイロードデータをデコードするためのコードを記述します。例えば、以下の Python サンプルコードを使用できます。スクリプトの名前を指定します (例: *base64_decode.py*)。

      base64_decode.py の内容

      ```
      // ---------------------------------------------------------
      // ----- 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. Python ファイルを含む zip ファイルとしてデプロイパッケージを作成し、そのパッケージに *base64_decode.zip* という名前を付けます。CreateFunction API または `create-function` CLI コマンドを使用して、サンプルコード *base64_decode.py* の Lambda 関数を作成します。

   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
```

次のような出力が表示されます。トピックルールを作成するときは、出力、FunctionArn の Amazon リソースネーム (ARN) の値を使用します。

```
{
    "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. コマンドラインから呼び出しのログを取得するには、invoke コマンドと共に --log-type オプションを使用します。レスポンスには、LogResult フィールドが含まれます。このフィールドには、呼び出しから base64 コードされた最大 4 KB のログが含まれます。

```
aws lambda invoke --function-name my-function out --log-type Tail
```

StatusCode が 200 のレスポンスが返ってくるはずです。Lambda 関数の作成と使用の詳細については、「AWS CLI で Lambda を使用する」を参照してください。

3. トピックルールを作成する

CreateTopicRule API または create-topic-rule CLI コマンドを使用して、この Lambda 関数にメッセージを送信するトピックルールを作成します。AWS IoT トピックに再公開する 2 つ目のルールアクションを追加することもできます。このトピックルールに「*Sidewalkrule*」という名前を付けてください。

```
aws iot create-topic-rule --rule-name Sidewalkrule \
    --topic-rule-payload file://myrule.json
```

myrule.json ファイルを使用して、ルールに関する詳細を指定できます。例えば、次の JSON ファイルには、AWS IoT トピックを再発行して Lambda 関数にメッセージを送信する方法が示されています。

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

Sidewalk デバイスを接続してアップリンクメタデータ形式を表示する

このチュートリアルでは、MQTT テストクライアントを使用して接続をテストし、エンドデバイスと AWS クラウド との間で交換されるメッセージを確認します。メッセージを受信するには、MQTT テストクライアントで、宛先の IoT ルールを作成するときに指定したトピックに登録します。SendDataToWirelessDevice API オペレーションを使用して、AWS IoT Core for Amazon Sidewalk からデバイスにダウンリンクメッセージを送信することもできます。メッセージ配信ステータスイベント通知を有効にすることで、メッセージが配信されたことを確認できます。

> **Note**
>
> ハードウェアプラットフォームの接続とセットアップの詳細については、「Amazon Sidewalk ドキュメント」の「エンドデバイスのプロビジョニングと登録」と「Hardware Development Kit (HDK) の設定」を参照してください。

エンドデバイスにダウンリンクメッセージを送信する

SendDataToWirelessDevice API オペレーションまたは send-data-to-wireless-device CLI コマンドを使用して、AWS IoT Core for Amazon Sidewalk から Sidewalk エンドデバイスにダウンリンクメッセージを送信します。次のコードは、このコマンドの実行例を示しています。ペイロードデータは送信されるバイナリで、base64 でエンコードされます。

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

次に、このコマンドを実行したときの出力例を示します。これは、デバイスに送信されるダウンリンクメッセージの ID です。

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

> **Note**
>
> SendDataToWirelessDevice API はメッセージ ID を返すことができますが、メッセージは正常に配信されない可能性があります。デバイスに送信されたメッセージのステータスを確認するには、Sidewalk アカウントとデバイスのメッセージ配信ステータスイベントを有効にできます。このイベントを有効にする方法の詳細については、「Sidewalk リソースのイベント通知」を参照してください。このイベントタイプの詳細については、「メッセージ配信イベント」を参照してください。

デバイスから送信されたアップリンクメッセージの形式の表示

デバイスを接続した後、送信先ルール作成時に指定したトピック (例えば、*project/sensor/observed*) をサブスクライブし、デバイスからのアップリンクメッセージを確認することができます。

送信先を作成するときにトピック名を指定した場合、そのトピックをサブスクライブして、エンドデバイスからのアップリンクメッセージをモニタリングできます。AWS IoT コンソールの [テスト] ページの [MQTT テストクライアント] に移動し、トピック名 (*project/sensor/observed* など) を入力して、[サブスクライブ] を選択します。

次の例は、Sidewalk デバイスから AWS IoT に送信されるアップリンクメッセージの形式を示しています。`WirelessMetadata` にはメッセージリクエストに関するメタデータが含まれています。

```
{
   "PayloadData":"ZjRlNjY1ZWNlNw==",
   "WirelessDeviceId":"wireless_device_id",
   "WirelessMetadata":{
      "Sidewalk":{
         "CmdExStatus":"Cmd",
         "SidewalkId":"device_id",
         "Seq":0,
         "MessageType":"messageType"
      }
    }
}
```

以下の表は、アップリンクメタデータの各パラメータの定義を示したものです。*device-id* は、ワイヤレスデバイスの ID (*ABCDEF1234* など) です。*messageType* は、デバイスから受信されるアップリンクメッセージのタイプです。

Sidewalk アップリンクメタデータのパラメータ

パラメータ	説明	タイプ	必須
`PayloadData`	ワイヤレスデバイスから送信されるメッセージペイロード。	文字列	はい
`WirelessDeviceID`	データを送信しているワイヤレスデバイスの識別子	文字列	はい

パラメータ	説明	タイプ	必須
Sidewalk.CmdExStatus	コマンドのランタイムステータス。レスポンスタイプのメッセージには、ステータスコード COMMAND_EXEC_STATUS_SUCCESS が含まれます。ただし、通知にはステータスコードが含まれない場合があります。	一覧表	いいえ
Sidewalk.NackExStatus	レスポンス nack のステータスで、RADIO_TX_ERROR または MEMORY_ERROR となります。	文字列の配列	いいえ

AWS IoT Core for Amazon Sidewalk を使用したデバイスの一括プロビジョニング

一括プロビジョニングを使用すると、多数のエンドデバイスを一括して AWS IoT Core for Amazon Sidewalk にオンボードできます。一括プロビジョニングは、特にファクトリーで多数のデバイスを製造していて、それらのデバイスを AWS IoT にオンボードする場合に役立ちます。デバイスの製造に関する詳細については、「Amazon Sidewalk ドキュメント」の「Amazon Sidewalk デバイスの製造」を参照してください。

以下のトピックでは、一括プロビジョニングの仕組みについて説明します。

- Amazon Sidewalk の一括プロビジョニングのワークフロー

  このトピックでは、一括プロビジョニングの主要な概念とその仕組みについて説明します。また、Sidewalk デバイスを AWS IoT Core for Amazon Sidewalk にインポートするために実行する必要がある手順も示しています。

- ファクトリーサポートによるデバイスプロファイルの作成

  このトピックでは、デバイスプロファイルを作成し、ファクトリーサポートを受ける方法について説明します。また、デバイスの製造後に制御ログを取得するために、YubiHSM キーを取得し、製造元に送信する方法についても学びます。

- インポートタスクを使用した Sidewalk デバイスのプロビジョニング

このトピックでは、インポートタスクを作成および使用して Sidewalk デバイスを一括プロビジョニングする方法を説明します。また、インポートタスクを更新または削除する方法、インポートタスクやタスク内のデバイスのステータスを表示する方法についても説明します。

トピック

Amazon Sidewalk の一括プロビジョニングのワークフロー

以下のセクションでは、一括プロビジョニングの主要な概念とその仕組みについて説明します。一括プロビジョニングには以下のステップが含まれます。

1. AWS IoT Core for Amazon Sidewalk を使用してデバイスプロファイルを作成します。
2. Amazon Sidewalk チームに YubiHSM キーと、ファクトリーサポートによるデバイスプロファイルの更新をリクエストします。
3. YubiHSM キーを製造元に送信して、デバイスの製造後に AWS IoT Core for Amazon Sidewalk が制御ログを取得できるようにします。
4. インポートタスクを作成し、AWS IoT Core for Amazon Sidewalk にオンボードするデバイスのシリアル番号 (SMSN) を指定します。

一括プロビジョニングのコンポーネント

次の概念は、一括プロビジョニングの主なコンポーネントと、それらを Sidewalk デバイスの一括プロビジョニングの一部として使用する方法を示しています。

YubiHSM キー

Amazon は、Sidewalk 製品ごとに 1 つ以上の HSM (ハードウェアセキュリティモジュール) を作成します。各 HSM には YubiHSM キーと呼ばれる固有のシリアル番号がハードウェアモジュールに印刷されています。このキーは「Yubico のウェブページ」から購入できます。

キーは各 HSM に固有で、AWS IoT Core for Amazon Sidewalk で作成する各デバイスプロファイルに関連付けられています。YubiHSM キーを入手するには、Amazon Sidewalk チームにお問い合わせください。YubiHSM キーを製造元に送信すると、Sidewalk デバイスがファクトリーで製造された

後、AWS IoT Core for Amazon Sidewalk はデバイスのシリアル番号を含む制御ログファイルを受信します。次に、この情報を入力された CSV ファイルと比較し、AWS IoT にデバイスをオンボードします。

デバイス認証キー (**DAK**)

Sidewalk エンドデバイスが Sidewalk ネットワークに参加する際には、Sidewalk デバイス証明書を使用してプロビジョニングする必要があります。デバイスのセットアップに使用される証明書には、プライベートデバイス固有の証明書と、Sidewalk 証明書チェーンに対応するパブリックデバイス証明書が含まれます。Sidewalk デバイスが製造されると、YubiHSM はデバイス証明書に署名します。

以下は、デバイス証明書とプライベートキーを含むサンプル JSON ファイルを示しています。詳細については、「プロビジョニング用のデバイス JSON ファイルを取得する」を参照してください。

```
{
  "p256R1": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbH ... DANKkOKoNT3bUGz+/f/pyTE
+xMRdIUBZ1Bw==",
  "eD25519": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbHD ... UiZmntHiUr1GfkTOFMYqRB+Aw==",
  "metadata": {
    "devicetypeid": "fe98",

    ...

    "devicePrivKeyP256R1":
 "3e704bf8d319b3a475179f1d68c60737b28c708f845d0198f2d00d00c88ee018",
    "devicePrivKeyEd25519":
 "17dacb3a46ad9a42d5c520ca5f47f0167f59ce54d740aa13918465faf533b8d0"
   },
   "applicationServerPublicKey":
 "5ce29b89c2e3ce6183b41e75fe54e45f61b8bb320efbdd2abd7aefa5957a316b"
}
```

デバイス認証キー (DAK) は、デバイスプロファイルの作成時に取得するプライベートキーです。これは、Sidewalk の各製品に対して発行される固有の証明書である製品証明書に相当します。Amazon Sidewalk チームに連絡すると、Sidewalk 証明書チェーン、YubiHSM キー、および製品デバイス認証キー (DAK) でプロビジョニングされた HSM が届きます。

デバイスプロファイルも新しいデバイス認証キー (DAK) で更新され、ファクトリーサポートが有効になります。デバイスプロファイルの DAK メタデータ情報では、DAK 名、証明書 ID、ApId (アドバタイズ済み製品 ID)、ファクトリーサポートが有効かどうか、DAK が署名できる署名の最大数などの詳細が提供されます

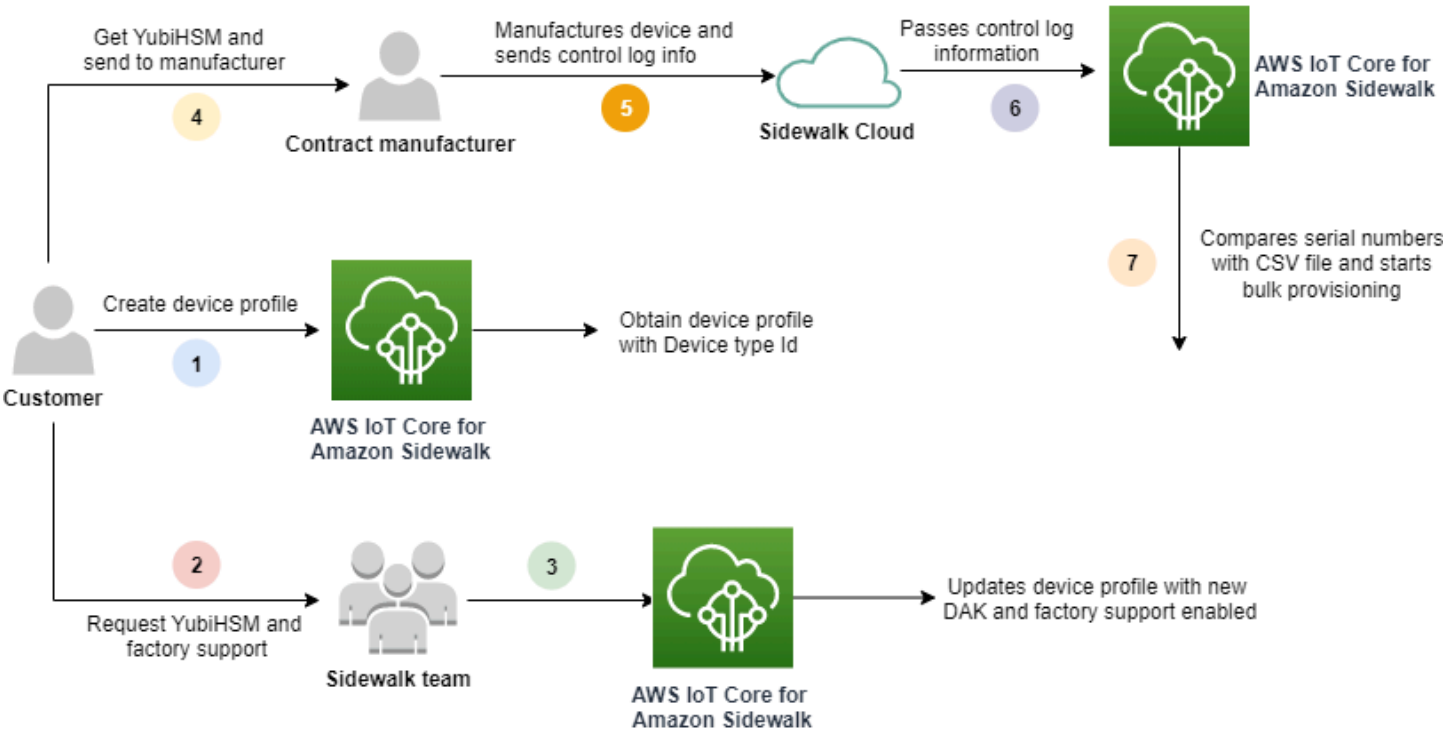
アドバタイズ済み製品 ID (**ApId**)

ApId パラメータは、アドバタイズ済み製品を識別する英数字の文字列です。このフィールドは、一括プロビジョニングする Sidewalk デバイスに特定のデバイスプロファイルを使用する場合に指定する必要があります。AWS IoT Core for Amazon Sidewalk は次に DAK を生成し、YubiHSM キーを通じてユーザーに提供します。関連する DAK 情報は、デバイスプロファイルに表示されます。

ApId を入手するには、作成したデバイスプロファイルに関する情報を取得した後、Amazon Sidewalk サポ - トチームにお問い合わせください。デバイスプロファイル情報は、AWS IoT コンソールから、または、GetDeviceProfile API 操作、または get-device-profile CLI コマンドを使用して取得できます。

一括プロビジョニングの仕組み

このフローチャートは、AWS IoT Core for Amazon Sidewalk で、一括プロビジョニングがどのように機能するかを示しています。



次の手順は、一括プロビジョニングプロセスのさまざまなステップを示しています。

1. Sidewalk デバイス用のデバイスプロファイルを作成する

   エンドデバイスをファクトリーに持ち込む前に、まずデバイスプロファイルを作成します。デバイスプロファイルと Sidewalk エンドデバイスを追加します で説明されているように、このプロファイルを使用して、個々のデバイスをプロビジョニングできます。

2. プロファイルのファクトリーサポートをリクエストする

   エンドデバイスをファクトリーに持ち込む準備ができたら、Amazon Sidewalk チームに YubiHSM キーとデバイスプロファイルのファクトリーサポートを依頼します。

3. DAK およびファクトリーサポートされているプロファイルを取得する

   その後、Amazon Sidewalk サポートチームが、製品のデバイス認証キー (DAK) とファクトリーサポートを使用して、お客様のデバイスプロファイルを更新します。デバイスプロファイルは、アドバタイズ済み製品 ID (ApID)、および新しい DAKと証明書 ID などの証明書情報で自動的に更新されます。このプロファイルを使用する Sidewalk デバイスは、一括プロビジョニングでの使用に適しています。

4. YubiHSM キーを製造元 (CM) に送信する

   これでエンドデバイスの認定が完了したので、YubiHSM キーを委託製造元 (CM) に送信して製造プロセスを開始できます。詳細については、「Amazon Sidewalk ドキュメント」の「Amazon Sidewalk デバイスの製造」を参照してください。

5. デバイスを製造し、制御ログとシリアル番号を送信する

   CM はデバイスを製造し、制御ログを生成します。CM は、製造するデバイスのリストとその Sidewalk 製造シリアル番号 (SMSN) を含む CSV ファイルも提供します。以下のコードは、制御ログの例を示しています。デバイスのシリアル番号、APID、およびデバイスのパブリック証明書が含まれています。

```
{
    "controlLogs": [
    {
        "version": "4-0-1",
        "device":
        {
            "serialNumber": "device1",
            "productIdentifier": {
                "advertisedProductId": "abCD"
             },
             "sidewalkData": {
                "SidewalkED25519CertificateChain": "...",
                "SidewalkP256R1CertificateChain": "..."
             }
         }
      }
   ]
```

```
}
```

6. AWS IoT Core for Amazon Sidewalk に制御ログ情報を渡す

   Amazon Sidewalk クラウドは、製造元から制御ログ情報を取得し、この情報を AWS IoT Core for Amazon Sidewalk に渡します。その後、デバイスをシリアル番号を使って作成することができます。

7. シリアル番号が一致することを確認し、一括プロビジョニングを開始する

   AWS IoT コンソールまたは AWS IoT Core for Amazon Sidewalk の API オペレーション `StartWirelessDeviceImportTask` を使用すると、AWS IoT Core for Amazon Sidewalk は、Amazon Sidewalk から取得した各デバイスの Sidewalk 製造シリアル番号 (SMSN) を、CSV ファイル内の対応するシリアル番号と比較します。この情報が一致すると、一括プロビジョニングプロセスが開始され、AWS IoT Core for Amazon Sidewalk にインポートするデバイスが作成されます。

ファクトリーサポートによるデバイスプロファイルの作成

Amazon Sidewalk デバイスを一括プロビジョニングする前に、デバイスプロファイルを作成し、Amazon Sidewalk サポートチームに連絡してファクトリーサポートをリクエストする必要があります。その後、Amazon Sidewalk チームがお客様のデバイスプロファイルを新しいデバイス認証キー (DAK) で更新し、ファクトリーサポートを追加します。このプロファイルを使用する Sidewalk デバイスは、AWS IoT Core for Amazon Sidewalk での使用を認定され、一括プロビジョニングのためにオンボードできるようになります。

次の手順で、ファクトリーサポートを受けるデバイスプロファイルを作成する方法を示します。

1. デバイスプロファイルを作成する

   まず、デバイスプロファイルを作成します。プロファイルを作成するときは、名前とオプションのタグを名前と値のペアとして指定します。必要なパラメータおよびプロファイルの作成と使用の詳細については、「デバイスを作成して追加する方法」を参照してください。

2. プロファイルのファクトリーサポートを取得する

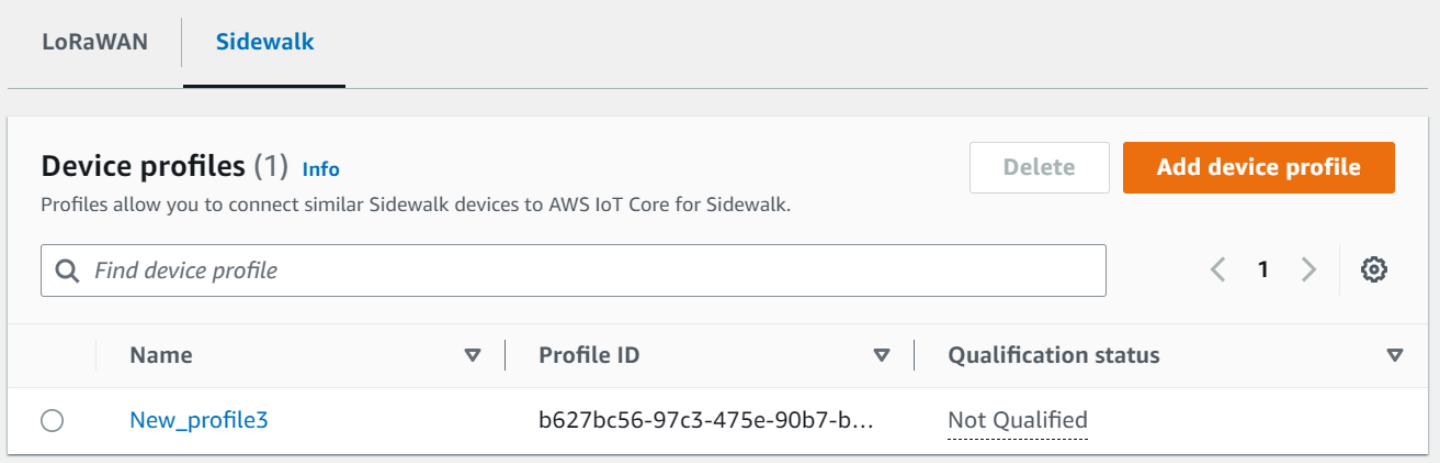
   次に、デバイスプロファイルのファクトリーサポートを取得すると、このプロファイルを使用するデバイスが認定されます。認定を受けるには、Amazon Sidewalk チームでチケットを作成します。チームが確認すると、ApId (アドバタイズ済み製品 ID) が届き、ファクトリー発行の DAK でプロフィールが更新されます。このプロファイルを使用する Sidewalk エンドデバイスが認定されます。

デバイスプロファイルは、AWS IoT コンソール、AWS IoT Core for Amazon Sidewalk の API オペレーション、または AWS CLI のいずれかを使用して作成できます。

トピック

プロファイルの作成 (コンソール)

AWS IoT コンソールを使用してデバイスプロファイルを作成するには、プロファイルハブの Sidewalk タブに移動し、[プロファイルの作成] を選択します。



プロファイルを作成するには、次のフィールドを指定し、[送信] を選択します。

- 名前

  プロファイルの[名前]を入力します。
- タグ

  プロファイルの識別が容易になるように、オプションのタグを名前と値のペアとして入力します。また、タグを使用すると請求を簡単に追跡できます。

プロファイル情報を表示してプロファイルを認定する

「プロファイルハブ」で作成したプロファイルが表示されます。プロファイルを選択して、その詳細を表示します。次の情報が表示されます。

- デバイスプロファイル名と一意の識別子、および名前と値のペアとして指定したオプションのタグ。
- プロファイルのアプリケーションサーバーのパブリックキーとデバイスタイプ ID。
- 認定ステータス。ファクトリーサポートされていないデバイスプロファイルを使用していることを示します。デバイスプロファイルの認定を受け、ファクトリーサポートされるようにするには、Amazon Sidewalk サポートにお問い合わせください。
- デバイス認証キー (DAK) 情報。デバイスプロファイルが認定されると、新しい DAK が発行され、プロファイルは新しい DAK 情報で自動的に更新されます。

プロファイルの作成 (CLI)

デバイスプロファイルを作成するには、CreateDeviceProfile API オペレーションまたは create-device-profile CLI コマンドを使用します。例えば、次のコマンドは Sidewalk エンドデバイス用のプロファイルを作成します。

```
aws iotwireless create-device-profile \
    --name sidewalk_device_profile --sidewalk {}
```

このコマンドを実行すると、Amazon リソースネーム (ARN) やプロファイルの ID を含むプロファイルの詳細が返されます。

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

プロフィール情報を表示してプロフィールを認定する

GetDeviceProfile API オペレーションまたは get-device-profile CLI コマンドを使用して、AWS IoT Core for Amazon Sidewalk のアカウントに追加したデバイスプロファイルに関する情報を取得します。デバイスプロファイルに関する情報を取得するには、プロファイル ID を指定します。その後、API は指定された識別子に一致するデバイスプロファイルに関する情報を返します。

CLI コマンドの例を以下に示します。

```
aws iotwireless get-device-profile \
    --id "12345678-234a-45bc-67de-e8901234f0a1" > device_profile.json
```

このコマンドを実行すると、デバイスプロファイルのパラメータ、アプリケーションサーバのパブリックキー、DeviceTypeId、ApId、認定ステータス、および DAKCertificate 情報が返されます。

この例では、認定ステータスと DAK 情報から、お使いのデバイスプロファイルが認定されていないことがわかります。プロファイルの認定を受けるには、Amazon Sidewalk サポートに連絡すると、プロファイルにデバイス制限のない新しい DAK が発行されます。

```
{
    "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 サポートチームがこの情報を確認すると、次の例に示すように、APID とファクトリーサポートされた DAK が届きます。

> **Note**
>
> -1 が MaxAllowedSignature であることは、DAK にデバイスの制限がないことを示します。DAK パラメータの詳細については、「DAKCertificateMetadata」を参照してください。

```
{
```

```
    "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
    }
}
```


次のステップ

ファクトリーサポートされている DAK を持つデバイスプロファイルを作成したので、チームから入手した YubiHSM キーを製造元に提供します。その後、デバイスはファクトリーで製造され、デバイスのシリアル番号 (SMSN) を含む制御ログ情報が Amazon Sidewalk に渡されます。このワークフローの詳細については、「Amazon Sidewalk ドキュメント」の「Amazon Sidewalk デバイスの製造」を参照してください。

その後、AWS IoT Core for Amazon Sidewalk にオンボードするデバイスのシリアル番号を指定することで、Sidewalk デバイスを一括プロビジョニングできます。AWS IoT Core for Amazon Sidewalk が制御ログを受信すると、制御ログのシリアル番号と入力されたシリアル番号を比較します。シリアル番号が一致すると、インポートタスクが AWS IoT Core for Amazon Sidewalk へのデバイスのオンボードを開始します。詳細については、「インポートタスクを使用した Sidewalk デバイスのプロビジョニング」を参照してください。

インポートタスクを使用した Sidewalk デバイスのプロビジョニング

このセクションでは、AWS IoT コンソール、AWS IoT Core for Amazon Sidewalk の API オペレーション、または AWS CLI を使用して Sidewalk デバイスを一括プロビジョニングする方法について

説明します。以下のセクションでは、Sidewalk デバイスを一括プロビジョニングする方法について説明します。

トピック

Sidewalk の一括プロビジョニングの仕組み

次の手順は、一括プロビジョニングの仕組みを示しています。

1. ワイヤレスデバイスのインポートタスクを開始する

   Sidewalk デバイスを一括でプロビジョニングするには、インポートタスクを作成し、AWS IoT Core for Amazon Sidewalk にオンボードするデバイスの Sidewalk 製造シリアル番号 (SMSN) を入力する必要があります。製造元が制御ログを Amazon Sidewalk にアップロードした後、デバイスの Sidewalk 製造シリアル番号 (SMSN) を CSV ファイルとして E メールで取得しました。ワークフローと制御ログの取得方法の詳細については、「Amazon Sidewalk ドキュメント」の「Amazon Sidewalk デバイスの製造」を参照してください。
2. インポートプロセスをバックグラウンドで実行する

   AWS IoT Core for Amazon Sidewalk はインポートタスクのリクエストを受け取ると、セットアップを開始し、システムを頻繁にポーリングするバックグラウンドプロセスを開始します。バックグラウンドプロセスがインポートタスクの指示を受け取ると、CSV ファイルの読み取りを開始します。AWS IoT Core for Amazon Sidewalk は同時に、制御ログが Amazon Sidewalk から受信されたかどうかをチェックします。
3. ワイヤレスデバイスレコードを作成する

   Amazon Sidewalk から制御ログを受け取ったら、AWS IoT Core for Amazon Sidewalk は制御ログのシリアル番号が CSV ファイルの SMSN 値と一致するかどうかをチェックします。シリアル番号が一致すると、AWS IoT Core for Amazon Sidewalk はこれらのシリアル番号に対応する Sidewalk デバイスのワイヤレスデバイスレコードの作成を開始します。すべてのデバイスのオンボードが完了すると、インポートタスクは完了とマークされます。

Sidewalk 一括プロビジョニングに関する主な考慮事項

Sidewalk デバイスを AWS IoT Core for Amazon Sidewalk に一括プロビジョニングする場合、次の主な考慮事項があります。

- デバイスプロファイルを作成したときと同じ AWS アカウント で、AWS IoT コンソールまたは AWS IoT Core for Amazon Sidewalk の API オペレーションを使用して、一括プロビジョニングを実行する必要があります。
- Sidewalk デバイスを一括プロビジョニングする前に、デバイスプロファイルにファクトリーサポートを示す DAK 情報が含まれている必要があります。そうしないと、AWS IoT コンソールを使用した一括プロビジョニングや、一括プロビジョニング API オペレーションが失敗する可能性があります。
- インポートタスクを開始した後、CSV ファイルの処理、ワイヤレスデバイスのインポート、およびワイヤレスデバイスの AWS IoT Core for Amazon Sidewalk へのオンボードに少なくとも 10 分以上かかる場合があります。
- ワイヤレスデバイスのインポートタスクは、開始後 90 日間実行されます。この間、Amazon Sidewalk から制御ログを受信したかどうかを確認します。90 日以内に Amazon Sidewalk から制御ログが届かない場合、タスクの詳細を表示すると、そのタスクは「完了」としてマークされ、有効期限が切れたことを示すメッセージが表示されます。インポートタスクで制御ログを待機していたデバイスのオンボーディングステータスが「失敗」としてマークされます。
- 既に作成したインポートタスクを更新しようとすると、そのタスクには、他のデバイスのみを追加できます。インポートタスクを作成した後、インポートタスクに既に追加されているデバイスでタスクが開始される前であれば、いつでも新しいデバイスを追加できます。更新ファイルに、元のインポートタスクに既に存在するデバイスのシリアル番号が含まれている場合、これらのシリアル番号は無視されます。
- 更新オペレーションをリクエストすると、インポートタスクの作成時に使用したものと同じ IAM ロールが Amazon S3 バケットの CSV ファイルにアクセスするとみなされます。
- インポートタスクは、タスクが既に正常に完了しているか、タスクが更新に失敗した場合にのみ削除できます。正しくない IAM ロールが提供された場合や、Amazon S3 バケットファイルが見つからない場合など、タスクの更新に失敗することがあります。インポートタスクが `PENDING` 状態の場合、更新も削除もできません。
- タスクにインポートする CSV ファイルは、次のセクションで説明する形式を使用する必要があります。

CSV ファイル形式

インポートタスクに指定する Amazon S3 バケットに含まれる CSV ファイルは、次の形式を使用する必要があります。

- 1 行目は `smsn` というキーワードを使用する必要があり、これはインポートする CSV ファイルにインポートするデバイスの SMSN が含まれていることを示しています。
- 2 行目以降には、オンボードするデバイスの SMSN が含まれている必要があります。デバイスの SMSN は 64 進数文字形式である必要があります。

この JSON ファイルは、サンプル CSV ファイル形式を示します。

```
smsn
1C1A10B0AC0A200C012BBAC2CBB1B21CB12C0CA2AC1C1BB22CAA01C1B0B01122
B122C2B1121BACA2221001AC1B22012AAC11112C11C2A100C1C2B012A1100C10
02B222C110B0A210B0A0C2C112CCCAC21C1C0B0AA1221AB1022A2CC11B1B1122
C2C021CA1C111CCAB1221C0021C1C2AAA0AA1A2A01ABC10CBAACCA2A0121022A
0CB22C01BBC2CA2C0B11001121ACB2ABB0BB0121C2BA101C012CC2B20C011AC0
```

Sidewalk 一括プロビジョニングの使用方法

以下のステップで、Amazon Sidewalk 一括プロビジョニングの使用方法を示します。

1. デバイスのシリアル番号を入力する

   Sidewalk デバイスをプロビジョニングするには、オンボードするデバイスのシリアル番号を入力する必要があります。デバイスをプロビジョニングするには、次のいずれかの方法を使用します。

   - Sidewalk の製造シリアル番号 (SMSN) を使用して、各デバイスを個別にプロビジョニングします。この方法は、適切な IAM ロールを持つ CSV ファイルをアップロードしたり、デバイスがタスクにオンボードされる準備が整うのを待ったりせずに、ワークフローをテストしてデバイスをより速くオンボードしたい場合に役立ちます。
   - プロビジョニングするデバイスの SMSN を含む Amazon S3 バケット URL を CSV ファイルで指定して、デバイスを一括でプロビジョニングします。この方法は、オンボードするデバイスが多数ある場合に特に役立ちます。この場合、各デバイスを個別にオンボードするのは手間がかかる場合があります。代わりに、Amazon S3 バケットにアップロードされた CSV ファイルへのパスと、ファイルにアクセスするための IAM ロールを指定するだけで済みます。

2. インポートタスクとデバイスオンボーディングステータスを取得する

   作成したインポートタスクごとに、タスクのオンボーディングステータスと、タスクに追加されたデバイスのオンボーディングステータスに関する情報を取得できます。タスクやデバイスのオンボーディングが失敗した理由など、その他のステータス情報も確認できます。詳細については、以下を参照してください。

3. (オプション) インポートタスクを更新または削除する

   既に作成したインポートタスクを更新または削除できます。

   - 既に追加されているデバイスでタスクが開始される前であれば、いつでもインポートタスクを更新したり、タスクにデバイスを追加したりできます。AWS IoT Core for Amazon Sidewalk は、インポートタスクの作成時に使用したものと同じ IAM ロールを引き受けます。タスクを作成するときは、タスクに追加するデバイスのシリアル番号を含む新しい CSV ファイルを指定します。

     > **Note**
     >
     > 既存のインポートタスクを更新する場合、タスクにデバイスを追加することのみ可能です。AWS IoT Core for Amazon Sidewalk は、インポートタスクに既にあるデバイスと、タスクに追加しようとしているデバイスとの間で、統合オペレーションを実行します。新しいファイルにインポートタスクに既に存在するデバイスのシリアル番号が含まれている場合、これらのシリアル番号は無視されます。

   - 既に正常に完了したインポートタスク、または IAM ロール情報が正しくない場合や、タスクの作成または更新時に S3 バケットファイルを使用できない場合などに更新に失敗したインポートタスクを削除できます。

トピック

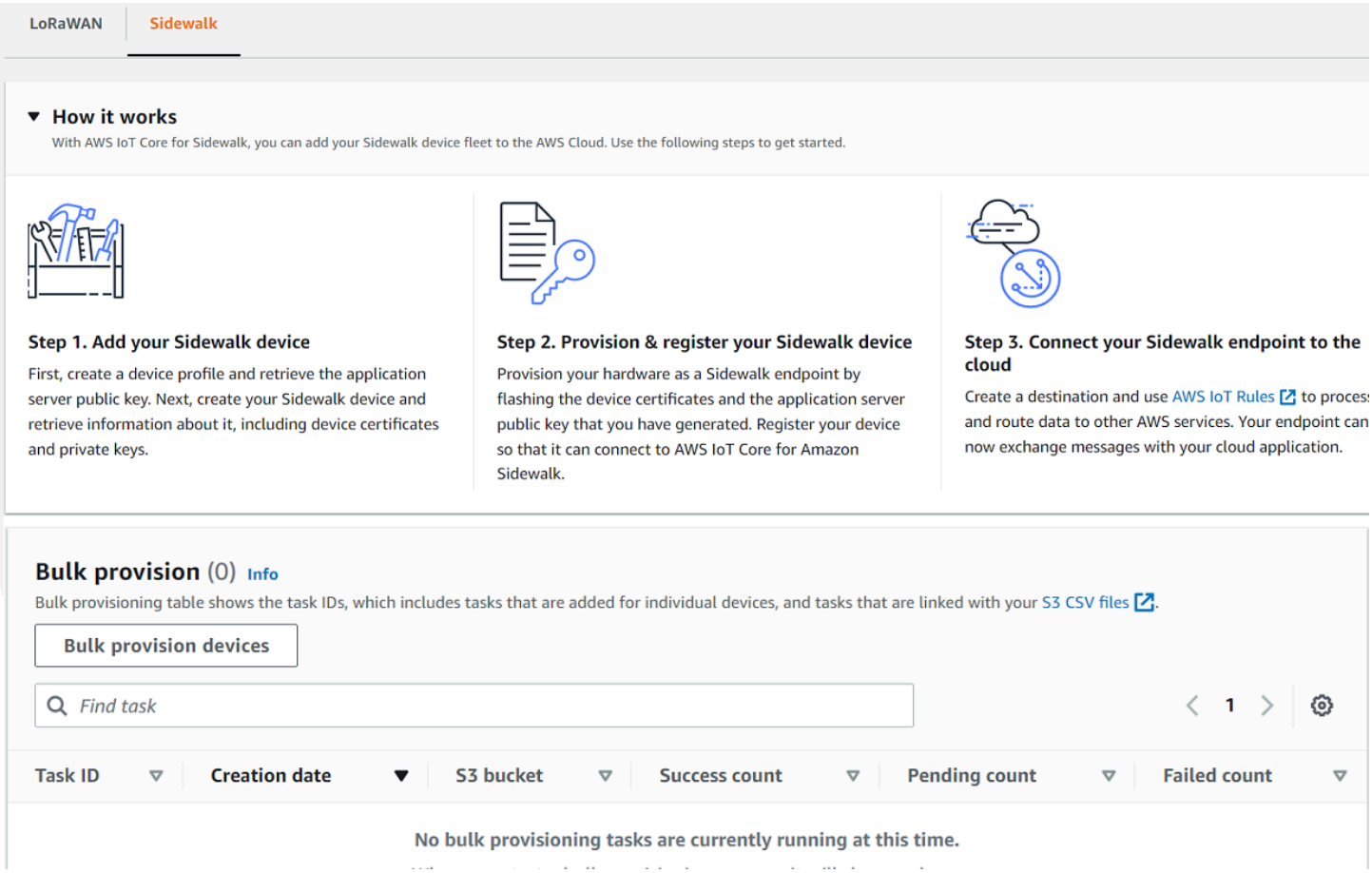
- Sidewalk デバイスの一括プロビジョニング
- インポートタスクとデバイスオンボーディングステータスの表示

Sidewalk デバイスの一括プロビジョニング

このセクションでは、AWS IoT コンソールと AWS CLI を使用して Sidewalk デバイスを AWS IoT Core for Amazon Sidewalk に一括でプロビジョニングする方法について説明します。

Sidewalk デバイスの一括プロビジョニング (コンソール)

AWS IoT コンソールを使用して Sidewalk デバイスを追加するには、デバイスハブの Sidewalk タブに移動し、[デバイスの一括プロビジョニング] を選択して、次の手順を実行します。



1. インポート方法を選択する

   AWS IoT Core for Amazon Sidewalk にオンボードするデバイスを一括でインポートする方法を指定します。

   - SMSN を使用して個々のデバイスをプロビジョニングするには、[ファクトリサポートされているデバイスを個別にプロビジョニング] を選択します。
   - デバイスとその SMS のリストを含む CSV ファイルを提供してデバイスを一括でプロビジョニングするには、[S3 バケットを使用] を選択します。

2. オンボードするデバイスを指定する

   デバイスをオンボードするために選択した方法に応じて、デバイス情報とシリアル番号を追加します。

a. [ファクトリサポートされているデバイスを個別にプロビジョニング] を選択した場合は、次の情報を指定します。

   i. オンボードする各デバイスの[名前]。名前は AWS アカウント および AWS リージョンで一意である必要があります。

   ii. [SMSN を入力] フィールドに、Sidewalk の製造シリアル番号 (SMSN) を入力します。

   iii. デバイスからのメッセージを他の AWS のサービス にルーティングするための IoT ルールを説明する[送信先]。

b. [S3 バケットを使用] を選択した場合:

   i. S3 URL 情報で構成される [S3 バケットの送信先] 情報を入力します。CSV ファイルを提供するには、[S3 を参照] を選択し、使用する CSV ファイルを選択します。

   AWS IoT Core for Amazon Sidewalk は、S3 バケット内にある CSV ファイルへのパスである S3 URL を自動的に入力します。パスの形式は `s3://bucket_name/file_name` です。Amazon Simple Storage Service コンソールでファイルを表示するには、[View] (表示) を選択します。

   ii. AWS IoT Core for Amazon Sidewalk がユーザーに代わって S3 バケット内の CSV ファイルにアクセスできるようにする、[S3 プロビジョニングロール] を提供します。新しいサービスロールを作成するか、既存のサービスロールを選択することができます。

   新しいロールを作成するには、[ロール名] を指定するか、空白のままにするとランダムな名前が自動的に生成されます。

   iii. デバイスから他の AWS のサービス にメッセージをルーティングするための IoT ルールを説明した [送信先] を指定します。

3. インポートタスクを開始する

   任意のタグを名前と値のペアとして指定し、[送信] を選択してワイヤレスデバイスのインポートタスクを開始します。

Sidewalk デバイスの一括プロビジョニング (CLI)

Sidewalk デバイスを AWS IoT Core for Amazon Sidewalk のアカウントにオンボードするには、デバイスを個別に追加するか、S3 バケットに含まれる CSV ファイルを指定するかに応じて、次の API オペレーションのいずれかを使用します。

- S3 CSV ファイルを使用してデバイスを一括アップロードする

S3 バケットに CSV ファイルを指定してデバイスを一括アップロードするには、StartWirelessDeviceImportTask API オペレーションまたは start-wireless-device-import-task AWS CLI コマンドを使用します。タスク作成時に、Amazon S3 バケット内の CSV ファイルへのパスと、AWS IoT Core for Amazon Sidewalk に CSV ファイルにアクセスするアクセス許可を付与する IAM ロールを指定します。

タスクの実行が開始されると、AWS IoT Core for Amazon Sidewalk は CSV ファイルの読み取りを開始し、ファイル内のシリアル番号 (SMSN) と Amazon Sidewalk から受け取った制御ログの対応する情報を比較します。シリアル番号が一致すると、これらのシリアル番号に対応するワイヤレスデバイスレコードの作成が開始されます。

次のコマンドは、インポートタスクの作成例を示しています。

```
aws iotwireless start-wireless-device-import-task \
    --cli-input-json "file://task.json"
```

以下は、task.json ファイルの内容を示しています。

task.json の内容

```
{
    "DestinationName": "Sidewalk_Destination",
    "Sidewalk": {
        "DeviceCreationFile": "s3://import_task_bucket/import_file1",
        "Role": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI"
    }
}
```

このコマンドを実行すると、インポートタスクの ID と ARN が返されます。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"
    "Id": "a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"
}
```

- SMSN を使用してデバイスを個別にプロビジョニングする

SMSN を使用してデバイスを個別にプロビジョニングするには、StartSingleWirelessDeviceImportTaskAPI オペレーションまたは start-single-wireless-device-import-task AWS CLI コマンドを使用します。タスクの作成時に、Sidewalk の送信先と、オンボードするデバイスのシリアル番号を指定します。

シリアル番号が Amazon Sidewalk から受信した制御ログの対応する情報と一致すると、タスクが実行され、ワイヤレスデバイスレコードが作成されます。

次のコマンドは、インポートタスクの作成例を示しています。

```
aws iotwireless start-single-wireless-device-import-task \
    --destination-name sidewalk_destination \
    --sidewalk
 '{"SidewalkManufacturingSn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A
```

このコマンドを実行すると、インポートタスクの ID と ARN が返されます。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/e2a5995e-743b-41f2-
a1e4-3ca6a5c5249f"
    "Id": "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
}
```

インポートタスクの更新または削除

インポートタスクに他のデバイスを追加する場合は、タスクを更新できます。タスクが不要になった場合や失敗した場合は、タスクを削除することもできます。タスクを更新または削除するタイミングについては、「Sidewalk 一括プロビジョニングの使用方法」を参照してください。

> **Warning**
>
> 削除の操作は永続的で、元には戻せません。既に正常に完了したインポートタスクを削除しても、そのタスクを使用してすでにオンボードされたエンドデバイスは削除されません。

インポートタスクを更新または削除するには:

- AWS IoT コンソールを使用する場合

次の手順では、AWS IoT コンソールを使用してインポートタスクを更新または削除する方法について説明します。

インポートタスクを更新するには:

1. AWS IoT コンソールの Sidewalk デバイスハブに移動します。
2. 更新するインポートタスクを選択し、次に [編集]を選択します。
3. タスクに追加するデバイスのシリアル番号を含む別の S3 ファイルを指定し、[送信] を選択します。

インポートタスクを削除するには:

1. AWS IoT コンソールの Sidewalk デバイスハブに移動します。
2. 削除するタスクを選択し、[削除] を選択します。

- AWS IoT Wireless API または AWS CLI の使用

  次の AWS IoT Wireless API オペレーションまたは CLI コマンドを使用して、インポートタスクを更新または削除します。

  - **`UpdateWirelessDeviceImportTask`** API または **`update-wireless-device-import-task`** CLI

    この API オペレーションは、Amazon S3 CSV ファイルの内容を既存のインポートタスクに追加します。追加できるのは、以前にタスクに含まれていなかったデバイスのシリアル番号のみです。

  - **`DeleteWirelessDeviceImportTask`** API または **`delete-wireless-device-import-task`** CLI

    この API オペレーションは、インポートタスク ID を使用して削除対象としてマークされたインポートタスクを削除します。

インポートタスクとデバイスオンボーディングステータスの表示

タスクに追加したワイヤレスデバイスのインポートタスクと、Sidewalk デバイスには、次のステータスメッセージのいずれかが表示されます。これらのメッセージは、AWS IoT コンソールに表示されるか、AWS IoT Wireless API オペレーションまたは AWS CLI コマンドのいずれかを使用してこれらのタスクとそのデバイスに関する情報を取得するときに表示されます。

インポートタスクのステータス情報を表示する

インポートタスクを作成すると、作成したインポートタスクと、タスクに追加されたデバイスのオンボーディングステータスを表示できます。オンボーディングステータスには、オンボードが保留になっているデバイスの数、正常にオンボードされたデバイスの数、オンボードに失敗したデバイスの数が表示されます。

インポートタスクが作成されたばかりの場合、保留中のカウントには、追加されたデバイスの数に対応する値が表示されます。タスクが開始され、CSV ファイルを読み取ってワイヤレスデバイスレコードを作成すると、[保留中のカウント] は減少し、デバイスが正常にオンボードされると、[成功カウント] は増加します。いずれかのデバイスがオンボードに失敗すると、[失敗カウント] が増加します。

インポートタスクとデバイスオンボーディングステータスを表示するには:

- AWS IoT コンソールを使用する場合

  AWS IoT コンソールの Sidewalk デバイスハブでは、作成したインポートタスクと、デバイスのオンボーディングステータス情報の概要の数を確認できます。作成したインポートタスクの詳細を表示すると、デバイスのオンボーディングステータスに関する追加情報が表示されます。

- AWS IoT Wireless API または AWS CLI の使用

  デバイスのオンボーディングステータスを表示するには、次の AWS IoT Wireless API オペレーションまたは対応する AWS CLI コマンドのいずれかを使用します。

  - **ListWirelessDeviceImportTasks** API または **list-wireless-device-import-tasks** CLI

    この API オペレーションは、AWS IoT Wireless のアカウントに追加されたすべてのインポートタスクとそのステータスに関する情報を返します。また、これらのタスクにおける Sidewalk デバイスのオンボーディングステータスの概要も返されます。

  - **ListDevicesForWirelessDeviceImportTask** API または **list-devices-for-wireless-device-import-task** CLI

    この API オペレーションは、指定されたインポートタスクとそのステータスに関する情報、およびインポートタスクに追加されたすべての Sidewalk デバイスとそのオンボーディングステータス情報を返します。

  - **GetWirelessDeviceImportTask** API または **get-wireless-device-import-task** CLI

この API オペレーションは、指定されたインポートタスクとそのステータスに関する情報、およびそのタスクに含まれる Sidewalk デバイスのオンボーディングステータスの概要数を返します。

インポートタスクのステータス

AWS アカウント で作成したインポートタスクには、次のステータスメッセージのいずれかが表示されます。ステータスは、インポートタスクが処理を開始したか、完了したか、失敗したかを示します。AWS IoT コンソールまたは任意の AWS IoT Wireless API オペレーションの `StatusReason` パラメータを使用して、追加のステータス詳細を取得することもできます。

- 初期化

  AWS IoT Core for Amazon Sidewalk は、ワイヤレスデバイスのインポートタスクリクエストを受信し、タスクをセットアップ中です。

- 初期化

  AWS IoT Core for Amazon Sidewalk はインポートタスクのセットアップを完了し、シリアル番号 (SMSN) を使用してデバイスをインポートし、タスクの処理を続行するために、制御ログが届くのを待っています。

- 保留中

  インポートタスクはキューで処理を待機中です。AWS IoT Core for Amazon Sidewalk は、処理キューにある他のタスクを評価しています。

- 完了

  インポートタスクが処理され、完了しました。

- FAILED

  インポートタスクまたはデバイスタスクが失敗しました。`StatusReason` パラメータを使用して、検証例外など、インポートタスクが失敗した理由を特定できます。

- DELETING

  インポートタスクは削除対象としてマークされており、削除中です。

デバイスのオンボーディングステータス

インポートタスクに追加した Sidewalk デバイスには、次のステータスメッセージのいずれかが表示されます。ステータスは、デバイスをオンボードする準備ができているか、オンボード済みか、オンボードに失敗したかを示します。AWS IoT コンソールまたは AWS IoT Wireless API オペレーション、`ListDevicesForWirelessDeviceImportTask` の `OnboardingStatusReason` パラメータを使用して、追加のステータス詳細を取得することもできます。

- 初期化

  AWS IoT Core for Amazon Sidewalk はインポートタスクのセットアップを完了し、シリアル番号 (SMSN) を使用してデバイスをインポートし、タスクの処理を続行するために、制御ログが届くのを待っています。
- 保留中

  インポートタスクは、処理と、タスクへのデバイスのオンボーディングが開始されるのをキューで待機しています。AWS IoT Core for Amazon Sidewalk は、処理キューにある他のタスクを評価しています。
- オンボード済み

  Sidewalk デバイスはインポートタスクに正常にオンボードされました。
- FAILED

  インポートタスクまたはデバイスタスクが失敗し、Sidewalk デバイスがタスクへのオンボードに失敗しました。`OnboardingStatusReason` パラメータを使用して、デバイスのオンボーディングが失敗した理由に関する追加情報を取得できます。

AWS IoT Wireless のセキュリティ

AWS ではクラウドセキュリティが最優先事項です。セキュリティを最も重視する組織の要件を満たすために構築された AWS のデータセンターとネットワークアーキテクチャは、お客様に大きく貢献します。

セキュリティは、AWS とお客様とが共有する責務です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウドで AWS サービスを実行するインフラストラクチャを保護する責任を負います。また AWS は、安全に使用できるサービスを提供します。AWS コンプライアンスプログラムの一環として、サードパーティー監査者が定期的にセキュリティの有効性をテストおよび検証します。AWS IoT Wireless に適用するコンプライアンスプログラムの詳細については、コンプライアンスプログラムによる対象範囲内の AWS のサービスをご参照ください。
- クラウド内のセキュリティ - ユーザーの責任は、使用する AWS サービスに応じて異なります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、AWS IoT Wireless を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。ここでは、セキュリティとコンプライアンスの目標を達成するように AWS IoT Wireless を設定する方法について説明します。また、AWS IoT Wireless リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法も説明します。

コンテンツ

AWS IoT Wireless でのデータ保護

AWS 責任共有モデルは、AWS IoT Wireless のデータ保護に適用されます。このモデルで説明されているように、AWS には、AWS クラウド のすべてを実行するグローバルインフラストラクチャを保

護する責任があります。ユーザーには、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、「AWS セキュリティブログ」に投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データを保護するため、AWS アカウント の認証情報を保護し、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーをセットアップすることをお勧めします。こうすると、それぞれのジョブを遂行するために必要なアクセス許可のみを各ユーザーに付与できます。また、次の方法でデータを保護することをおすすめします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 が必須です。TLS 1.3 が推奨されます。
- AWS CloudTrail で API とユーザーアクティビティロギングをセットアップします。
- AWS のサービス内でデフォルトである、すべてのセキュリティ管理に加え、AWS の暗号化ソリューションを使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API により AWS にアクセスするときに FIPS 140-2 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

お客様の E メールアドレスなどの機密情報やセンシティブ情報は、タグや名前フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これは、AWS IoT Wirelessで作業する場合や、コンソール、API、AWS CLI、または AWS SDK を使用しているその他の AWS のサービスで作業する場合も同様です。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

AWS IoT Wireless でのデータの暗号化

デフォルトでは、転送中および保管中のすべての AWS IoT Wireless データは暗号化されます。AWS IoT Wireless は、AWS KMS key からのカスタマーマネージド AWS KMS キーをサポートしていません。データを暗号化するために、AWS IoT Wireless は AWS 所有のキー のみを使用します。

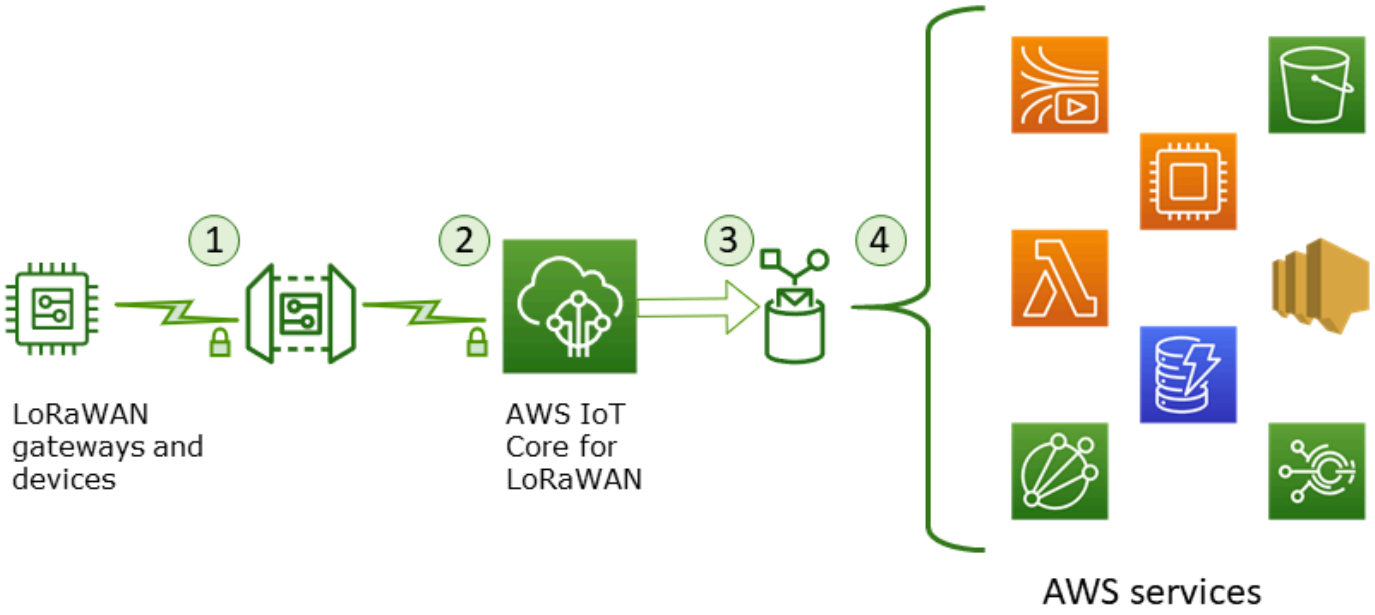
AWS IoT Core for LoRaWAN によるデータとトランスポートのセキュリティ

AWS IoT Core for LoRaWAN は、以下の方法を使用して、LoRaWAN デバイス、ゲートウェイ、および AWS IoT Core for LoRaWAN 間のデータと通信を保護します。

- デバイスが LoRaWAN ゲートウェイと通信するときに従うセキュリティのベストプラクティス (ホワイトペーパー「LoRaWAN セキュリティ」で説明されている通り)。
- ゲートウェイを AWS IoT Core for LoRaWAN に接続し、データを他の AWSのサービスに送信するためにAWS IoT Core が使用するセキュリティ。詳細については、「AWS IoT Core でのデータ保護」を参照してください。

システム全体でデータを保護する方法

この図は、AWS IoT Core for LoRaWAN に接続された LoRaWAN システムの主要な要素を特定し、全体的にデータをどのように保護しているかを示すものです。



1. LoRaWAN ワイヤレスデバイスは、バイナリメッセージを送信する前に AES128 CTR モードを使用して暗号化します。
2. AWS IoT Core for LoRaWAN へのゲートウェイ接続は、「AWS IoT でのトランスポートセキュリティ」で説明されているように、TLS によって保護されています。AWS IoT Core for LoRaWAN

は、バイナリメッセージを復号化し、復号化されたバイナリメッセージペイロードを base64 文字列としてエンコードします。

3. 結果として発生する base64 でエンコードされたメッセージは、デバイスに割り当てられた送信先で説明されている AWS IoT ルールにメッセージペイロードとして送信されます。AWS 内のデータは、AWS が所有するキーを使用して暗号化されます。
4. AWS IoT ルールは、ルールの設定に記載されているサービスにメッセージデータを送信します。AWS 内のデータは、AWS が所有するキーを使用して暗号化されます。

LoRaWAN デバイスとゲートウェイトランスポートセキュリティ

LoRaWAN デバイスと AWS IoT Core for LoRaWAN は、事前共有ルートキーを保存します。プロトコルに従って、LoRaWAN デバイスと AWS IoT Core for LoRaWAN の両方によって、セッションキーが生成されます。対称セッションキーは、標準の AES-128 CTR モードでの暗号化および復号に使用されます。4 バイトのメッセージ整合性コード (MIC) も、標準の AES-128 CMAC アルゴリズムに従ってデータ整合性をチェックするために使用されます。セッションキーは、Join/Rejoin プロセスを使用して更新できます。

LoRa ゲートウェイ用のセキュリティプラクティスは、LoRaWAN 仕様に記載されています。LoRa ゲートウェイは `Basics Station` を使用して WebSocket を介して AWS IoT Core for LoRaWAN に接続します。AWS IoT Core for LoRaWAN でサポートされるのは `Basics Station` バージョン 2.0.4 以降のみです。

ウェブソケット接続が確立される前に、AWS IoT Core for LoRaWAN は TLS Server and Client Authentication モードを使用してゲートウェイを認証します。LoRaWAN プロトコルの機密性を確保するために、TLS バージョン 1.2 が使用されます。TLS サポートは、多くのプログラミング言語とオペレーティングシステムで使用できます。AWS 内のデータは、特定の AWS のサービスによって暗号化されます。他の AWS のサービスのデータ暗号化の詳細については、そのサービスのセキュリティドキュメントを参照してください。

AWS IoT Core for LoRaWAN は、TLS 認証に使用される証明書とキーを設定および更新する Configuration and Update Server (CUPS) も保守します。

AWS IoT Wireless の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御するために役立つ AWS のサービスです。IAM 管理者は、AWS IoT Wireless リソースの使用を 認証 (サインイン) し、 認可 (アクセス許可を持つ) できるユーザーを制御します。IAM は、追加費用なしで使用できる AWS のサービスです。

トピック

対象者

AWS Identity and Access Management (IAM) の使用方法は、AWS IoT Wireless で行う作業によって異なります。

サービスユーザー - ジョブを実行するために AWS IoT Wireless サービスを使用する場合は、管理者が必要なアクセス許可と認証情報を用意します。作業を実行するためにさらに多くの AWS IoT Wireless 機能を使用するとき、追加のアクセス許可が必要になる場合があります。アクセスの管理方法を理解すると、管理者から適切な権限をリクエストするのに役に立ちます。AWS IoT Wireless の機能にアクセスできない場合は、AWS IoT Wireless の ID とアクセスのトラブルシューティング を参照してください。

サービス管理者 – 社内の AWS IoT Wireless リソースを担当している場合は、おそらく AWS IoT Wireless へのフルアクセスがあります。サービスのユーザーがどの AWS IoT Wireless 機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。お客様の会社で AWS IoT Wireless で IAM を利用する方法の詳細については、AWS IoT Wireless と IAM の連携方法 を参照してください。

IAM 管理者 - 管理者は、AWS IoT Wireless へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる AWS IoT Wireless の ID ベースのポリシーの例を表示するには、AWS IoT Wireless の ID ベースのポリシーの例 を参照してください。

アイデンティティによる認証

認証とは、アイデンティティ認証情報を使用して AWSにサインインする方法です。ユーザーは、AWS アカウントのルートユーザー として、または IAM ロールを引き受けることによって、認証済み (AWS にサインイン済み) である必要があります。

ID ソースから提供された認証情報を使用して、フェデレーティッドアイデンティティとして AWS にサインインできます。 AWS IAM Identity Centerフェデレーティッドアイデンティティの例としては、IAM アイデンティティセンターユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報などがあります。フェデレーティッドアイデンティティとしてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーションを使用して AWSにアクセスする場合、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、AWS Management Console または AWS アクセスポータルにサインインできます。AWS へのサインインの詳細については、『AWS サインイン ユーザーガイド』の「AWS アカウント にサインインする方法」を参照してください。

プログラムで AWS にアクセスする場合、AWS は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) を提供し、認証情報でリクエストに暗号で署名します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。リクエストに署名する推奨方法の使用については、『IAM ユーザーガイド』の「AWS API リクエストの署名」を参照してください。

使用する認証方法を問わず、追加のセキュリティ情報の提供が求められる場合もあります。例えば、AWS では多要素認証 (MFA) を使用してアカウントのセキュリティを高めることを推奨しています。詳細については、「AWS IAM Identity Center ユーザーガイド」の「多要素認証」 および「IAM ユーザーガイド」の「AWS での多要素認証 (MFA) の使用」を参照してください。

AWS アカウントのルートユーザー

AWS アカウント を作成する場合、このアカウントのすべての AWS のサービス とリソースに対して完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。このアイデンティティは AWS アカウントのルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることによってアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報を保護し、それらを使用してルートユーザーのみが実行できるタスクを実行してください。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

IAM ユーザーとグループ

IAM ユーザー は、1 人のユーザーまたは 1 つのアプリケーションに対して特定の許可を持つ AWS アカウント 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合

は、アクセスキーをローテーションすることをお勧めします。詳細については、「IAM ユーザーガイド」の「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する権限を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳細については、「IAM ユーザーガイド」の「IAM ユーザー (ロールではなく) の作成が適している場合」を参照してください。

IAM ロール

> **Note**
>
> AWS IoT Wireless は、サービスロールとサービスにリンクされたロールをサポートしていません。

IAM ロール は、特定の許可を持つ、AWS アカウント 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロールを切り替えることにより、AWS Management Console で一時的に IAM ロールを引き受けることができます。ロールを引き受けるには、AWS CLI または AWS API オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「IAM ロールの使用」を参照してください。

IAM ロールと一時的な認証情報は、次のような状況で役立ちます。

- フェデレーションユーザーアクセス – フェデレーティッドアイデンティティに許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッドアイデンティティが認証されると、そのアイデンティティはロールに関連付けられ、ロールで定義されている権限が付与されます。フェデレーションの詳細については、「IAM ユーザーガイド」の「Creating a role for a third-party Identity Provider」(サードパーティーアイデンティティプロバイダー向けロールの作成) を参照してください。IAM Identity Center を使用する場合、許可セットを設定します。ア

イデンティティが認証後にアクセスできるものを制御するため、IAM アイデンティティセンターは、アクセス許可セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「アクセス許可セット」を参照してください。

- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースへのアクセスを別のアカウントの人物 (信頼できるプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の AWS のサービスでは、(ロールをプロキシとして使用する代わりに) リソースにポリシーを直接アタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。
- クロスサービスアクセス - 一部の AWS のサービスでは、他の AWS のサービスの機能を使用します。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの権限、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して AWS でアクションを実行するユーザーは、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、AWS のサービスを呼び出すプリンシパルの権限を、AWS のサービスのリクエストと合わせて使用し、ダウンストリームのサービスに対してリクエストを行います。FAS リクエストは、サービスが、完了するために他の AWS のサービス またはリソースとのやりとりを必要とするリクエストを受け取ったときにのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービス にアクセス許可を委任するロールの作成」を参照してください。
  - サービスにリンクされたロール - サービスにリンクされたロールは、AWS のサービス にリンクされたサービスロールの一種です。サービスがロールを引き受け、ユーザーに代わってアクションを実行できるようになります。サービスにリンクされたロールは、AWS アカウント に表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの許可を表示できますが、編集はできません。

- Amazon EC2 で実行されているアプリケーション - EC2 インスタンスで実行され、AWS CLI または AWS API 要求を行っているアプリケーションの一時的な認証情報を管理するには、IAM ロールを使用できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスに添付されたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する」参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、「IAM ユーザーガイド」の「(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?」を参照してください。

ポリシーを使用したアクセス権の管理

AWS でアクセスを制御するには、ポリシーを作成して AWS アイデンティティまたはリソースにアタッチします。ポリシーは AWS のオブジェクトであり、アイデンティティやリソースに関連付けて、これらのアクセス許可を定義します。AWS は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うと、これらのポリシーを評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。大半のポリシーは JSON ドキュメントとして AWSに保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は AWSJSON ポリシーを使用して、だれが何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するためのアクセス許可をユーザーに付与するため、IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの権限を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。このポリシーがあるユーザーは、AWS Management Console、AWS CLI、または AWS API からロール情報を取得できます。

アイデンティティベースポリシー

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザー

とロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

アイデンティティベースのポリシーは、さらに インラインポリシー または マネージドポリシー に分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれます。管理ポリシーは、AWS アカウント 内の複数のユーザー、グループ、およびロールにアタッチできるスタンドアロンポリシーです。マネージドポリシーには、AWS マネージドポリシーとカスタマー管理ポリシーがあります。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「マネージドポリシーとインラインポリシーの比較」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーションユーザー、または AWS のサービスを含めることができます。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは IAM の AWSマネージドポリシーは使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、AWS WAF、および Amazon VPC は、ACL をサポートするサービスの例です。ACL の詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS では、他の一般的ではないポリシータイプをサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与される最大の許可を設定できます。

- 権限の境界 - 権限の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる許可の上限を設定する高度な機能です。エンティティに権限の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとその権限の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、権限の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCP) - SCP は、AWS Organizations で組織や組織単位 (OU) の最大許可を指定する JSON ポリシーです。AWS Organizations は、ユーザーのビジネスが所有する複数の AWS アカウント をグループ化し、一元的に管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP はメンバーアカウントのエンティティに対する権限を制限します (各 AWS アカウントのルートユーザー など)。Organizations と SCP の詳細については、「AWS Organizations ユーザーガイド」の「SCP の仕組み」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限の範囲は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関連するとき、リクエストを許可するかどうかを AWS が決定する方法の詳細については、『IAM ユーザーガイド』の「ポリシーの評価ロジック」を参照してください。

AWS IoT Wireless と IAM の連携方法

IAM を使用して AWS IoT Wireless へのアクセスを管理する前に、AWS IoT Wireless で使用できる IAM 機能について理解しておく必要があります。AWS IoT Wireless およびその他の AWS のサービスが IAM と連携する方法の概要を把握するには、IAM ユーザーガイドの「IAM と連携する AWS のサービス」を参照してください。

AWS IoT Wireless で使用できる IAM の機能

IAM の機能	AWS IoT Wireless サポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	いいえ
ポリシーアクション	あり
ポリシーリソース	はい
ポリシー条件キー	はい
ACL	なし
ABAC (ポリシー内のタグ)	はい
一時的な認証情報	あり
プリンシパル権限	あり
サービスロール	いいえ
サービスリンクロール	いいえ

トピック

- AWS IoT Wireless の ID ベースのポリシー
- AWS IoT Wireless 内のリソースベースのポリシー
- ポリシーアクション
- ポリシーリソース
- 条件キー
- アクセスコントロールリスト (ACL)
- AWS IoT Wireless を備えた ABAC
- AWS IoT Wireless での一時的な認証情報の使用
- AWS IoT Wireless のクロスサービスプリンシパル権限
- サービスロール

- AWS IoT Wireless のサービスにリンクされたロール

AWS IoT Wireless の ID ベースのポリシー

アイデンティティベースポリシーをサポートする	あり

アイデンティティベースのポリシーは、IAM ユーザー、ユーザーグループ、ロールなどのアイデンティティにアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それがアタッチされているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素について学ぶには、『IAM ユーザーガイド』の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

例

AWS IoT Wireless の ID ベースのポリシーの例を表示するには、AWS IoT Wireless の ID ベースのポリシーの例 を参照してください。

AWS IoT Wireless 内のリソースベースのポリシー

リソースベースのポリシーのサポート	なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリン

シパルには、アカウント、ユーザー、ロール、フェデレーションユーザー、または AWS のサービスを含めることができます。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる AWS アカウント にある場合、信頼できるアカウントの IAM 管理者は、リソースへのアクセス許可をプリンシパルエンティティ (ユーザーまたはロール) に付与する必要もあります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーを追加する必要はありません。詳細については、『IAM ユーザーガイド』の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

ポリシーアクション

ポリシーアクションに対するサポート	あり

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーの`Action`要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じです。一致する API オペレーションのない権限のみのアクションなど、いくつかの例外があります。また、ポリシーに複数アクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

AWS IoT Wireless のポリシーアクションは、アクションの前にプレフィックス `iotwireless:` を使用します。例えば、AWS アカウント に登録されているすべてのワイヤレスデバイスを `ListWirelessDevices` APIオペレーションで一覧表示するアクセス許可を誰かに付与するには、ポリシーに `iotwireless:ListWirelessDevices` アクションを含めます。ポリシーステートメントには、`Action` または `NotAction` 要素を含める必要があります。AWS IoT Wireless は、このサービスで実行できるタスクを記述する独自のアクションのセットを定義します。

単一ステートメントに複数アクションを指定するには、次のようにカンマで区切ります:

```
"Action": [
      "iotwireless:ListMulticastGroups",
      "iotwireless:ListFuotaTasks"
   ]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、Get という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

```
"Action": "iotwireless:Get*"
```

AWS IoT Wireless アクションのリストを表示するには、IAM ユーザーガイドの「AWS IoT Wireless によって定義されたアクション」を参照してください。

ポリシーリソース

ポリシーリソースに対するサポート	あり

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

Resource JSON ポリシーの要素は、オブジェクトあるいはアクションが適用されるオブジェクトを指定します。ステートメントには、Resource または NotResource 要素を含める必要があります。ベストプラクティスとしては、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの権限と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

AWS IoT Wireless サービスには次の ARN があります。

```
arn:${Partition}:iotwireless:${Region}:${Account}:${Resource}/${Resource-id}
```

ARN の形式の詳細については、「Amazon リソースネーム (ARN) と AWS サービスの名前空間」を参照してください。

例えば、ステートメントでネットワークアナライザ設定 NAConfig1 を指定するには、次の ARN を使用します。

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:NetworkAnalyzerConfiguration/NAConfig1"
```

特定のアカウントに属するすべての FUOTA タスクを指定するには、ワイルドカード (*) を使用します。

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:FuotaTask/*"
```

リソースの一覧表示など、一部の AWS IoT Wireless アクションは、特定のリソースで実行できません。このような場合は、ワイルドカード *を使用する必要があります。

```
"Resource": "*"
```

AWS IoT Wireless API アクションの多くが複数のリソースと関連します。例えば、AssociateWirelessDeviceWithThing はワイヤレスデバイスを AWS IoT モノに関連付けるため、IAM ユーザーはデバイスと IoT モノを使用するためのアクセス許可を持っている必要があります。複数リソースを単一ステートメントで指定するには、ARN をカンマで区切ります。

```
"Resource": [
      "WirelessDevice",
      "thing"
```

AWS IoT Wireless リソースタイプとその ARN のリストを表示するには、IAM ユーザーガイドの「AWS IoT Wireless で定義されるリソース」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、AWS IoT Wireless で定義されるアクションを参照してください。

条件キー

サービス固有のポリシー条件キーのサポート	はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。equal や less than などの条件演算子を使用して条件式を作成することによって、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素が指定されている場合、または 1 つの Condition 要素に複数のキーが指定されている場合、AWS では AND 論理演算子を使用してそれらを評価します。単一の条件キーに複数の値が指定されている場合、AWS では OR 論理演算子を使用して条件を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる許可を付与できます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS はグローバル条件キーとサービス固有の条件キーをサポートしています。すべての AWS グローバル条件キーを確認するには、『IAM ユーザーガイド』の「AWS グローバル条件コンテキストキー」を参照してください。

AWS IoT Wireless は独自の条件キーを定義し、一部のグローバル条件キーの使用をサポートしています。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」の「AWS グローバル条件コンテキストキー」を参照してください。AWS IoT Wireless 条件キーのリストを表示するには、IAM ユーザーガイドの「AWS IoT Wireless の条件キー」を参照してください。どのアクションおよびリソースと条件キーを使用できるかについては、「AWS IoT Wireless で定義されるアクション」を参照してください。

アクセスコントロールリスト (ACL)

ACL のサポート	なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための権限を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

AWS IoT Wireless を備えた ABAC

ABAC のサポート (ポリシー内のタグ)	はい

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義するアクセス許可戦略です。AWS では、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール)、および多数の AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。次に、プリンシパルのタグがアクセスを試行するリソースのタグと一致したときにオペレーションを許可するよう、ABAC ポリシーを設計します。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを制御するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素 でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値は Yes です。サービスが一部のリソースタイプに対してのみ 3 つの条件キーすべてをサポートする場合、値は Partial です。

ABAC の詳細については、『IAM ユーザーガイド』の「ABAC とは?」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「属性に基づくアクセスコントロール (ABAC) を使用する」を参照してください。

AWS IoT Wireless リソースにタグをアタッチすることも、AWS IoT Wireless へのリクエストでタグを渡すこともできます。タグに基づいてアクセスを制御するには、`YOUR-SERVICE-PREFIX:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの 条件要素 でタグ情報を提供します。AWS IoT Wireless リソースのタグ付けの詳細については、「AWS IoT Wireless リソースのタグ付け」を参照してください。

AWS IoT Wireless での一時的な認証情報の使用

一時的な認証情報のサポート	あり

AWS のサービス には、一時的な認証情報を使用してサインインしても機能しないものがあります。一時的な認証情報で機能する AWS のサービスなどの詳細については、「IAM ユーザーガイド」の「IAM と連携する AWS のサービス」を参照してください。

ユーザー名とパスワード以外の方法で AWS Management Console にサインインする場合は、一時的な認証情報を使用していることになります。例えば、会社のシングルサインオン (SSO) リンクを使用して AWS にアクセスすると、そのプロセスは自動的に一時的な認証情報を作成します。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報

が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ロールへの切り替え (コンソール)」を参照してください。

一時的な認証情報は、AWS CLI または AWS API を使用して手動で作成できます。作成後、一時的な認証情報を使用して AWS にアクセスできるようになります。AWS は、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「IAM の一時的セキュリティ認証情報」を参照してください。

AWS IoT Wireless のクロスサービスプリンシパル権限

フォワードアクセスセッション (FAS) をサポート	はい

IAM ユーザーまたはロールを使用して AWSでアクションを実行するユーザーは、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行してから、別のサービスの別のアクションを開始することがあります。FAS は、AWS のサービスを呼び出すプリンシパルの権限を、AWS のサービスのリクエストと合わせて使用し、ダウンストリームのサービスに対してリクエストを行います。FAS リクエストは、サービスが、完了するために他の AWS のサービス またはリソースとのやりとりを必要とするリクエストを受け取ったときにのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

サービスロール

サービスロールのサポート	いいえ

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービス にアクセス許可を委任するロールの作成」を参照してください。

AWS IoT Wireless のサービスにリンクされたロール

サービスにリンクされたロールのサポート	いいえ

サービスにリンクされたロールは、AWS のサービスにリンクされているサービスロールの一種です。サービスがロールを引き受け、ユーザーに代わってアクションを実行できるようになります。サービスにリンクされたロールは、AWS アカウント に表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの許可を表示できますが、編集はできません。

AWS IoT Wireless の ID ベースのポリシーの例

デフォルトでは、IAM ユーザーとロールには AWS IoT Wireless リソースを作成または変更するためのアクセス許可はありません。AWS Management Console、AWS CLI、または AWS API を使用してタスクを実行することもできません。IAM 管理者は、ユーザーとロールに必要な、指定されたリソースで特定の API オペレーションを実行する権限をユーザーとロールに付与する IAM ポリシーを作成する必要があります。続いて、管理者はそれらの権限が必要な IAM ユーザーまたはグループにそのポリシーをアタッチする必要があります。

JSON ポリシードキュメントのこれらの例を使用して、IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「JSON タブでのポリシーの作成」を参照してください。

トピック

- ポリシーのベストプラクティス
- AWS IoT Wireless コンソールの使用
- ユーザーが自分の許可を表示できるようにする
- AWS IoT Wireless ワイヤレスデバイスのアクションを実行するために必要なアクセス許可

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが AWS IoT Wireless リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、AWS アカウント に料金が発生する可能性があります。アイデンティティベースのポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください。

- AWS マネージドポリシーを使用して開始し、最小特権の権限に移行する – ユーザーとワークロードへの権限の付与を開始するには、多くの一般的なユースケースのために権限を付与する AWS マネージドポリシーを使用します。これらは AWS アカウントで使用できます。ユースケースに応じた AWS カスタマー管理ポリシーを定義することで、許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「AWS ジョブ機能の管理ポリシー」を参照してください。

- 最小特権を適用する - IAM ポリシーで許可を設定するときは、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権権限とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「IAM でのポリシーとアクセス許可」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。また、AWS のサービス など特定の AWS CloudFormation を介して使用する場合、条件を使用してサービスアクションへのアクセスを許可することもできます。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素：条件) を参照してください。
- IAM アクセスアナライザーを使用して IAM ポリシーを検証し、安全で機能的な許可を確保する - IAM アクセスアナライザーは、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、「IAM ユーザーガイド」の「IAM Access Analyzer ポリシーの検証」を参照してください。
- 多要素認証 (MFA) を要求する - AWS アカウント で IAM ユーザーまたはルートユーザーを要求するシナリオがある場合は、セキュリティを強化するために MFA をオンにします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「MFA 保護 API アクセスの設定」を参照してください。

IAM でのベストプラクティスの詳細については、『IAM ユーザーガイド』の「IAM でのセキュリティのベストプラクティス」を参照してください。

AWS IoT Wireless コンソールの使用

AWS IoT Wireless コンソールにアクセスするには、一連の最小限のアクセス許可が必要です。これらのアクセス許可により、AWS アカウントの AWS IoT Wireless リソースの詳細をリストおよび表示できます。最小限必要な許可よりも厳しく制限されたアイデンティティベースポリシーを作成すると、そのポリシーを添付したエンティティ (IAM ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

これらのエンティティが AWS IoT Wireless コンソールを使用できるように、エンティティに次の AWS マネージドポリシーもアタッチします。詳細については、IAM ユーザーガイドの「ユーザーへのアクセス許可の追加」を参照してください。

```
AWSIoTWirelessFullAccess
```

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソール許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーが自分の許可を表示できるようにする

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を、IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI か AWS API を使用してプログラム的に、このアクションを完了する権限が含まれています。

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

```
        }
    ]
}
```

AWS IoT Wireless ワイヤレスデバイスのアクションを実行するために必要なアクセス許可

ID ベースのポリシーの条件を使用して、タグに基づいて AWS IoT Wireless アクションへのアクセスを制御できます。この例では、デバイスを作成して管理できるポリシーを作成する方法を示します。ただし、アクセス許可は、モノタグ Owner にそのユーザーのユーザー名の値がある場合のみ、付与されます。このポリシーでは、このアクションをコンソールで実行するために必要なアクセス権限も付与します。

```
{
 "Version": "2012-10-17",
 "Statement": [{
    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
              "iotwireless:CreateWirelessDevice",
              "iotwireless:GetWirelessDevice",
              "iotwireless:ListWirelessDevices",
              "iotwireless:UpdateWirelessDevice",
              "iotwireless:DeleteWirelessDevice"
           ],
    "Resource": "*"
    }
 ]
}
```

このポリシーには、CreateWirelessDevice、GetWirelessDevice、ListWirelessDevices、UpdateWirelessDe および DeleteWirelessDevice アクションを使用するアクセス許可を付与するステートメントが 1 つあります。AWS IoT Wireless はこれらのメソッドを呼び出してワイヤレスデバイスを作成および管理します。

ID ベースのポリシーでアクセス許可を得るプリンシパルを指定していないため、ポリシーでは Principal エレメントを指定していません。ユーザーにポリシーをアタッチすると、そのユーザーが暗黙のプリンシパルになります。IAM ロールにアクセス権限ポリシーをアタッチすると、ロールの信頼ポリシーで識別されたプリンシパルがアクセス権限を得ることになります。

AWS の AWS IoT Wireless マネージドポリシー

ユーザー、グループ、ロールに権限を追加するには、自分でポリシーを作成するよりも、AWS マネージドポリシーを使用する方が簡単です。チームに必要な権限のみを提供する IAM カスタマーマネージドポリシーを作成するには、時間と専門知識が必要です。すぐに使用を開始するために、AWS マネージドポリシーを使用できます。これらのポリシーは、一般的なユースケースをターゲット範囲に含めており、AWS アカウント で利用できます。AWS マネージドポリシーの詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS のサービスは、AWS マネージドポリシーを維持および更新します。AWS マネージドポリシーの許可を変更することはできません。サービスでは、新しい機能を利用できるようにするために、AWS マネージドポリシーに権限が追加されることがあります。この種類の更新は、ポリシーがアタッチされている、すべてのアイデンティティ (ユーザー、グループおよびロール) に影響を与えます。新しい機能が立ち上げられた場合や、新しいオペレーションが使用可能になった場合に、各サービスが AWS マネージドポリシーを更新する可能性が最も高くなります。サービスは、AWS マネージドポリシーから権限を削除しないため、ポリシーの更新によって既存の権限が破棄されることはありません。

さらに、AWS は、複数のサービスにまたがるジョブ機能の特徴に対するマネージドポリシーもサポートしています。例えば、ReadOnlyAccess AWS マネージドポリシーでは、すべての AWS のサービスおよびリソースへの読み取り専用アクセスを許可します。サービスが新しい機能を起動する場合、AWS は、新たなオペレーションとリソース用に、読み取り専用の許可を追加します。ジョブ機能のポリシーの一覧および詳細については、「IAM ユーザーガイド」の「AWS のジョブ機能のマネージドポリシー」を参照してください。

AWS マネージドポリシー: AWSIoTWirelessDataAccess

`AWSIoTWirelessDataAccess` ポリシーは IAM ID にアタッチできます。

このポリシーは、関連付けられた ID アクセス許可を付与し、`SendDataToWirelessDevice` API を使用して LoRaWAN および Sidewalk デバイスにデータを送信するためのアクセスを許可しま

す。AWS Management Console でこのポリシーを表示するには、「AWSIoTWirelessDataAccess」を参照してください。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `iotwireless` – AWS IoT Wireless データを取得します。

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

AWS マネージドポリシー: AWSIoTWirelessFullAccess

`AWSIoTWirelessFullAccess` ポリシーは IAM ID にアタッチできます。

このポリシーは、すべての AWS IoT Wireless オペレーションへのアクセスを許可する、関連付けられた ID のアクセス許可を付与します。AWS Management Console でこのポリシーを表示するには、「AWSIoTWirelessFullAccess」を参照してください。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `iotwireless` – AWS IoT Wireless データを取得し、すべての AWS IoT Wireless オペレーションを実行します。

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

AWS マネージドポリシー: AWSIoTWirelessFullPublishAccess

`AWSIoTWirelessFullPublishAccess` ポリシーは IAM ID にアタッチできます。

このポリシーは、ユーザーに代わって AWS IoT ルールにパブリッシュするための制限付きアクセスを許可する、関連付けられた ID のアクセス許可を付与します。AWS Management Console のこのポリシーを表示するには、「AWSIoTWirelessFullPublishAccess」を参照してください。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `iot` – エンドポイント URL を取得し、AWS IoT ルールエンジンにパブリッシュするオペレーションを実行します。

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

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

AWS マネージドポリシー: AWSIoTWirelessLogging

`AWSIoTWirelessLogging` ポリシーは IAM ID にアタッチできます。

このポリシーは、Amazon CloudWatch Logs ロググループの作成を許可し、グループにログをストリーミングするための関連付けられた ID のアクセス許可を付与します。このポリシーは CloudWatch ログ記録用のロールにアタッチされます。AWS Management Console のこのポリシーを表示するには、「`AWSIoTWirelessLogging`」を参照してください。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `logs` — CloudWatch ログを取得します。また、CloudWatch Logs のグループを作成し、グループにログをストリーミングすることができます。

```
{
    "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 マネージドポリシー: AWSIoTWirelessReadOnlyAccess

AWSIoTLogging ポリシーは IAM ID にアタッチできます。

このポリシーは、すべての AWS IoT Wireless オペレーションへの読み取り専用アクセスを許可する、関連付けられた ID のアクセス許可を付与します。AWS Management Console のこのポリシーを表示するには、「AWSIoTWirelessReadOnlyAccess」を参照してください。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- logs – AWS IoT Wireless List および Get API オペレーションを実行します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iotwireless:List*",
                "iotwireless:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS マネージドポリシー: AWSIoTWirelessGatewayCertManager

AWSIoTWirelessGatewayCertManager ポリシーは IAM ID にアタッチできます。

このポリシーは、AWS IoT 証明書を作成、一覧表示、および記述することを許可する関連付けられた ID アクセス許可を付与します。AWS Management Console のこのポリシーを表示するには、「AWSIoTWirelessGatewayCertManager」を参照してください。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- iot – 証明書を作成、説明、一覧表示するアクションを実行します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IoTWirelessGatewayCertManager",
            "Effect": "Allow",
            "Action": [
                "iot:CreateKeysAndCertificate",
                "iot:DescribeCertificate",
                "iot:ListCertificates"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS IoT Wireless: AWS マネージドポリシーへの更新

このサービスがこれらの変更の追跡を開始してからの、AWS IoT Wireless の AWS マネージドポリシーの更新に関する詳細を表示します。このページの変更に関する自動通知については、「AWS IoT Wireless ドキュメント履歴ページ」ページで RSS フィードをサブスクライブしてください。

変更	説明	日付
AWS IoT Wireless は変更の追跡を開始しました	AWS IoT Wireless が AWS マネージドポリシーの変更の追跡を開始しました。	2022 年 5 月 18 日

AWS IoT Wireless の ID とアクセスのトラブルシューティング

次の情報は、AWS IoT Wireless と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

トピック

- AWS IoT Wireless でアクションを実行する権限がない
- マイアクセスキーを表示したい
- 管理者として AWS IoT Wireless へのアクセスを他のユーザーに許可する
- 自分の AWS アカウント以外のユーザーに AWS IoT Wireless リソースへのアクセスを許可したい

AWS IoT Wireless でアクションを実行する権限がない

AWS Management Console から、アクションを実行することが認可されていないと通知された場合、管理者に問い合わせ、サポートを依頼する必要があります。担当の管理者はお客様のユーザー名とパスワードを発行した人です。

以下の例のエラーは、mateojackson IAM ユーザーがコンソールを使用して、*#########*の詳細を表示しようとしているが、YOUR-SERVICE-PREFIX:*GetWirelessDevice* アクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: YOUR-
SERVICE-PREFIX:GetWirelessDevice on resource: my-LoRaWAN-device
```

この場合、Mateo は、YOUR-SERVICE-PREFIX:*GetWirelessDevice* アクションを使用して *my-LoRaWAN-device* リソースへのアクセスが許可されるように、管理者にポリシーの更新を依頼します。

マイアクセスキーを表示したい

IAM ユーザーアクセスキーを作成した後は、いつでもアクセスキー ID を表示できます。ただし、シークレットアクセスキーを再表示することはできません。シークレットアクセスキーを紛失した場合は、新しいアクセスキーペアを作成する必要があります。

アクセスキーは、アクセスキー ID (例: AKIAIOSFODNN7EXAMPLE) とシークレットアクセスキー (例: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY) の 2 つの部分で構成されています。ユーザー名とパスワードと同様に、リクエストを認証するために、アクセスキー ID とシークレットアクセスキーの両方を使用する必要があります。ユーザー名とパスワードと同様に、アクセスキーは安全に管理してください。

> **Important**
>
> 正規のユーザー ID を検索するためであっても、アクセスキーを第三者に提供しないでください。これを行うと、AWS アカウント への永続的なアクセス権が第三者に付与される可能性があります。

アクセスキーペアを作成する場合、アクセスキー ID とシークレットアクセスキーを安全な場所に保存するように求めるプロンプトが表示されます。このシークレットアクセスキーは、作成時にのみ使用できます。シークレットアクセスキーを紛失した場合、IAM ユーザーに新規アクセスキーを追加する必要があります。アクセスキーは最大 2 つまで持つことができます。既に 2 つある場合は、新規キーペアを作成する前に、いずれかを削除する必要があります。手順を表示するには、「IAM ユーザーガイド」の「アクセスキーの管理」を参照してください。

管理者として AWS IoT Wireless へのアクセスを他のユーザーに許可する

AWS IoT Wireless へのアクセスを他のユーザーに許可するには、アクセスを必要とする人またはアプリケーションの IAM エンティティ (ユーザーまたはロール) を作成する必要があります。ユーザーまたはアプリケーションは、そのエンティティの認証情報を使用して AWS にアクセスします。次に、AWS IoT Wireless の適切なアクセス許可を付与するポリシーを、そのエンティティにアタッチする必要があります。

すぐに開始するには、『IAM ユーザーガイド』の「Creating your first IAM delegated user and group (IAM が委任した最初のユーザーおよびグループの作成)」を参照してください。

自分の AWS アカウント以外のユーザーに AWS IoT Wireless リソースへのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセス制御リスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください。

- AWS IoT Wireless がこれらの機能をサポートしているかどうかを確認するには、「AWS IoT Wireless と IAM の連携方法」を参照してください。
- 所有している AWS アカウント 全体のリソースへのアクセス権を提供する方法については、『IAM ユーザーガイド』の「所有している別の AWS アカウント アカウントへのアクセス権を IAM ユーザーに提供」を参照してください。
- サードパーティーの AWS アカウント にリソースへのアクセス権を提供する方法については、「IAM ユーザーガイド」の「サードパーティーが所有する AWS アカウント にアクセス権を提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、「IAM ユーザーガイド」の「外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可」を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いの詳細については、『IAM ユーザーガイド』の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

AWS IoT Wireless のコンプライアンス検証

サードパーティーの監査者は、さまざまな AWS コンプライアンスプログラムの一環として AWS IoT Wireless のセキュリティとコンプライアンスを評価します。これらのプログラムには、SOC、PCI、FedRAMP、HIPAA などがあります。

特定のコンプライアンスプログラムの対象となる AWS サービスのリストについては、「コンプライアンスプログラムによる AWS 対象範囲内のサービス」を参照してください。一般的な情報については、「AWS コンプライアンスプログラム」を参照してください。

AWS Artifact を使用して、サードパーティーの監査レポートをダウンロードできます。詳細については、「Downloading Reports in AWS Artifact」を参照してください。

AWS IoT Wireless を使用する際のユーザーのコンプライアンス責任は、ユーザーのデータの機密性や貴社のコンプライアンス目的、適用される法律および規制によって決まります。AWS では、コンプライアンスに役立つ以下のリソースを用意しています。

- 「セキュリティとコンプライアンスのクイックスタートガイド」「」 – これらのデプロイガイドには、アーキテクチャ上の考慮事項の説明と、AWS でセキュリティとコンプライアンスに重点を置いたベースライン環境をデプロイするためのステップが記載されています。
- HIPAA セキュリティおよびコンプライアンスホワイトペーパーのアーキテクチャの設計 - このホワイトペーパーでは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。
- AWSコンプライアンスのリソース– このワークブックおよびガイドのコレクションは、お客様の業界と拠点に適用されるものである場合があります。
- AWS Configデベロッパーガイドのルールでのリソースの評価 – AWS Configは、リソース設定が、社内のプラクティス、業界のガイドラインそして規制にどの程度適合しているのかを評価します。
- AWS Security Hub– AWSのこのサービスは、 AWS内でのユーザーのセキュリティ状態に関する包括的な見解を提供し、業界のセキュリティ標準、およびベストプラクティスに対するコンプライアンスを確認するために役立ちます。

AWS IoT Wireless の耐障害性

AWS のグローバルインフラストラクチャは、AWS リージョンとアベイラビリティーゾーンを中心として構築されています。リージョンには、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立および隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョンとアベイラビリティーゾーンの詳細については、AWS グローバルインフラストラクチャを参照してください。

AWS IoT Wireless のインフラストラクチャセキュリティ

マネージドサービスとして、AWS IoT Wireless は、ホワイトペーパー「Amazon Web Services: セキュリティプロセスの概要」で説明されている AWS グローバルネットワークセキュリティ手順によって保護されています。

AWS がパブリッシュした API コールを使用して、ネットワーク経由で AWS IoT Wireless にアクセスします。クライアントで Transport Layer Security (TLS) 1.0 以降がサポートされている必要があります。TLS 1.2 以降が推奨されています。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token ServiceAWS STSを使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

Amazon CloudWatch Logs を使用した AWS IoT Wireless リソースのモニタリング

モニタリングは、AWS IoT Wireless および他の AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要です。LoRaWAN デバイスと Sidewalk デバイスの両方のモニタリングを使用して、AWS IoT Wireless にオンボーディングされたときに から情報メッセージとエラーを取得できます。

マルチポイント障害が発生した場合にデバッグしやすくなるように、AWS ソリューションのすべての部分から監視データを収集することを強くお勧めします。まず、以下の質問に答えて監視計画を作成します。どのように答えるべきかわからない場合でも、引き続き ログ記録を有効化して、パフォーマンスのベースラインを確立できます。

- どのような目的でモニタリングしますか?
- どのリソースをモニタリングしますか?
- どのくらいの頻度でこれらのリソースをモニタリングしますか?
- どのモニタリングツールを使用しますか?
- 誰がモニタリングタスクを実行しますか?
- 問題が発生したときに誰が通知を受け取りますか?

次のステップでは、ログ記録を有効化し、さまざまなタイミングと負荷条件でパフォーマンスを測定することにより、使用環境における AWS IoT Wireless の正常なパフォーマンスについて、ベースラインを確定します。AWS IoT Wireless をモニタリングする際には、現在のパフォーマンスデータと比較できるように、履歴監視データを保持します。これにより、通常のパフォーマンスパターンとパフォーマンスの異常を特定し、問題に対処するための方法を考えることができます。

モニタリングツール

AWS IoT Wireless を監視して異常を検出した場合に報告し、必要に応じて自動的に対処するために、次のモニタリングツールを使用することができます。

- Amazon CloudWatch は、AWS のリソースおよび AWS で実行しているアプリケーションをリアルタイムでモニタリングします。メトリクスの収集と追跡、カスタマイズしたダッシュボードの作成、および指定したメトリクスが指定したしきい値に達したときに通知またはアクションを実行す

るアラームの設定を行うことができます。例えば、CloudWatch で Amazon EC2 インスタンスの CPU 使用率などのメトリクスを追跡し、必要に応じて新しいインスタンスを自動的に起動できます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

- ネットワークアナライザを使用すると、LoRaWAN デバイスとゲートウェイを含む LoRaWAN リソースをモニタリングでき、接続のセットアップとトレースメッセージの受信開始にかかる時間が大幅に短縮され、ジャストインタイムのログ情報を取得できます。詳細については、「ネットワークアナライザを使用したワイヤレスリソースフリートのリアルタイムでのモニタリング」を参照してください。

Amazon CloudWatch を使用してリソースをモニタリングする方法

CloudWatch を使用して AWS IoT Wireless をモニタリングし、raw データを収集し、それを読み取り可能なほぼリアルタイムのメトリクスに処理します。これらの統計は 15 か月間保持されるため、履歴情報にアクセスし、ウェブアプリケーションまたはサービスの動作をより的確に把握できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

AWS IoT Wireless リソースのログ記録とモニタリングを行うには、以下のステップを実行します。

1. 「AWS IoT Wireless 用のログ記録ロールとポリシーを作成する」で説明されているようにログ記録ロールを作成し、AWS IoT Wireless リソースのログ記録を行います。
2. CloudWatch Logs コンソールのログメッセージには、デフォルトのログ記録レベル ERROR が設定されていますが、これはあまり詳細ではなく、エラー情報のみが含まれています。より詳細なメッセージを表示する場合は、「AWS IoT Wireless リソースのログ記録の設定」で説明されているように、最初に CLI を使用してログ記録を設定することをお勧めします。
3. 次に、CloudWatch Logs コンソールでログエントリを表示して、リソースをモニタリングできます。詳細については、「CloudWatch AWS IoT Wireless ログエントリの表示」を参照してください。
4. [Logs groups] (ロググループ) を使用してフィルター式を作成することができますが、まず単純なフィルターを作成し、ロググループ内のログエントリを表示してから、CloudWatch Insights に移動して、モニタリングするリソースまたはイベントに応じてログエントリをフィルターするクエリを作成することをお勧めします。詳細については、「CloudWatch Insights を使用して AWS IoT Wireless のログをフィルタリングする」を参照してください。

AWS IoT Wireless のログ記録の設定

AWS IoT アクティビティをモニタリングおよびログ記録する前に、まず CLI または API のいずれかを使用して、AWS IoT Wireless リソースのログ記録を有効にします。

AWS IoT Wireless ログ記録の設定方法を検討する場合、特に指定がない限り、AWS IoT アクティビティのログ記録方法はデフォルトのログ記録の設定によって決まります。最初は、デフォルトのログレベル (INFO) を使用して、詳細なログを取得できます。

初期ログを確認した後、デフォルトのログレベルを、あまり詳細ではない ERROR レベルに変更することができます。また、注意が必要なリソースに対しては、より詳細な、リソース固有のログレベルに設定できます。ログレベルはいつでも変更できます。

次のトピックでは、AWS IoT Wireless リソースのログ記録を設定する方法を示します。

トピック

AWS IoT Wireless 用のログ記録ロールとポリシーを作成する

次に、AWS IoT Wireless リソースのログ記録ロールを作成する方法を示します。AWS IoT Core のログ記録ロールも作成したい場合は、「https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html」を参照してください。

AWS IoT Wireless のログ記録ロールを作成する

ログ記録を有効にする前に、IAM ロールと、ユーザーに代わって AWS IoT Wireless アクティビティをモニタリングするための AWS アクセス許可を与えるポリシーを作成する必要があります。

ログ記録用の IAM ロールを作成する

AWS IoT Wireless のログ記録ロールを作成するには、IAM コンソールの [Roles] (ロール) ハブを開き、[Create role] (ロールの作成) を選択します。

1. [Select type of trusted entity](信頼できるエンティティのタイプを選択) で、[AnotherAWSaccount](別のアカウント) を選択します。
2. [Account ID] (アカウント ID) で AWS アカウント ID を入力し、[Next: Permissions] (次へ: アクセス許可) を選択します。

3. 検索ボックスに「**AWSIoTWirelessLogging**」と入力します。
4. AWSIoTWirelessLogging という名前のポリシーの横にあるボックスを選択し、[Next: Tags] (次へ: タグ) を選択します。
5. [Next: Review] を選択します。
6. [Role name] (ロール名) に **IoTWirelessLogsRole** と入力し、[Create role] (ロールの作成) を選択します。

IAM ロールの信頼関係を編集する

前のステップを実行した後に表示される確認メッセージで、作成したロールの名前である IoTWirelessLogsRole を選択します。次に、ロールを編集して、次の信頼関係を追加します。

1. IoTWirelessLogsRole ロールの Summary (概要) セクションで、[Trust relationships] (信頼関係) タブを選択し、続いて [Edit trust relationship] (信頼関係の編集) を選択します。
2. [Policy Document] (ポリシードキュメント) で、`Principal` プロパティを次の例のように変更します。

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

`Principal` プロパティを変更すると、完全なポリシードキュメントは次の例のようになります。

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

3. 変更を保存して終了するには、[Update Trust Policy] (信頼ポリシーの更新) を選択します。

AWS IoT Wireless のログ記録ポリシー

次のポリシードキュメントでは、お客様に代わって AWS IoT Wireless が CloudWatch にログエントリを送信できるようにするロールポリシーと信頼ポリシーを示しています。

> **Note**
>
> この AWS マネージドポリシードキュメントは、ログ記録ロール IoTWirelessLogsRole を作成したときに自動的に作成されました。

ロールポリシー

以下は、ロールポリシードキュメントを示しています。

```
{
    "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 IoT Wireless アクティビティのみを口グに記録する信頼ポリシー

次に、AWS IoT Wireless アクティビティのみを口グ記録するための信頼ポリシーを示します。

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

```
          "Effect": "Allow",
          "Principal": {
            "Service": [
            "iotwireless.amazonaws.com"
            ]
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
```

AWS IoT Core アクティビティもログ記録する IAM ロールを作成した場合、このポリシードキュメントで両方のアクティビティを記録できます。AWS IoT Core のログ記録ロールの作成の詳細については、「https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html」を参照してください。

次のステップ

AWS IoT Wireless リソースをログ記録するログ記録ロールを作成する方法を学習しました。デフォルトでは、ログに設定されているログレベルは ERROR なので、エラー情報のみを表示したい場合は CloudWatch AWS IoT Wireless ログエントリの表示 に移動して、ログエントリを表示してワイヤレスリソースをモニタリングします。

ログエントリにより多くの情報が必要な場合は、ログレベルを INFO に設定するなど、リソースまたはさまざまなイベントタイプに対してデフォルトのログレベルを構成できます。リソースのログ記録の構成に関する詳細は、「AWS IoT Wireless リソースのログ記録の設定」を参照してください。

AWS IoT Wireless リソースのログ記録の設定

AWS IoT Wireless リソースのログ記録を設定するには、API または CLI のいずれかを使用できます。AWS IoT Wireless リソースのモニタリングを開始する際、デフォルト設定を使用できます。これを行うには、このトピックをスキップして、CloudWatch Logs を使用して AWS IoT Wireless をモニタリングする に進んでログをモニタリングします。

ログのモニタリングを開始したら、CLI を使用して、ログレベルをより詳細なオプションに変更できます。例えば、INFO および ERROR の情報を指定したり、より多くのリソースに対するログ記録を有効にしたりするなどです。

AWS IoT Wireless リソースとログレベル

API または CLI を使用する前に、次の表を使用して、ログ記録を設定できるさまざまなログレベルとリソースについて学習します。この表は、リソースをモニタリングするときに CloudWatch ログに表示されるパラメータを示しています。リソースに対するログ記録の設定方法によって、コンソールに表示されるログが決まります。

サンプルの CloudWatch ログがどのように表示されるか、およびこれらのパラメータを使用して AWS IoT Wireless リソースに関する有益な情報をログ記録する方法ついては、「CloudWatch AWS IoT Wireless ログエントリの表示」を参照してください。

ログレベルとリソース

名前	使用できる値:	説明
`logLevel`	`INFO`, `ERROR`, または `DISABLED`	• `ERROR`: オペレーションの失敗につながるすべてのエラーを表示します。ログに含まれるのは `ERROR` の情報のみです。 • `INFO`: モノのフローに関する概要を提供します。ログに含まれるのは `INFO` および `ERROR` の情報です。 • `DISABLED`: すべてのログ記録を無効にします。
`resource`	`WirelessGateway` または `WirelessDevice`	リソースのタイプ。`WirelessGateway` または `WirelessDevice` です。
`wirelessGatewayType`	`LoRaWAN`	`resource` が `WirelessGateway` の場合の、ワイヤレスゲートウェイのタイプ。常に `LoRaWAN` です。
`wirelessDeviceType`	`LoRaWAN` または `Sidewalk`	`resource` が `WirelessDevice` の場合の、ワイヤレスデバイスのタイプ。`LoRaWAN` または `Sidewalk` です。
`wirelessGatewayId`	-	`resource` が `WirelessGateway` の場合の、ワイヤレスゲートウェイの ID。

名前	使用できる値:	説明
wirelessDeviceId	-	resource が WirelessDevice の場合の、ワイヤレスデバイスの ID。
event	Join、Rejoin、Registration 、Uplink_data 、Downlink_data 、CUPS_Request 、および Certificate	ログに記録されるイベントのタイプ。ログに記録するリソースがワイヤレスデバイスかワイヤレスゲートウェイかによって異なります。詳細については、「CloudWatch AWS IoT Wireless ログエントリの表示」を参照してください。

AWS IoT Wireless ログ記録 API

以下の API アクションを使用して、リソースのログ記録を設定できます。この表には、API アクションを使用するために作成する必要のあるサンプルの IAM ポリシーも示されています。以下のセクションでは、API を使用してリソースのログレベルを設定する方法について説明します。

API アクションのログ記録

API 名	説明	サンプルの IAM ポリシー
GetLogLevelsByResourceTypes	現在のデフォルトのログレベル、またはリソースタイプ別のログレベルを返します。ワイヤレスデバイスまたはワイヤレスゲートウェイのログオプションを含めることができます。	{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:GetLogLevelsByResourceTypes"], "Resource": [

API 名	説明	サンプルの IAM ポリシー

		<pre>"*"
		
		]
		}
		]
		}</pre>

GetResourceLogLevel	指定されたリソース IDとリソースタイプに対してログレベルのオーバーライドを返します。リソースは、ワイヤレスデバイスまたはワイヤレスゲートウェイにすることができます。	<pre>{
		"Version":
		"2012-10-17",
		"Statement": [
		
		{
		"Effect":
		"Allow",
		"Action": [
		
		"iotwireless:GetResourceLogLevel"
		
		],
		
		"Resource":
		[
		
		"arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc537-ab12-cd3a-d00e-1f0e20c1204a",
		]
		}
		]
		}</pre>

API 名	説明	サンプルの IAM ポリシー

PutResourceLogLevel	指定されたリソース IDとリソースタイプに対してログレベルのオーバーライドを設定します。リソースは、ワイヤレスゲートウェイまたはワイヤレスデバイスにすることができます。
	
	Note
	この API では、ログレベルのオーバーライドが 1 アカウントにつき 200 に制限されています。
		<pre>{
		    "Version":
		"2012-10-17",
		    "Statement": [
		
		        {
		            "Effect":
		"Allow",
		            "Action": [
		
		"iotwireless:PutRe
		sourceLogLevel"
		
		            ],
		
		            "Resource":
		[
		
		"arn:aws:iotwirele
		ss:us-east-1:12345
		6789012:WirelessDe
		vice/012bc537-ab12
		-cd3a-d00e-1f0e20c
		1204a",
		            ]
		        }
		    ]
		}</pre>

API 名	説明	サンプルの IAM ポリシー
ResetAllResourceLogLevels	ワイヤレスゲートウェイとワイヤレスデバイスの両方を含む、すべてのリソースに対するログレベルのオーバーライドを削除します。 **Note** この API は、UpdateLogLevelsByResourceTypes API を使用して設定されたログレベルには影響しません。	`{ "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/* ] } ] }`

API 名	説明	サンプルの IAM ポリシー

ResetResourceLogLevel	指定されたリソース ID とリソースタイプに対するログレベルのオーバーライドを削除します。リソースは、ワイヤレスゲートウェイまたはワイヤレスデバイスにすることができます。	{
		"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	説明	サンプルの IAM ポリシー

UpdateLogLevelsByResourceTypes	デフォルトのログレベル、またはリソースタイプ別のログレベルを設定します。この API は、ワイヤレスデバイスまたはワイヤレスゲートウェイのログオプションに使用でき、CloudWatch に表示されるログメッセージを制御できます。
	
	Note
	イベントはオプションで、イベントタイプはリソースタイプに関連付けられています。詳細については、「イベントタイプとリソースタイプ」を参照してください。
		<pre>{
		    "Version":
		"2012-10-17",
		    "Statement": [
		
		        {
		            "Effect":
		"Allow",
		            "Action": [
		
		"iotwireless:UpdateLogLevelsByResourceTypes"
		            ],
		
		            "Resource":
		[
		                "*"
		
		            ]
		        }
		    ]
		}</pre>

CLI を使用してリソースのログレベルを設定する

このセクションでは、API または AWS CLI を使用して AWS IoT Wireless リソースのログレベルを設定する方法について説明します。

CLI を使用する前に、以下を行ってください。

- 前述のように、CLI コマンドを実行する API の IAM ポリシーを作成していることを確認します。
- 使用するロールの Amazon リソースネーム (ARN) が必要です。ログ記録に使用するロールを作成する必要がある場合は、「AWS IoT Wireless 用のログ記録ロールとポリシーを作成する」を参照してください。

AWS CLI を使用する理由

デフォルトでは、「AWS IoT Wireless 用のログ記録ロールとポリシーを作成する」で説明されているように IAM ロール `IoTWirelessLogsRole` を作成した場合、デフォルトのログレベル `ERROR` を持つ CloudWatch ログが AWS Management Console に表示されます。すべてのリソースまたは特定のリソースのデフォルトのログレベルを変更するには、AWS IoT Wireless ログ記録 API または CLI を使用します。

AWS CLI の使用方法

API アクションは、すべてのリソースと特定のリソースのどちらに対してログレベルを設定するかに応じて、次のタイプに分類できます。

- API アクション `GetLogLevelsByResourceTypes` および `UpdateLogLevelsByResourceTypes` は、ワイヤレスゲートウェイ、LoRaWAN または Sidewalk デバイスなど、特定のタイプのアカウント内のすべてのリソースのログレベルを取得および更新できます。
- API アクション `GetResourceLogLevel`、`PutResourceLogLevel`、および `ResetResourceLogLevel` は、リソース ID を使用して、指定した個々のリソースのログレベルを取得、更新、リセットできます。
- API アクション `ResetAllResourceLogLevels` は、`PutResourceLogLevel` API を使用してログレベルのオーバーライドを指定したすべてのリソースに対して、ログレベルのオーバーライドを `null` にリセットします。

CLI を使用して AWS IoT でリソース固有のログ記録を設定するには

> **Note**
>
> この手順は、ここに示す CLI コマンドに対応する AWS API のメソッドを使用することにより、API で行うこともできます。

1. デフォルトでは、すべてのリソースのログレベルは `ERROR` に設定されています。アカウント内のすべてのリソースについて、デフォルトのログレベル、またはリソースタイプ別のログレベルを設定するには、update-log-levels-by-resource-types コマンドを使用します。次の例は、JSON ファイル `Input.json` を作成して、入力として CLI コマンドに提供する方法を示しています。このコマンドを使用すると、特定のタイプのリソースおよびイベントに対して、選択的にログ記録を無効にしたり、デフォルトのログレベルをオーバーライドしたりできます。

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

各パラメータの意味は次のとおりです。

WirelessDeviceLogOptions

ワイヤレスデバイスのログオプションのリスト。各ログオプションには、ワイヤレスデバイスのタイプ (Sidewalk または LoRaWAN) と、ワイヤレスデバイスのイベントログオプションのリストが含まれます。各ワイヤレスデバイスのイベントログオプションには、必要に応じて、イベントタイプとそのログレベルを含めることができます。

WirelessGatewayLogOptions

ワイヤレスゲートウェイのログオプションのリスト。各ログオプションには、ワイヤレスゲートウェイのタイプ (LoRaWAN) と、ワイヤレスゲートウェイのイベントログオプションのリストが含まれます。各ワイヤレスゲートウェイのイベントログオプションには、必要に応じて、イベントタイプとそのログレベルを含めることができます。

DefaultLogLevel

すべてのリソースに使用するログレベル。有効な値は、ERROR、INFO、DISABLED です。デフォルト値は INFO です。

LogLevel

個々のリソースタイプとイベントに使用するログレベル。これらのログレベルは、LoRaWAN ゲートウェイに対するログレベル INFO や、2 つのイベントタイプに対する DISABLED および ERROR などのデフォルトのログレベルをオーバーライドします。

次のコマンドを実行すると、Input.json ファイルが入力としてコマンドに提供されます。このコマンドでは、出力が生成されません。

```
aws iotwireless update-log-levels-by-resource-types \
    --cli-input-json Input.json
```

ワイヤレスデバイスとワイヤレスゲートウェイの両方のログオプションを削除する場合は、次のコマンドを実行します。

```
{
    "DefaultLogLevel":"DISABLED",
    "WirelessDeviceLogOptions": [],
    "WireslessGatewayLogOptions":[]
}
```

2. update-log-levels-by-resource-types コマンドは出力を返しません。リソース固有のログ記録情報を取得するには、get-log-levels-by-resource-types コマンドを使用します。このコマンドは、デフォルトのログレベルと、ワイヤレスデバイスおよびワイヤレスゲートウェイのログオプションを返します。

> **Note**
>
> get-log-levels-by-resource-types コマンドは、CloudWatch コンソールでログレベルを直接取得することはできません。get-log-levels-by-resource-types コマンドは、update-log-levels-by-resource-types コマンドを使用してリソースに対して指定した最新のログレベルの情報を取得するために使用できます。

```
aws iotwireless get-log-levels-by-resource-types
```

次のコマンドを実行すると、update-log-levels-by-resource-types を使用して指定した最新のログ記録情報が返されます。例えば、ワイヤレスデバイスのログオプションを削除した場合、get-log-levels-by-resource-types を実行すると、この値が `null` として返されます。

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions": null,
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
```

```
            "Event": "CUPS_Request",
            "LogLevel": "DISABLED"
          },
          {
            "Event": "Certificate",
            "LogLevel": "ERROR"
          }
        ]
      }
    ]
}
```

3. 個々のワイヤレスゲートウェイまたはワイヤレスデバイスリソースのログレベルを制御するには、次の CLI コマンドを使用します。

   - put-resource-log-level
   - get-resource-log-level
   - reset-resource-log-level

   これらの CLI をいつ使用するかの例として、ログに記録されているアカウント内に多数のワイヤレスデバイスまたはワイヤレスゲートウェイがあるとします。一部のワイヤレスデバイスに対してのみエラーのトラブルシューティングを行う場合は、`DefaultLogLevel` を `DISABLED` に設定してすべてのワイヤレスデバイスのログ記録を無効にし、put-resource-log-level を使用して、アカウント内のこれらのデバイスに対してのみ `LogLevel` を `ERROR` に設定できます。

   ```
   aws iotwireless put-resource-log-level \
       --resource-identifier
       --resource-type WirelessDevice
       --log-level ERROR
   ```

   この例では、コマンドによって、指定したワイヤレスデバイスのリソースに対してのみログレベルが `ERROR` に設定され、その他のすべてのリソースのログが無効になります。このコマンドでは、出力が生成されません。この情報を取得して、ログレベルが設定されたことを確認するには、get-resource-log-level コマンドを使用します。

4. 前のステップでは、問題をデバッグしてエラーを解決した後、reset-resource-log-level コマンドを実行して、そのリソースのログレベルを `null` にリセットできます。`put-resource-log-level` コマンドを使用して、複数のワイヤレスデバイスまたはワイヤレスゲートウェイリソースに対してログレベルのオーバーライドを設定した場合 (複数のデバイスのエラーのトラブル

シューティングなど)、reset-all-resource-log-levels コマンドを使用して、これらのすべてのリソースに対して、ログレベルのオーバーライドを null にリセットできます。

```
aws iotwireless reset-all-resource-log-levels
```

このコマンドでは、出力が生成されません。リソースのログ記録情報を取得するには、get-resource-log-level コマンドを実行します。

次のステップ

ログ記録ロールを作成し、AWS IoT Wireless API を使用して AWS IoT Core for LoRaWAN リソースのログ記録を設定する方法を学習しました。続いてログエントリのモニタリングについて学習するには、CloudWatch Logs を使用して AWS IoT Wireless をモニタリングする に移動してください。

CloudWatch Logs を使用して AWS IoT Wireless をモニタリングする

AWS IoT Core for LoRaWAN には、デフォルトで有効になっている CloudWatch のログエントリが 50 以上あります。各ログエントリで、イベントタイプ、ログレベル、リソースタイプが記載されています。詳細については、「AWS IoT Wireless リソースとログレベル」を参照してください。

AWS IoT Wireless リソースのモニタリング方法

AWS IoT Wireless に対してログ記録が有効になっている場合、各メッセージが AWS IoT を経由してデバイスから渡される際には、AWS IoT Wireless により各メッセージに関する進行状況イベントが送信されます。デフォルトでは、AWS IoT Wireless のログエントリにエラーのデフォルトのログレベルがあります。「AWS IoT Wireless 用の口グ記録ロールとポリシーを作成する」で説明されているようにログ記録を有効にすると、デフォルトのログレベルが ERROR であるメッセージが CloudWatch コンソールに表示されます。このログレベルを使用すると、メッセージには、使用しているすべてのワイヤレスデバイスとワイヤレスゲートウェイのリソースについて、エラー情報のみが表示されます。

ログでその他の情報 (ログレベルが INFO であるなど) を表示したり、一部のデバイスのログを無効にしたり、一部のデバイスのログメッセージのみを表示する場合は、AWS IoT Wireless ログ記録 API を使用できます。詳細については、「CLI を使用してリソースのログレベルを設定する」を参照してください。

フィルター式を作成して、必要なメッセージのみを表示することもできます。

コンソールで AWS IoT Wireless のログを表示する前に

/aws/iotwireless ロググループが CloudWatch コンソールに表示されるようにするには、次の操作が完了している必要があります。

- AWS IoT Wireless でログ記録を有効にします。AWS IoT Wireless でログ記録を有効にする方法の詳細については、「AWS IoT Wireless のログ記録の設定」を参照してください。
- AWS IoT Wireless オペレーションを実行して、ログエントリを書き込みます。

フィルター式をより効果的に作成および使用するには、以下のトピックで説明されているように CloudWatch インサイトを使用することをお勧めします。また、ここに示されている順序でトピックに従うことをお勧めします。これは、最初に CloudWatch ロググループを使用して、コンソールでログエントリを表示するために使用できるさまざまなタイプのリソース、イベントタイプ、ログレベルについて学習するうえで役立ちます。それから、CloudWatch Insights を使用してフィルター式を作成する方法を学習し、リソースからさらに役立つ情報を得ることができます。

トピック

- CloudWatch AWS IoT Wireless ログエントリの表示
- CloudWatch Insights を使用して AWS IoT Wireless のログをフィルタリングする

CloudWatch AWS IoT Wireless ログエントリの表示

AWS IoT Wireless 用のログ記録ロールとポリシーを作成する で説明されているように AWS IoT Wireless のログ記録を設定し、いくつかのログエントリを書き込んだ後、次の手順を実行して CloudWatch コンソールでログエントリを表示できます。

CloudWatch ロググループコンソールでの AWS IoT ログの表示

CloudWatch コンソールで、CloudWatch のログは /aws/iotwireless という名前のロググループに表示されます。CloudWatch Logs の詳細については、CloudWatch Logs を参照してください。

CloudWatch コンソールでAWS IoT ログを表示するには

CloudWatch コンソールに移動して、ナビゲーションペインで [Log groups] (ロググループ) を選択します。

1. [Filter] (フィルター) テキストボックスに **`/aws/iotwireless`** と入力して、`/aws/iotwireless` ロググループを選択します。

2. アカウントに対して生成された AWS IoT Core for LoRaWAN ログの完全なリストを表示するには、[Search all] (すべて検索) を選択します。個々のログストリームを表示するには、展開アイコンを選択します。
3. ログストリームをフィルターするには、[Filter events] (イベントをフィルター) テキストボックスにクエリを入力することもできます。試すべきクエリの例がいくつかあります。

   - `{ $.logLevel = "ERROR" }`

     このフィルターを使用すると、ログレベルが ERROR であるすべてのログを検索して、個々のエラーストリームを展開してエラーメッセージを読み取ることができます。これは、エラーメッセージの解決に役立ちます。

   - `{ $.resource = "WirelessGateway" }`

     ログレベルに関係なく、WirelessGateway リソースのすべてのログを検索します。

   - `{ $.event = "CUPS_Request" && $.logLevel = "ERROR" }`

     イベントタイプ CUPS_Request で、ログレベルが ERROR であるすべてのログを検索します。

イベントタイプとリソースタイプ

次の表に、ログエントリが表示されるさまざまなイベントのタイプを示します。イベントタイプは、リソースタイプがワイヤレスデバイスかワイヤレスゲートウェイかによっても異なります。リソースおよびイベントタイプにデフォルトのログレベルを使用することも、それぞれのログレベルを指定することでデフォルトのログレベルをオーバーライドすることもできます。

使用されるリソースに基づくイベントタイプ

リソース	リソースタイプ	イベントタイプ
ワイヤレスゲートウェイ	LoRaWAN	• CUPS_Request • 証明書
ワイヤレスデバイス	LoRaWAN	• 参加 • Rejoin • Uplink_Data • Downlink_Data

リソース	リソースタイプ	イベントタイプ	
ワイヤレスデバイス	Sidewalk	• 登録 • Uplink_Data • Downlink_Data	

次のトピックでは、これらのイベントタイプと、ワイヤレスゲートウェイおよびワイヤレスデバイスのログエントリについて詳しく説明します。

トピック

- ワイヤレスゲートウェイとワイヤレスデバイスのリソースのログエントリ

ワイヤレスゲートウェイとワイヤレスデバイスのリソースのログエントリ

ログ記録を有効にすると、ワイヤレスゲートウェイとワイヤレスデバイスのログエントリを表示できます。次のセクションでは、リソースタイプとイベントタイプに基づくさまざまな種類のログエントリについて説明します。

ワイヤレスゲートウェイのログエントリ

このセクションでは、CloudWatch コンソールに表示されるワイヤレスゲートウェイリソースのサンプルログエントリの一部を紹介します。これらのログメッセージにはイベントタイプ `CUPS_Request` または `Certificate` を含めることができ、ログレベル `INFO`、`ERROR`、または `DISABLED` をリソースレベルまたはイベントレベルで表示するように構成できます。エラー情報のみを表示する場合は、ログレベルを `ERROR` に設定します。`ERROR` ログエントリ内のメッセージには、失敗した理由に関する情報が含まれます。

ワイヤレスゲートウェイリソースのログエントリは、次のイベントタイプに基づいて分類できます。

- CUPS_Request

  ゲートウェイで実行されている LoRa Basics Station は、定期的に更新のためのリクエストを Configuration and Update Server (CUPS) に送信します。このイベントタイプの場合、ワイヤレスゲートウェイリソース用に CLI を構成するときにログレベルを `INFO` に設定すると、ログには次のように表示されます。

  - イベントが成功すると、`logLevel` が `INFO` であるログメッセージが表示されます。メッセージには、ゲートウェイに送信された CUPS 応答の詳細と、ゲートウェイの詳細が含まれます。

このログエントリの例を以下に示します。ログエントリ内の logLevel およびその他のフィールドに関する詳細は、「AWS IoT Wireless リソースとログレベル」を参照してください。

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "fefffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "INFO",
    "message": "Sending CUPS response of total length 3213 to GatewayEui:
 fefffff00000000e2 with  TC Credentials,"
}
```

- エラーがある場合、logLevel が ERROR であるログエントリが表示され、メッセージにはエラーに関する詳細が含まれます。CUPS_Request イベントでエラーが発生する場合の例としては、CUPS CRC の欠落、ゲートウェイの TC Uri と AWS IoT Core for LoRaWAN の不一致、IoTWirelessGatewayCertManagerRole の欠落、ワイヤレスゲートウェイレコードを取得できなかったことなどがあります。次の例は、CRC 欠落のログエントリを示しています。エラーを解決するには、ゲートウェイ設定をチェックして、正しい CUPS CRC を入力したことを確認します。

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "fefffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "ERROR",
    "message": "The CUPS CRC is missing from the request. Check your gateway setup
 and enter the CUPS CRC,"
}
```

- 証明書

これらのログエントリは、AWS IoT への接続を認証するためにワイヤレスゲートウェイが正しい証明書を提供したかどうかをチェックするのに役立ちます。このイベントタイプの場合、ワイヤレスゲートウェイリソース用に CLI を構成するときにログレベルを INFO に設定すると、ログには次のように表示されます。

- イベントが成功すると、logLevel が INFO であるログメッセージが表示されます。メッセージには、証明書 ID とワイヤレスゲートウェイ ID に関する詳細が含まれます。このログエントリの例を以下に示します。ログエントリ内の logLevel およびその他のフィールドに関する詳細は、「AWS IoT Wireless リソースとログレベル」を参照してください。

```
{
    "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)"
}
```

- エラーがある場合、logLevel が ERROR であるログエントリが表示され、メッセージにはエラーに関する詳細が含まれます。Certificate イベントでエラーが発生する場合の例としては、証明書 ID やワイヤレスゲートウェイ ID が無効であること、またはワイヤレスゲートウェイ ID と証明書 IDの不一致などがあります。次の例は、無効なワイヤレスゲートウェイ ID が原因の ERROR を示しています。エラーを解決するには、ゲートウェイ ID を確認します。

```
{
    "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)"
}
```

ワイヤレスデバイスのログエントリ

このセクションでは、CloudWatch コンソールに表示されるワイヤレスデバイスリソースのサンプルログエントリの一部を紹介します。これらのログメッセージのイベントタイプは、LoRaWAN デバイスと Sidewalk デバイスのどちらを使用しているかによって異なります。各ワイヤレスデバイスリ

ソースまたはイベントタイプは、INFO、ERROR、またはDISABLED のログレベルを表示するように構成できます。

> **Note**
>
> リクエストに、LoRaWAN と Sidewalk の両方のワイヤレスメタデータを同時に含めてはなりません。このシナリオで ERROR ログエントリを回避するには、LoRaWAN または Sidewalk のいずれかのワイヤレスデータを指定してください。

LoRaWAN デバイスログエントリ

LoRaWAN ワイヤレスデバイスのログエントリは、次のイベントタイプに基づいて分類できます。

- **Join および Rejoin**

  LoRaWAN デバイスを追加して AWS IoT Core for LoRaWAN に接続する際、デバイスからアップリンクデータを送信できるようにするには、activation または join procedure と呼ばれるプロセスを完了する必要があります。詳細については、「ワイヤレスデバイスを AWS IoT Core for LoRaWAN に追加する」を参照してください。

  このイベントタイプの場合、ワイヤレスゲートウェイリソース用に CLI を構成するときにログレベルを INFO に設定すると、ログには次のように表示されます。

  - イベントが成功すると、logLevel が INFO であるログメッセージが表示されます。メッセージには、参加または再参加リクエストのステータスに関する詳細が含まれます。このログエントリの例を以下に示します。ログエントリ内の logLevel およびその他のフィールドに関する詳細は、「AWS IoT Wireless リソースとログレベル」を参照してください。

    ```
    {
        "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"
    }
    ```

- エラーがある場合、logLevel が ERROR であるログエントリが表示され、メッセージにはエラーに関する詳細が含まれます。Join および Rejoin イベントでエラーが発生する場合の例としては、無効な LoRaWAN リージョン設定、または無効な Message Integrity Code (MIC) などがあります。次の例は、MIC チェックが原因の参加エラーを示しています。エラーを解決するには、正しいルートキーを入力したかどうかを確認します。

```
{
    "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 および Downlink_Data

  イベントタイプ Uplink_Data は、ペイロードが LoRaWAN または Sidewalk デバイスから AWS IoT に送られたときに AWS IoT Wireless によって生成されたメッセージに使用されます。イベントタイプ Downlink_Data は、AWS IoT からワイヤレスデバイスに送信されたダウンリンクメッセージに関連するメッセージに使用されます。

  このイベントタイプの場合、ワイヤレスデバイス用に CLI を構成するときにログレベルを INFO に設定すると、ログには次のように表示されます。

  - イベントが成功すると、logLevel が INFO であるログメッセージが表示されます。メッセージには、送信されたアップリンクメッセージまたはダウンリンクメッセージのステータスに関する詳細と、ワイヤレスデバイス ID が含まれます。Sidewalk デバイスのこのログエントリの例を次に示します。ログエントリ内の logLevel およびその他のフィールドに関する詳細は、「AWS IoT Wireless リソースとログレベル」を参照してください。

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

- エラーがある場合、logLevel が ERROR であるログエントリが表示されます。メッセージにはエラーの詳細が含まれており、解決に役立ちます。Registration イベントでエラーが発生する場合の例としては、認証の問題、リクエストが無効であるか多すぎること、ペイロードの暗号化または復号化ができなかったこと、指定された ID を使用してワイヤレスデバイスを検出できなかったことなどがあります。次の例は、メッセージの処理中に発生したアクセス許可エラーを示しています。

```
{
    "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 デバイスのログエントリ

Sidewalk デバイスのログエントリは、次のイベントタイプに基づいて分類できます。

- **Registration**

  これらのログエントリは、AWS IoT Wireless に登録している任意の Sidewalk デバイスのステータスをモニタリングするのに役立ちます。このイベントタイプの場合、ワイヤレスデバイスリソース用に CLI を構成するときにログレベルを INFO に設定すると、logLevel が INFO および ERROR であるログメッセージが表示されます。メッセージには、開始から完了までの登録の進行状況に関する詳細が含まれます。ERROR ログメッセージには、デバイスの登録に関する問題のトラブルシューティング方法に関する情報が含まれます。

以下は、ログレベルが INFO のログメッセージの例を示しています。ログエントリ内の logLevel およびその他のフィールドに関する詳細は、「AWS IoT Wireless リソースとログレベル」を参照してください。

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "8d0b2775-e19b-4b2a-a351-cb8a2734a504",
    "wirelessDeviceType": "Sidewalk",
    "event": "Registration",
    "logLevel": "INFO",
    "message": "Successfully completed device registration. Amazon SidewalkId =
 2000000002"
}
```

- Uplink_Data および Downlink_Data

  イベントタイプ Uplink_Data および Downlink_Data は、LoRaWAN デバイスの対応するイベントタイプと同様です。詳細については、LoRaWAN デバイスのログエントリに関して前述された「Uplink_Data および Downlink_Data」セクションを参照してください。

次のステップ

AWS IoT Wireless へのログ記録を有効にした後に CloudWatch コンソールでリソースのログエントリを表示する方法と、表示できるさまざまなログエントリについて学習しました。ロググループを使用してフィルターストリームを作成することができますが、CloudWatch Insights を使用して、フィルターストリームを作成および使用することをお勧めします。詳細については、「CloudWatch Insights を使用して AWS IoT Wireless のログをフィルタリングする」を参照してください。

CloudWatch Insights を使用して AWS IoT Wireless のログをフィルタリングする

CloudWatch Logs を使用してフィルター式を作成することもできますが、CloudWatch Insights を使用して、アプリケーションに応じてフィルター式をより効果的に作成し、使用することをお勧めします。

コンソールでログエントリを表示するために使用できるさまざまなタイプのリソース、イベントタイプ、ログレベルについて学習するために、最初に CloudWatch ロググループを使用することをお勧めします。その後、このページに記載されたいくつかのフィルター式の例を参照として使用して、AWS IoT Wireless リソースの独自のフィルターを作成できます。

CloudWatch Logs insights コンソールでの AWS IoT ログの表示

CloudWatch コンソールで、CloudWatch のログは /aws/iotwireless という名前のロググループに表示されます。CloudWatch Logs の詳細については、CloudWatch Logs を参照してください。

CloudWatch コンソールで AWS IoT ログを表示するには

CloudWatch コンソールに移動して、ナビゲーションペインで [Log Insights] (ログインサイト) を選択します。

1. [Filter] (フィルター) テキストボックスに **/aws/iotwireless** と入力して、/aws/iotwireless Logs Insights を選択します。
2. ロググループの完全なリストを表示するには、[Select log group(s)] (ロググループを選択) を選択します。AWS IoT Wireless のロググループを表示するには、/aws/iotwireless を選択します。

これで、ロググループをフィルターするためのクエリの入力を開始できます。以下のセクションでは、リソースメトリクスに関するインサイトを得るのに役立つ、いくつかの便利なクエリについて説明します。

フィルタリングするための便利なクエリを作成して AWS IoT Wireless のインサイトを得る

フィルター式を使用して、CloudWatch Insights で追加の有用なログ情報を表示できます。以下に、サンプルのクエリをいくつか示します。

特定のリソースタイプのログのみを表示する

LoRaWAN ゲートウェイや Sidewalk デバイスなど、特定のリソースタイプのログのみを表示するのに役立つクエリを作成できます。例えば、Sidewalk デバイスのメッセージのみを表示するようにログをフィルターするには、次のクエリを入力し、[Run query] (クエリの実行) を選択します。このクエリを保存するには、[保存] を選択します。

```
fields @message
| filter @message like /Sidewalk/
```

クエリの実行後、[Logs] (ログ) タブに結果が表示され、アカウント内の Sidewalk デバイスに関連するログのタイムスタンプが表示されます。また、Sidewalk デバイスに関して以前発生したイベントがある場合は、そのイベントの発生時刻を示す棒グラフも表示されます。以下に、[Logs] (ログ) タ

ブのいずれかの結果を展開した場合の例を示します。また、Sidewalk デバイスに関連するエラーのトラブルシューティングを行う場合は、ログレベルを ERROR に設定してエラー情報のみを表示するフィルターを追加できます。

```
Field           Value
@ingestionTime      1623894967640
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-715adccfb34170214ec2f6667ddfa13cb5af2c3ddfc52fbeee0e554a2e780bed
@message      {
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "Sidewalk",
                    "devEui": "fefffff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Successfully sent downlink message. Amazon SidewalkId =
 2000000006, Sequence number = 0"
                    }
@timestamp          1623894967640
devEui            fefffff000000011a
event         Downlink_Data
logLevel            INFO
message           Successfully sent downlink message. Amazon SidewalkId = 2000000006,
 Sequence number = 0
messageId     7e752a10-28f5-45a5-923f-6fa7133fedda
resource      WirelessDevice
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  Sidewalk
```

特定のメッセージまたはイベントを表示する

特定のメッセージを表示し、いつイベントが発生したかを確認するのに役立つクエリを作成できます。例えば、LoRaWAN ワイヤレスデバイスからダウンリンクメッセージがいつ送信されたかを確認するには、次のクエリを入力し、[Run query] (クエリの実行) を選択します。このクエリを保存するには、[保存] を選択します。

```
filter @message like /Downlink message sent/
```

クエリの実行後、[Logs] (ログ) タブに結果が表示され、ダウンリンクメッセージがワイヤレスデバイスに正常に送信された時刻のタイムスタンプが表示されます。また、以前ワイヤレスデバイスに送信

されたダウンリンクメッセージがある場合、ダウンリンクメッセージが送信された時刻を示す棒グラフも表示されます。以下に、[Logs] (ログ) タブのいずれかの結果を展開した場合の例を示します。ダウンリンクメッセージが送信されなかった場合は、問題をデバッグできるように、メッセージが送信されなかったときの結果のみが表示されるようにクエリを変更できます。

```
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
```

次のステップ

CloudWatch Insights を使用してログメッセージをフィルターするクエリを作成することで、より役立つ情報を取得する方法を学びました。前述のフィルターをいくつか組み合わせて、モニタリングするリソースに応じて独自のフィルタを設計できます。CloudWatch Logs Insights の詳細については、「CloudWatch Logs Insights を使用したログデータの分析」を参照してください。

CloudWatch Insights を使用してクエリを作成した後に保存した場合、必要に応じて、保存されたクエリをロードして実行できます。または、CloudWatch Logs Insights コンソールの [History] (履歴) ボタンをクリックすると、以前に実行したクエリを表示して必要に応じて再実行したり、追加のクエリを作成してさらに変更したりできます。

AWS IoT Wireless イベント通知

AWS IoT Wireless では、AWS IoT Core にオンボードした LoRaWAN および Sidewalk デバイスのイベントを通知させるメッセージを発行できます。例えば、アカウント内の Sidewalk デバイスがプロビジョニングまたは登録されたときなどのイベントの通知を受けることができます。

イベントをリソースに通知する方法

イベント通知は、特定のイベントが発生したときに発行されます。例えば、Sidewalk デバイスのプロビジョニング時にイベントが生成されます。各イベントによって、単一のイベント通知が送信されます。イベント通知は、MQTT を介して JSON ペイロードを使用して発行されます。ペイロードのコンテンツは、イベントの種類によって異なります。

> **Note**
>
> イベント通知は少なくとも 1 回発行されます。複数回発行されることもあります。イベント通知の順序は保証されません。

イベントタイプとリソースタイプ

次の表に、通知を受け取るさまざまなイベントのタイプを示します。イベントタイプは、リソースタイプがワイヤレスデバイス、ワイヤレスゲートウェイまたは Sidewalk アカウントのいずれであるかによって異なります。次のセクションで説明するように、特定のタイプのすべてのリソースに適用されるリソースレベルで、または選択したリソースに対して、リソースのイベントを有効にすることもできます。さまざまなイベントタイプの詳細については、「LoRaWAN リソースのイベント通知」および「Sidewalk リソースのイベント通知」を参照してください。

リソースに基づくイベントタイプ

リソース	リソースタイプ	イベントタイプ	
ワイヤレスデバイス	LoRaWAN	参加	
	Sidewalk	• デバイス登録状態 • 近接	

リソース	リソースタイプ	イベントタイプ	
ワイヤレスゲートウェイ	LoRaWAN	接続ステータス	
Sidewalk アカウント	Sidewalk	• デバイス登録状態 • 近接	

ワイヤレスイベント通知を受信するためのポリシー

イベント通知を受信するには、デバイスが AWS IoT デバイスゲートウェイに接続し、MQTT イベントトピックをサブスクライブできるようにする適切なポリシーを使用する必要があります。また、適切なトピックフィルターを受信登録する必要があります。

以下は、さまざまなワイヤレスイベントの通知を受信するために必要なポリシーの例です。

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

ワイヤレスイベントの MQTT トピックの形式

ワイヤレスリソースのイベントの通知を送信するために、AWS IoT はドル記号 ($) で始まる MQTT 予約済みトピックを使用します。これらの予約済みのトピックを発行しサブスクライブできます。ただし、ドル記号で始まる新しいトピックを作成することはできません。

> **Note**
>
> MQTT トピックは、AWS アカウント に固有で、`arn:aws:iotwireless:aws-region:AWS-account-ID:topic/Topic` の形式を使用します。詳細については、AWS IoT デベロッパーガイドの「MQTT トピック」を参照してください。

ワイヤレスデバイス用に予約された MQTT トピックは、次の形式を使用します。

- リソースレベルのトピック

  これらのトピックは、AWS IoT Wireless にオンボーディングした AWS アカウント の特定のタイプのすべてのリソースに適用されます。

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/resources
  ```

- 識別子レベルのトピック

  これらのトピックは、リソース識別子で指定された、AWS IoT Wireless にオンボーディングした AWS アカウント の特定のタイプの選択されたリソースに適用されます。

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/
  {resourceIdentifierType}/{resourceID}/{id}
  ```

リソースレベルおよび識別子レベルのトピックの詳細については、「イベント設定」を参照してください。

次の表は、さまざまなイベントの MQTT トピックの例を示しています。

イベントと MQTT トピック

イベント	MQTT トピック	メモ
Sidewalk デバイスの登録状態	• リソースレベルのトピック `$aws/iotwireless/events/device_registration_state/`	• `{eventType}` は、`registered` か `provisioned` のいずれかとなります。 • `{resourceType}` は、`sidewalk_accounts` か `wireless_devices` のいずれかとなります。 • `{resourceID}` は、`sidewalk_accounts` の場合は `amazon_id`

イベント	MQTT トピック	メモ
	{eventType}/sidewalk/wireless_devices • 識別子レベルのトピック $aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}	、wireless_devices の場合は wireless_device_id となります。
Sidewalk 近接	• リソースレベルのトピック $aws/iotwireless/events/proximity/{eventType}/sidewalk/wireless_devices • 識別子レベルのトピック $aws/iotwireless/events/proximity/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}	• {eventType} は、beacon_discovered か beacon_lost のいずれかとなります。 • {resourceType} は、sidewalk_accounts か wireless_devices のいずれかとなります。 • {resourceID} は、sidewalk_accounts の場合は amazon_id 、wireless_devices の場合は wireless_device_id となります。

イベント	MQTT トピック	メモ

LoRaWAN 参加	• リソースレベルのトピック
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices`
	• 識別子レベルのトピック
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices/{resourceID}/{id}`
		• `{eventType}` は、`join_req_0_received` 、`join_req_2_received` 、`join_accepted` のいずれかとなります。
		• `{resourceID}` は、`wireless_device_id` か `dev_eui` のいずれかとなります。

LoRaWAN ゲートウェイ接続ステータス	• リソースレベルのトピック
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways`
	• 識別子レベルのトピック
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways/{resourceID}/{id}`
		• `{eventType}` は、`connected` か `disconnected` のいずれかとなります。
		• `{resourceID}` は、`wireless_gateway_id` か `gateway_eui` のいずれかとなります。

さまざまなイベントの詳細については、「LoRaWAN リソースのイベント通知」および「Sidewalk リソースのイベント通知」を参照してください。

これらのトピックをサブスクライブしている場合は、いずれかのイベント通知トピックにメッセージが発行されると通知されます。詳細については、AWS IoT デベロッパーガイドの「MQTT 予約済みトピック」を参照してください。

ワイヤレスイベントの料金

イベントのサブスクリプションと通知の受信に関する料金については、「AWS IoT Core の料金」を参照してください。

ワイヤレスリソースのイベントを有効にする

予約済みトピックのサブスクライバーがメッセージを受信できるようにするには、イベント通知を有効にする必要があります。これを行うには、AWS Management Console、AWS IoT Wireless API、または AWS CLI が使用できます。

イベント設定

特定のタイプに属するすべてのリソース、または個別のワイヤレスリソースのいずれかに通知を送信するようにイベントを設定できます。リソースタイプには、ワイヤレスゲートウェイ、Sidewalk パートナーアカウント、またはワイヤレスデバイス (LoRaWAN または Sidewalk デバイス) を使用できます。ワイヤレスデバイスで有効にできるイベントのタイプについては、「LoRaWAN リソースのイベントタイプ」および「Sidewalk リソースのイベントタイプ」を参照してください。

すべてのリソース

特定のリソースタイプに属する AWS アカウント 内のすべてのリソースが通知を受信するようにイベントを有効にすることができます。たとえば、AWS IoT Core for LoRaWAN でオンボーディングしたすべての LoRaWAN ゲートウェイの接続ステータスの変更を通知するイベントを有効にできます。これらのイベントをモニタリングすると、リソースフリート内の特定の LoRaWAN ゲートウェイが切断された場合や、AWS アカウント 内の多数の Sidewalk デバイスのビーコンが失われた場合などに通知を受け取るのに役立ちます。

個々のリソース

イベント設定に個々の LoRaWAN リソースおよび Sidewalk リソースを追加して、それらの通知を有効にすることもできます。これは、特定のタイプの個々のリソースをモニタリングするのに役立ちま

す。たとえば、選択した LoRaWANn および Sidewalk デバイスを設定に追加し、これらのリソースの参加またはデバイス登録状態イベントの通知を受信できます。

前提条件

LoRaWAN または Sidewalk リソースには、イベント通知の受信を許可する適切なポリシーが必要です。詳細については、「ワイヤレスイベント通知を受信するためのポリシー」を参照してください。

AWS Management Console を使用しての通知を有効にする

コンソールでイベントメッセージを有効にするには、AWS IoT コンソールの [Settings] (設定) タブに移動し、[LoRaWAN and Sidewalk event notification] (LoRaWAN および Sidewalk イベント通知) セクションに移動します。

特定のリソースタイプに属する AWS アカウント 内のすべてのリソースの通知を有効にして、それらをモニタリングできます。

すべてのリソースに対して通知を有効にするには

1. [LoRaWAN and Sidewalk event notification] (LoRaWAN および Sidewalk イベント通知) セクションの [All resources] (すべてのリソース) タブで、[Action] (アクション) を選択してから [Manage events] (イベントの管理) を選択します。
2. モニタリングするイベントを有効化してから、[Update events] (イベントの更新) を選択します。特定のイベントをモニタリングする必要がなくなった場合は、[Action] (アクション)、[Manage events] (イベントの管理) を選択して、これらのイベントを無効にします。

特定のリソースタイプに属する AWS アカウント 内の個々のリソースの通知を有効にして、それらをモニタリングすることもできます。

個々のリソースに対して通知を有効にするには

1. [LoRaWAN and Sidewalk event notification] (LoRaWAN および Sidewalk イベント通知) セクションで、[Action] (アクション) を選択してから、[Add resources] (リソースを追加する) を選択します。
2. 通知を受け取るリソースおよびイベントを選択します:

    a. [LoRaWAN resources] (LoRaWAN リソース) または [Sidewalk resources] (Sidewalk リソース) のどちらのイベントをモニタリングするかを選択します。

b. リソースタイプに応じて、リソースに対して有効化するイベントを選択できます。その後、これらのイベントをサブスクライブして通知を受け取ることができます。選択内容:

   - [LoRaWAN resources] (LoRaWAN リソース): LoRaWAN デバイスの [join] (参加) イベントまたは LoRaWAN ゲートウェイの [connection status] (接続ステータス) イベントを有効にできます。
   - [Sidewalk resources] (Sidewalk リソース): Sidewalk パートナーアカウントと Sidewalk デバイスの [device registration state] (デバイス登録状態) または [proximity] (近接) イベント、あるいはその両方を有効にできます。

3. 選択したリソースタイプとイベントに応じて、モニタリングするワイヤレスデバイスまたはゲートウェイを選択します。すべてのリソースを組み合わせて、最大 250 のリソースを選択できます。
4. [Submit] (送信) を選択してリソースを追加します。

追加したリソースは、コンソールの [LoRaWAN and Sidewalk event notification] (LoRaWAN および Sidewalk イベント通知) セクションのリソースタイプのタブに MQTT トピックとともに表示されます。

- [LoRaWAN join] (LoRaWAN 参加) イベントと Sidewalk デバイスのイベントは、コンソールの [Wireless devices] (ワイヤレスデバイス) セクションに表示されます。
- LoRaWAN ゲートウェイの [Connection status] (接続ステータス) イベントは、[Wireless gateways] (ワイヤレスゲートウェイ) セクションに表示されます。
- Sidewalk アカウントの [Device registration state] (デバイス登録状態) および [proximity] (近接) イベントは、[Sidewalk accounts] (Sidewalk アカウント) タブに表示されます。

MQTT クライアントを使用してトピックをサブスクライブする

すべてのリソースに対してイベントを有効にしたか、個々のリソースタイプに対してイベントを有効にしたかに応じて、有効にしたイベントはコンソールの [All resources] (すべてのリソース) タブまたは指定したリソースタイプのタブに MQTT トピックとともに表示されます。

- MQTT トピックの 1 つを選択した場合は、MQTT クライアントにアクセスして、これらのトピックをサブスクライブしてメッセージを受信できます。
- 複数のイベントを追加した場合は、複数のイベントトピックをサブスクライブして、それらの通知を受け取ることができます。複数のトピックをサブスクライブするには、トピックを選択し、[Action] (アクション) を選択してから、[Subscribe] (サブスクライブ) を選択します。

AWS CLI を使用しての通知を有効にする

AWS IoT Wireless API または AWS CLI を使用して、イベントを設定し、リソースを設定に追加できます。

すべてのリソースに対して通知を有効にする

UpdateEventConfigurationByResourceTypes API または update-event-configuration-by-resource-types CLI コマンドを使用して、特定のリソースタイプに属する AWS アカウント 内のすべてのリソースの通知を有効にし、それらをモニタリングできます。例:

```
aws iotwireless update-event-configuration-by-resource-types \
   --cli-input-json input.json
```

input.json の内容

```
{
   "DeviceRegistrationState": {
      "Sidewalk": {
         "AmazonIdEventTopic": "Enabled"
      }
   },
   "ConnectionStatus": {
      "LoRaWAN": {
         "WirelessGatewayEventTopic": "Enabled"
      }
   }
}
```

> **Note**
>
> 二重引用符 (") はバックスラッシュ (\) でエスケープされます。

現在のイベント設定は、GetEventConfigurationByResourceTypes API を呼び出すか、get-event-configuration-by-resource-types CLI コマンドを使用して取得できます。例:

```
aws iotwireless get-event-configuration-by-resource-types
```

個々のリソースに対して通知を有効にする

イベント設定に個々のリソースを追加し、発行されるイベントを API または CLI を使用して制御するには、UpdateResourceEventConfiguration API を呼び出すか、update-resource-event-configuration CLI コマンドを使用します。例:

```
aws iotwireless update-resource-event-configuration \
   --identifer 1ffd32c8-8130-4194-96df-622f072a315f \
   --identifier-type WirelessDeviceId \
   --cli-input-json input.json
```

input.json の内容

```
{
   "Join": {
      "LoRaWAN": {
         "DevEuiEventTopic": "Disabled"
      },
      "WirelessDeviceIdEventTopic": "Enabled"
   }
}
```

> **Note**
>
> 二重引用符 (") はバックスラッシュ (\) でエスケープされます。

現在のイベント設定は、GetResourceEventConfiguration API を呼び出すか、get-resource-event-configuration CLI コマンドを使用して取得できます。例:

```
aws iotwireless get-resource-event-configuration \
    --identifier-type WirelessDeviceId \
    --identifier 1ffd32c8-8130-4194-96df-622f072a315f
```

イベント設定を一覧表示する

また、AWS IoT Wireless API または AWS CLI を使用して、少なくとも 1 つのイベントトピックが有効になっているイベント設定を一覧表示することもできます。設定を一覧表示するには、ListEventConfigurations API オペレーション、または list-event-configurations CLI コマンドを使用します。例:

```
aws iotwireless list-event-configurations --resource-type WirelessDevice
```

LoRaWAN リソースのイベント通知

AWS Management Console または AWS IoT Wireless API オペレーションを使用して、LoRaWAN デバイスおよびゲートウェイのイベントについての通知を受け取ることができます。イベント通知とそれらを有効にする方法については、「AWS IoT Wireless イベント通知」および「ワイヤレスリソースのイベントを有効にする」を参照してください。

LoRaWAN リソースのイベントタイプ

LoRaWAN リソースに対して有効化できるイベントは次のとおりです。

- LoRaWAN デバイスの参加イベントを通知する参加イベント。デバイスが AWS IoT Core for LoRaWAN に参加したとき、またはタイプ 0 かタイプ 2 の再参加リクエストを受信したときに通知を受け取ります。
- LoRaWAN ゲートウェイの接続ステータスが接続済みまたは切断済みに変更されたときに通知する接続ステータスイベント。

以下のセクションでは、LoRaWAN リソースのイベントの詳細について説明します。

トピック

LoRaWAN 参加イベント

AWS IoT Core for LoRaWAN では、AWS IoT にオンボードした LoRaWAN デバイスの参加イベントを通知するメッセージを発行できます。参加イベントは、タイプ 0 またはタイプ 2 の参加または再参加リクエストが受信され、デバイスが AWS IoT Core for LoRaWAN に参加したときに通知します。

参加イベントの仕組み

LoRaWAN デバイスを AWS IoT Core for LoRaWAN にオンボードする場合、AWS IoT Core for LoRaWAN は AWS IoT Core for LoRaWAN を使用してデバイスの参加手順を実行します。その後、デバイスが使用可能になり、アップリンクメッセージを送信して、それが利用可能であることを示すことができます。デバイスが参加した後、アップリンクメッセージおよびダウンリンクメッセージを

デバイスと AWS IoT Core for LoRaWAN の間で交換できます。デバイスのオンボーディングの詳細については、「デバイスを AWS IoT Core for LoRaWAN にオンボードする」を参照してください。

デバイスが AWS IoT Core for LoRaWAN に参加したときに通知するイベントを有効にできます。また、参加イベントが失敗した場合、タイプ 0 またはタイプ 2 の再参加リクエストを受信した場合、およびそれが受け入れられた場合にも通知されます。

LoRaWAN 参加イベントを有効にする

LoRaWAN 参加の予約済みトピックのサブスクライバーがメッセージを受信できるようにするには、AWS Management Console から、あるいは API または CLI を使用して、イベント通知を有効にする必要があります。これらのイベントは、AWS アカウント 内のすべての LoRaWAN リソース、または選択したリソースに対して有効にできます。これらのイベントを有効にする方法の詳細については、「ワイヤレスリソースのイベントを有効にする」を参照してください。

LoRaWAN イベントの MQTT トピックの形式

LoRaWAN デバイス用に予約された MQTT トピックでは、次の形式を使用します。これらのトピックをサブスクライブしている場合は、AWS アカウント に登録されているすべての LoRaWAN デバイスで通知を受信できます。

- リソースレベルのトピック

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices
  ```

- 識別子のトピック

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices/
  {resourceID}/{id}
  ```

コードの説明は以下のとおりです。

{eventName}

{eventName} は `join` である必要があります。

{eventType}

{eventType} は次のいずれかになります。

- `join_req_received`

- rejoin_req_0_received
- rejoin_req_2_received
- join_accepted

{resourceID}

{resourceID} は dev_eui または wireless_device_id のいずれかになります。

例えば、以下のトピックをサブスクライブして、AWS IoT Core for LoRaWAN がデバイスからの参加リクエストを受け入れたときにイベント通知を受信できます。

```
$aws/iotwireless/events/join/join_accepted/lorawan/wireless_devices/
wireless_device_id/{id}
```

また、ワイルドカード文字 + を使用して、複数のトピックを同時にサブスクライブできます。ワイルドカード文字 + は、次のトピックなど、その文字を含むレベル内の任意の文字列に一致します。

```
$aws/iotwireless/events/join/join_req_received/lorawan/wireless_devices/
wireless_device_id/+
```

> **Note**
>
> 予約済みトピックをサブスクライブするには、ワイルドカード文字 # を使用できません。

トピックをサブスクライブするときに + ワイルドカードを使用する方法の詳細については、AWS IoT デベロッパーガイドの「MQTT トピックフィルター」を参照してください。

LoRaWAN 参加イベントのメッセージペイロード

以下は、LoRaWAN 参加イベントのメッセージペイロードを示しています。

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

ペイロードには以下の属性が含まれます。

eventId

AWS IoT Core for LoRaWAN によって生成される一意のイベント ID (文字列)。

eventType

発生したイベントのタイプ。次のいずれかの値を指定できます。

- `join_req_received`: このフィールドには、EUI パラメータ `JoinEui` または `AppEui` が表示されます。
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`: このフィールドには、`NetId` または `DevAddr` が表示されます。

wirelessDeviceId

LoRaWAN デバイスの ID。

timestamp

イベントが発生したときの Unix タイムスタンプ。

DevEui

デバイスラベルまたはデバイスのドキュメントにあるデバイスの一意の識別子。

DevAddr および EUIs (オプション)

これらのフィールドは、オプションのデバイスアドレスと EUI パラメータ `JoinEUI` または `AppEUI` です。

接続ステータスイベント

AWS IoT Core for LoRaWAN では、AWS IoT にオンボードした LoRaWAN ゲートウェイの接続ステータスイベントを通知するメッセージを発行できます。LoRaWAN ゲートウェイの接続ステータスが接続済みまたは切断済みに変更されたときに通知する接続ステータスイベント。

接続ステータスイベントの仕組み

ゲートウェイを AWS IoT Core for LoRaWAN にオンボーディングした後、ゲートウェイを AWS IoT Core for LoRaWAN に接続して、その接続ステータスを確認できます。このイベントは、ゲートウェイ接続ステータスが接続済みまたは切断済みに変更されたときに通知します。ゲートウェイのオンボーディングおよび AWS IoT Core for LoRaWAN への接続の詳細については、「ゲートウェイを AWS IoT Core for LoRaWAN にオンボードする」および「LoRaWAN ゲートウェイを接続し、接続ステータスを確認する」を参照してください。

LoRaWAN ゲートウェイの MQTT トピックの形式

LoRaWAN ゲートウェイ用に予約された MQTT トピックでは、次の形式を使用します。これらのトピックをサブスクライブしている場合は、AWS アカウント に登録されているすべての LoRaWAN ゲートウェイが通知を受信できます。

- リソースレベルのトピックの場合:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways
  ```

- 識別子のトピックの場合:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/
  wireless_gateways/{resourceID}/{id}
  ```

コードの説明は以下のとおりです。

{eventName}

{eventName} は `connection_status` である必要があります。

{eventType}

{eventType} は、`connected` または `disconnected` のいずれかになります。

{resourceID}

{resourceID} は `gateway_eui` または `wireless_gateway_id` のいずれかになります。

たとえば、次のトピックをサブスクライブして、すべてのゲートウェイが AWS IoT Core for LoRaWAN に接続されたときにイベント通知を受け取ることができます。

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/{id}
```

また、ワイルドカード文字 + を使用して、複数のトピックを同時にサブスクライブできます。ワイルドカード文字 + は、次のトピックなど、その文字を含むレベル内の任意の文字列に一致します。

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/+
```

> **Note**
>
> 予約済みトピックをサブスクライブするには、ワイルドカード文字 # を使用できません。

トピックをサブスクライブするときに + ワイルドカードを使用する方法の詳細については、AWS IoT デベロッパーガイドの「MQTT トピックフィルター」を参照してください。

接続ステータスイベントのメッセージペイロード

以下は、接続ステータスイベントのメッセージペイロードを示しています。

```
{
 // General fields
    "eventId": "string",
    "eventType": "connected|disconnected",
    "WirelessGatewayId": "string",
    "timestamp": "timestamp",

 // Event-specific fields
    "LoRaWAN": {
        "GatewayEui": "string"
    }
}
```

ペイロードには以下の属性が含まれます。

eventId

AWS IoT Core for LoRaWAN によって生成される一意のイベント ID (文字列)。

eventType

発生したイベントのタイプ。connected または disconnected のいずれかを設定できます。

wirelessGatewayId

LoRaWAN ゲートウェイの ID。

timestamp

イベントが発生したときの Unix タイムスタンプ。

GatewayEui

ゲートウェイラベルまたはゲートウェイのドキュメントにあるゲートウェイの一意の識別子。

Sidewalk リソースのイベント通知

AWS Management Console または AWS IoT Wireless API オペレーションを使用して、Sidewalk デバイスおよびパートナーアカウントのイベントについて通知を受け取ることができます。イベント通知とそれらを有効にする方法については、「AWS IoT Wireless イベント通知」および「ワイヤレスリソースのイベントを有効にする」を参照してください。

Sidewalk リソースのイベントタイプ

Sidewalk リソースに対して有効化できるイベントは次のとおりです。

- デバイスが登録され、使用できるようになったときなど、Sidewalk デバイスの状態の変化を通知するデバイスイベント。
- ビーコンが検出された、またはビーコンが消失したという通知を AWS IoT Wireless が Amazon Sidewalk から受信したときに通知される近接イベント。

以下のセクションでは、Sidewalk リソースのイベントの詳細について説明します。

トピック

- デバイス登録状態イベント
- 近接イベント

デバイス登録状態イベント

デバイス登録状態イベントは、Sidewalk デバイスがプロビジョニングまたは登録されたときなど、デバイスの登録状態に変化があった場合に、イベント通知を発行します。イベントは、デバイスがプロビジョニングされてから登録された時点まで、デバイスが通過するさまざまな状態に関する情報を提供します。

デバイス登録状態イベントの仕組み

Amazon Sidewalk と AWS IoT Wireless で Sidewalk デバイスをオンボードすると、AWS IoT Wireless は `create` オペレーションを実行して、Sidewalk デバイスを AWS アカウント に追加します。その後、デバイスがプロビジョニング済み状態になり、`eventType` は `provisioned` になります。デバイスのオンボーディングの詳細については、「AWS IoT Core for Amazon Sidewalk の開始方法」を参照してください。

デバイスが `provisioned` になった後、Amazon Sidewalk は `register` オペレーションを実行して Sidewalk デバイスを AWS IoT Wireless に登録します。登録プロセスが開始され、暗号化キーとセッションキーが AWS IoT で設定されます。デバイスが登録されると、`eventType` は `registered` になり、デバイスを使用できるようになります。

デバイスが `registered` になった後、Sidewalk は `deregister` デバイスにリクエストを送ることができます。次に、AWS IoT Wireless はリクエストを実行し、デバイスの状態を `provisioned` に戻します。デバイス状態の詳細については、「DeviceState」を参照してください。

デバイス登録状態イベントの通知を有効にする

デバイス登録状態の予約済みトピックのサブスクライバーがメッセージを受信できるようにするには、AWS Management Console から、あるいは API または CLI を使用して、イベント通知を有効にする必要があります。これらのイベントは、AWS アカウント 内のすべての Sidewalk リソース、または選択したリソースに対して有効にできます。これらのイベントを有効にする方法の詳細については、「ワイヤレスリソースのイベントを有効にする」を参照してください。

デバイス登録状態イベントの MQTT トピックの形式

デバイス登録状態イベントの通知を受けるには、ドル ($) 記号で始まる MQTT 予約トピックをサブスクライブします。詳細については、AWS IoT デベロッパーガイドの「MQTT トピック」を参照してください。

Sidewalk デバイス登録状態イベント用に予約された MQTT トピックは、次の形式を使用します。

- リソースレベルのトピックの場合:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 識別子のトピックの場合:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

コードの説明は以下のとおりです。

{eventName}

{eventName} は `device_registation_state` である必要があります。

{eventType}

{eventType} は、`provisioned` または `registered` のいずれかになります。

{resourceType}

{resourceType} は、`sidewalk_accounts` または `wireless_devices` のいずれかになります。

{resourceID}

{resourceID} は `sidewalk_accounts` の {resourceType} の `amazon_id` で、`wireless_devices` の {resourceType} の `wireless_device_id` です。

また、ワイルドカード文字 + を使用して、複数のトピックを同時にサブスクライブできます。ワイルドカード文字 + は、その文字を含むレベル内の任意の文字列に一致します。例えば、可能なすべてのイベントタイプ (`provisioned` および `registered`) について、特定の Amazon ID に登録されているすべてのデバイス関する通知を受ける場合、次のトピックフィルターを使用できます。

```
$aws/iotwireless/events/device_registration_state/+/sidewalk/
sidewalk_accounts/amazon_id/+
```

> **Note**
>
> 予約済みトピックをサブスクライブするには、ワイルドカード文字 # を使用できません。トピックの詳細については、AWS IoT デベロッパーガイドの「MQTT トピックフィルター」を参照してください。

デバイス登録状態イベントのメッセージペイロード

デバイス登録状態イベントの通知を有効にすると、イベント通知は MQTT を介して JSON ペイロードを使用して発行されます。これらのイベントには、次のペイロード例が含まれています。

```
{
    "eventId": "string",
    "eventType": "provisioned|registered",
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",

    // Event-specific fields
    "operation": "create|deregister|register",
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

ペイロードには以下の属性が含まれます。

eventId

一意のイベント ID (文字列)。

eventType

発生したイベントのタイプ。`provisioned` または `registered` のいずれかを設定できます。

wirelessDeviceId

ワイヤレスデバイスの識別子。

timestamp

イベントが発生したときの Unix タイムスタンプ。

オペレーション

イベントをトリガーしたオペレーション。有効な値は、`create`、`register`、`deregister` です。

sidewalk

イベント通知を受け取る Sidewalk Amazon ID または `SidewalkManufacturingSn`。

近接イベント

近接イベントは、AWS IoT が Sidewalk デバイスからビーコンを受信したときにイベント通知を発行します。Sidewalk デバイスが Amazon Sidewalk に近づくと、デバイスから送信されるビーコンは Amazon Sidewalk によって定期的にフィルターされ、AWS IoT Wireless に送信されます。次に、ビーコンが受信されると、AWS IoT Wireless はこれらのイベントを通知します。

近接イベントの仕組み

近接イベントは AWS IoT がビーコンを受信すると通知されます。Sidewalk デバイスはいつでもビーコンを出力する可能性があります。デバイスが Amazon Sidewalk の近くにある場合、Sidewalk はビーコンを受信し、ビーコンを AWS IoT Wireless に一定の時間間隔で転送します。Amazon Sidewalk はこの時間間隔を 10 分に設定しています。AWS IoT Wireless が Sidewalk からビーコンを受信すると、イベントの通知が届きます。

近接イベントは、ビーコンが検出されたとき、またはビーコンが消失したときに通知されます。近接イベントが通知される間隔を設定できます。

近接イベントの通知を有効にする

Sidewalk 近接の予約済みトピックのサブスクライバーがメッセージを受信できるようにするには、AWS Management Console から、あるいは API または CLI を使用して、イベント通知を有効にする必要があります。これらのイベントは、AWS アカウント 内のすべての Sidewalk リソース、または選択したリソースに対して有効にできます。これらのイベントを有効にする方法の詳細については、「ワイヤレスリソースのイベントを有効にする」を参照してください。

近接イベントの MQTT トピックの形式

近接イベントの通知を受けるには、ドル ($) 記号で始まる MQTT 予約トピックをサブスクライブします。詳細については、AWS IoT デベロッパーガイドの「MQTT トピック」を参照してください。

Sidewalk の近接イベント用に予約された MQTT トピックは、次の形式を使用します。

- リソースレベルのトピックの場合:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 識別子のトピックの場合:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

コードの説明は以下のとおりです。

{eventName}

{eventName} は `proximity` である必要があります。

{eventType}

{eventType} は、`beacon_discovered` または `beacon_lost` のいずれかになります。

{resourceType}

{resourceType} は、`sidewalk_accounts` または `wireless_devices` のいずれかになります。

{resourceID}

{resourceID} は `sidewalk_accounts` の {resourceType} の `amazon_id` で、`wireless_devices` の {resourceType} の `wireless_device_id` です。

また、ワイルドカード文字 + を使用して、複数のトピックを同時にサブスクライブできます。ワイルドカード文字 + は、その文字を含むレベル内の任意の文字列に一致します。例えば、可能なすべてのイベントタイプ (`beacon_discovered` および `beacon_lost`) について、特定の Amazon ID に登録されているすべてのデバイス関する通知を受ける場合、次のトピックフィルターを使用できます。

```
$aws/iotwireless/events/proximity/+/sidewalk/sidewalk_accounts/amazon_id/+
```

> **Note**
>
> 予約済みトピックをサブスクライブするには、ワイルドカード文字 # を使用できません。トピックの詳細については、AWS IoT デベロッパーガイドの「MQTT トピックフィルター」を参照してください。

近接イベントのメッセージペイロード

近接イベントの通知を有効にすると、イベントメッセージは MQTT を介して JSON ペイロードを使用して発行されます。これらのイベントには、次のペイロード例が含まれています。

```
{
    "eventId": "string",
```

```
    "eventType": "beacon_discovered|beacon_lost",
    "WirelessDeviceId": "string",
    "timestamp": "1234567890123",

    // Event-specific fields
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

ペイロードには以下の属性が含まれます。

eventId

一意のイベント ID。文字列です。

eventType

発生したイベントのタイプ。`beacon_discovered` または `beacon_lost` のいずれかを設定できます。

WirelessDeviceId

ワイヤレスデバイスの識別子。

timestamp

イベントが発生したときの Unix タイムスタンプ。

sidewalk

イベント通知を受け取る Sidewalk Amazon ID または `SidewalkManufacturingSn`。

AWS IoT Wireless API オペレーション

LoRaWAN または Sidewalk エンドデバイスをオンボードするとき、または Sidewalk エンドデバイスを一括でプロビジョニングするためのインポートタスクを作成するときに、次の追加の API オペレーションを実行できます。

以下のセクションには、これらの API オペレーションに関する追加情報が含まれています。

トピック

- デバイスプロファイルの AWS IoT Wireless API オペレーション
- LoRaWAN および Sidewalk エンドデバイスの AWS IoT Wireless API オペレーション
- ワイヤレスデバイスの送信先の AWS IoT Wireless API オペレーション
- 一括プロビジョニングの AWS IoT Core for Amazon Sidewalk API オペレーション

デバイスプロファイルの AWS IoT Wireless API オペレーション

LoRaWAN および Sidewalk デバイスプロファイルでは、次の API オペレーションを実行できます。

- `CreateDeviceProfile` API または `create-device-profile` CLI
- `GetDeviceProfile` API または `get-device-profile` CLI
- `ListDeviceProfiles` API または `list-device-profiles` CLI
- `DeleteDeviceProfile` API または `delete-device-profile` CLI

次のセクションでは、プロファイルの一覧表示と削除の方法を説明します。デバイスプロファイルの作成と取得の詳細については、次を参照してください。

- デバイスプロファイルを追加する
- ステップ 1: デバイスプロファイルを作成する

AWS アカウント 内のデバイスプロファイルを一覧表示する

`ListDeviceProfiles` API オペレーションを使用して、AWS IoT Wireless に追加した AWS アカウント 内のデバイスプロファイルを一覧表示できます。この情報を使用して、このプロファイルを関連付けるデバイスを特定できます。

LoRaWAN または Sidewalk デバイスプロファイルのみを表示するようにリストをフィルタリングするには、API の実行時に Type に設定します。CLI コマンドの例を以下に示します。

```
aws iotwireless list-device-profiles --wireless-device-type "Sidewalk"
```

このコマンドを実行すると、プロファイル識別子と Amazon リソースネーム (ARN) を含む、追加したデバイスプロファイルのリストが返されます。特定のプロファイルに関する追加情報を取得するには、GetDeviceProfile API を使用します。

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

デバイスプロファイルを AWS アカウント から削除する

DeleteDeviceProfile API オペレーションを使用してデバイスプロファイルを削除できます。CLI コマンドの例を以下に示します。

> **Warning**
>
> 削除の操作は元に戻せません。デバイスプロファイルは AWS アカウント から完全に削除されます。

```
aws iotwireless delete-device-profile --name "SidewalkProfile"
```

このコマンドでは、出力が生成されません。GetDeviceProfile API または ListDeviceProfiles API オペレーションを使用して、プロファイルがアカウントから削除されたことを確認できます。

LoRaWAN および Sidewalk エンドデバイスの AWS IoT Wireless API オペレーション

LoRaWAN および Sidewalk デバイスでは、次の API オペレーションを実行できます。

- CreateWirelessDevice API または create-wireless-device CLI
- GetWirelessDevice API または get-wireless-device CLI
- ListWirelessDevices API または list-wireless-devices CLI
- DeleteWirelessDevice API または delete-wireless-device CLI
- UpdateWirelessDevice API または update-wireless-device CLI
- AssociateWirelessDeviceWithThing API または associate-wireless-device-with-thing CLI
- DisassociateWirelessDeviceFromThing API または disassociate-wireless-device-from-thing CLI

次のセクションでは、デバイスの一覧表示と削除の方法を説明します。ワイヤレスデバイスの作成とデバイス情報の取得については、次を参照してください。

- ワイヤレスデバイスを AWS IoT Core for LoRaWAN に追加する
- ステップ 2: Sidewalk デバイスを追加する

AWS アカウント 内のワイヤレスデバイスを IoT モノに関連付ける

LoRaWAN および Sidewalk デバイスを AWS IoT モノに関連付けるには、AssociateWirelessDeviceWithThing API オペレーションを使用します。

AWS IoT におけるモノを使用すると、デバイスの検索と管理を簡単に行えます。モノをデバイスに関連付けると、デバイスは他の AWS IoT Core 機能にアクセスできます。この API の使用の詳細については、「AssociateWirelessDeviceWithThing」を参照してください。

このコマンドの実行例を次に示します。このコマンドを実行しても、出力は生成されません。

```
aws iotwireless associate-wireless-device-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MySidewalkThing"
```

ワイヤレスデバイスと AWS IoT モノの関連付けを解除するには、次の例のように DisassociateWirelessDeviceFromThing API オペレーションを使用します。

```
aws iotwireless disassociate-wireless-device-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

AWS アカウント のワイヤレスデバイスを一覧表示する

AWS IoT Wireless に追加した AWS アカウント 内のワイヤレスデバイスを一覧表示するには、ListWirelessDevices API オペレーションを使用します。LoRaWAN または Sidewalk デバイスのみを返すようにリストをフィルタリングするには、WirelessDeviceType を設定します。

このコマンドの実行例を次に示します。

```
aws iotwireless list-wireless-devices --wireless-device-type Sidewalk
```

このコマンドを実行すると、プロファイル識別子と Amazon リソースネーム (ARN) を含む、追加したデバイスのリストが返されます。特定のデバイスに関する追加情報を取得するには、GetWirelessDevice API オペレーションを使用します。

```
{
    "Wireless DeviceList": [
        {
            "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"
        }
    ]
}
```

AWS アカウント からワイヤレスデバイスを削除する

ワイヤレスデバイスを削除するには、削除するデバイスの WirelessDeviceID を DeleteWirelessDevice API オペレーションに渡します。

コマンドの例を以下に示します。

```
aws iotwireless delete-wireless-device --id "23456789-abcd-0123-bcde-fabc012345678"
```

このコマンドでは、出力が生成されません。GetWirelessDevice API または ListWirelessDevices API オペレーションを使用して、デバイスがアカウントから削除されたことを確認できます。

ワイヤレスデバイスの送信先の AWS IoT Wireless API オペレーション

LoRaWAN および Sidewalk エンドデバイスの送信先に対して次の API オペレーションを実行できます。

- CreateDestination API または create-destination CLI
- GetDestination API または get-destination CLI
- UpdateDestination API または update-destination CLI
- ListDestinations API または list-destinations CLI
- DeleteDestination API または delete-destination CLI

次のセクションでは、送信先の取得、一覧表示、更新、削除の方法について説明します。送信先の作成方法の詳細については、「Sidewalk エンドデバイスの送信先を追加する」を参照してください。

送信先に関する情報を取得する

GetDestination API オペレーションを使用して、AWS IoT Wireless のアカウントに追加した送信先に関する情報を取得できます。API への入力として送信先名を指定します。API は、指定された識別子に一致する送信先に関する情報を返します。

CLI コマンドの例を以下に示します。

```
aws iotwireless get-destination --name SidewalkDestination
```

このコマンドを実行すると、送信先のパラメータが返されます。

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "SidewalkDestination",
    "Expression": "IoTWirelessRule",
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

送信先のプロパティを更新する

UpdateDestination API オペレーションを使用して、AWS IoT Wireless のアカウントに追加した送信先のプロパティを更新します。以下の例は、説明プロパティを更新する CLI コマンドの例を示しています。

```
aws iotwireless update-destination --name SidewalkDestination \
    --description "Destination for messages processed using IoTWirelessRule"
```

AWS アカウント の送信先を一覧表示する

ListDestinations API オペレーションを使用して、AWS IoT Wireless に追加した AWS アカウント の送信先を一覧表示します。LoRaWAN および Sidewalk エンドデバイスの送信先のみを返すようにリストをフィルタリングするには、WirelessDeviceType パラメータを使用します。

CLI コマンドの例を以下に示します。

```
aws iotwireless list-destinations --wireless-device-type "Sidewalk"
```

このコマンドを実行すると、Amazon リソースネーム (ARN) を含む、追加した送信先のリストが返されます。特定の送信先に関する追加情報を取得するには、GetDestination API を使用します。

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

AWS アカウント から送信先を削除する

送信先を削除するには、DeleteDestination API オペレーションの入力として、削除する送信先の名前を渡します。CLI コマンドの例を以下に示します。

> **Warning**
>
> 削除の操作は元に戻せません。送信先は AWS アカウント から完全に削除されます。

```
aws iotwireless delete-destination --name "SidewalkDestination"
```

このコマンドでは、出力が生成されません。GetDestination API または ListDestinations API オペレーションを使用して、送信先がアカウントから削除されたことを確認できます。

一括プロビジョニングの AWS IoT Core for Amazon Sidewalk API オペレーション

Sidewalk エンドデバイスを一括プロビジョニングするには、次の API オペレーションを実行します。

- StartWirelessDeviceImportTask API または start-wireless-device-import-task CLI
- StartSingleWirelessDeviceImportTask API または start-single-wireless-device-import-task CLI

- ListWirelessDeviceImportTasks API または list-wireless-device-import-tasks CLI
- ListDevicesForWirelessDeviceImportTask API または list-devices-for-wireless-device-import-task CLI
- GetWirelessDeviceImportTask API または get-wireless-device-import-task CLI
- UpdateWirelessDeviceImportTask API または update-wireless-device-import-task CLI
- DeleteWirelessDeviceImportTask API または delete-wireless-device-import-task CLI

次のセクションでは、インポートタスクの取得、一覧表示、更新、削除の方法について説明します。インポートタスクの作成に関する詳細については、「一括プロビジョニングの AWS IoT Core for Amazon Sidewalk API オペレーション」を参照してください。

インポートタスクの情報を取得する

ListDevicesForWirelessDeviceImportTask API オペレーションを使用して、特定のインポートタスクとそのタスク内のデバイスのオンボーディングステータスに関する情報を取得できます。API オペレーションの入力として、StartWirelessDeviceImportTask または StartSingleWirelessDeviceImportTask API オペレーションから取得したインポートタスク ID を指定します。その後、API は指定された識別子に一致するインポートタスクに関する情報を返します。

CLI コマンドの例を以下に示します。

```
aws iotwireless list-devices-for-wireless-device-import-task --id e2a5995e-743b-41f2-a1e4-3ca6a5c5249f
```

このコマンドを実行すると、インポートタスクの情報とデバイスのオンボーディングステータスが返されます。

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

インポートタスクデバイスの概要を取得する

特定のインポートタスクに追加したデバイスのオンボーディングステータスの概要情報の数を取得するには、GetWirelessDeviceImportTask API オペレーションを使用します。CLI コマンドの例を以下に示します。

```
aws iotwireless get-wireless-device-import-task --Id "e2a5995e-743b-41f2-
a1e4-3ca6a5c5249f"
```

以下のコードは、コマンドからのレスポンスの例を示しています。

```
{
   "NumberOfFailedImportedDevices": 2,
   "NumberOfOnboardedImportedDevices": 4,
   "NumberOfPendingImportedDevices": 1
}
```

デバイスをインポートタスクに追加する

UpdateWirelessDeviceImportTask API オペレーションを使用して、追加した既存のインポートタスクにデバイスを追加します。この API オペレーションを使用して、StartWirelessDeviceImportTask API オペレーションを使用して作成したタスクにこれまで含まれていなかったデバイスのシリアル番号 (SMSN) を追加できます。

デバイスをインポートタスクに追加するには、API リクエストの一部として、追加するデバイスのシリアル番号を含む Amazon S3 バケット内の新しい CSV ファイルを指定します。リク

エストは、現在インポートタスクにあるデバイスのオンボーディングプロセスがまだ開始されていない場合にのみ受け付けられます。オンボーディングプロセスが既に開始されている場合、UpdateWirelessDeviceImportTask API リクエストは失敗します。

それでもインポートタスクにデバイスを追加したい場合は、UpdateWirelessDeviceImportTask API オペレーションをもう一度実行できます。この API オペレーションを実行する前に、最初の UpdateWirelessDeviceImportTask API リクエストで S3 バケット内の CSV ファイルの処理が完了している必要があります。

> **Note**
>
> ListImportedWirelessDeviceTasks API リクエストを実行しても、UpdateWirelessDeviceImportTask API オペレーションを使用して指定された新しい CSV ファイルの S3 URL は現在返されません。代わりに、API オペレーションは、StartWirelessDeviceImportTask API リクエストを使用して最初に送信されたリクエストの S3 URL を返します。

CLI コマンドの例を以下に示します。

```
aws iotwireless update-wireless-device-import task \
    --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f" \
    --sidewalk '{"FileForCreateDevices": "s3://import_task_bucket/import_file3"}'
```

AWS アカウント 内のインポートタスクを一覧表示する

ListWirelessDeviceImportTasks API または list-imported-wireless-device-tasks CLI コマンドを使用して、AWS アカウント 内のインポートタスクを一覧表示します。CLI コマンドの例を以下に示します。

```
aws iotwireless list-wireless-device-import-tasks
```

このコマンドを実行すると、作成したインポートタスクのリストが返されます。リストには、Amazon S3 CSV ファイルと、指定された IAM ロール、インポートタスク ID、およびデバイスオンボーディングステータスの概要情報が含まれます。

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

インポートタスクを AWS アカウント から削除する

インポートタスクを削除するには、インポートタスク ID を DeleteWirelessDeviceImportTask API オペレーションまたは delete-wireless-device-import-task CLI コマンドに渡します。

> **Warning**
>
> 削除の操作は元に戻せません。インポートタスクは AWS アカウント から完全に削除されます。

DeleteWirelessDeviceImportTask API リクエストを実行すると、バックグラウンドプロセスによってインポートタスクの削除が開始されます。リクエストが進行中の場合、インポートタスク内のデバイスのシリアル番号 (SMSN) は削除処理中となります。削除が完了した後にのみ、ListImportedWirelessDeviceTasks または GetImportedWirelessDeviceTasks API オペレーションを使用してこの情報を表示できるようになります。

インポートタスクにまだオンボーディング待ちのデバイスが含まれている場合、DeleteWirelessDeviceImportTask API リクエストは、インポートタスクのすべてのデバイスがオンボーディングされたか、オンボーディングに失敗した後にのみ処理されます。インポー

トタスクは 90 日後に期限切れになります。タスクの有効期限が切れると、アカウントから削除できます。ただし、インポートタスクを使用して正常にオンボーディングされたデバイスは削除されません。

> **Note**
>
> DeleteWirelessDeviceImportTask API リクエストを使用して、削除が保留されているデバイスのシリアル番号を含む別のインポートタスクを作成しようとすると、StartWirelessDeviceImportTask API オペレーションはエラーを返します。

CLI コマンドの例を以下に示します。

```
aws iotwireless delete-import-task --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

このコマンドでは、出力が生成されません。タスクが削除された後、インポートタスクがアカウントから削除されたことを確認するには、GetWirelessDeviceImportTask API オペレーションまたは ListWirelessDeviceImportTasks API オペレーションを使用できます。

AWS CloudFormation を使用した AWS IoT Wireless リソースの作成

AWS IoT Wireless は AWS CloudFormation と統合されています。これは、 AWS リソースのモデル化とセットアップを支援するサービスです。これにより、リソースやインフラの作成と管理に費やす時間を短縮することができます。必要なすべての AWS リソースを記述するテンプレートを作成すると、これらのリソースが AWS CloudFormation で自動的にプロビジョニングおよび設定されます。

AWS CloudFormation を使用すると、テンプレートを再利用して AWS IoT Wireless リソースをいつでも繰り返しセットアップできます。リソースを一度記述するだけで、同じリソースを複数の AWS アカウント やリージョンで何度でもプロビジョニングすることができます。

AWS IoT Wireless および AWS CloudFormation テンプレート

AWS IoT Wireless および関連サービスのリソースをプロビジョニングして設定するには、AWS CloudFormation テンプレートについて理解しておく必要があります。テンプレートは、JSON や YAML でフォーマットされたテキストファイルです。これらのテンプレートには、AWS CloudFormation スタックにプロビジョニングしたいリソースを記述します。JSON や YAML に不慣れな方は、AWS CloudFormation デザイナー を使えば、AWS CloudFormation テンプレートを使いこなすことができます。詳細については、「AWS CloudFormation ユーザーガイド」の「AWS CloudFormation デザイナー とは」を参照してください。

AWS IoT Wireless は、AWS CloudFormation でのワイヤレスリソースの作成をサポートしています。AWS IoT Wireless リソースの JSON テンプレートと YAML テンプレートの例を含む詳細情報については、AWS CloudFormation ユーザーガイドの「AWS IoT Wireless リソースタイプのリファレンス」を参照してください。

AWS CloudFormation の詳細はこちら

AWS CloudFormation の詳細については、以下のリソースを参照してください。

- AWS CloudFormation
- AWS CloudFormation ユーザーガイド
- AWS CloudFormation コマンドラインインターフェイスユーザーガイド

AWS IoT Wireless のクォータ

AWS アカウント には、AWS のサービスごとにデフォルトのクォータ (以前は制限と呼ばれたもの) があります。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。

AWS IoT Wireless のクォータを表示するには、[Service Quotas コンソール] を開きます。ナビゲーションペインで、AWS のサービス を選択し、AWS IoT Wireless を選択します。

クォータの引き上げをリクエストするには、「Service Quotas ユーザーガイド」の「クォータ引き上げリクエスト」を参照してください。Service Quotas でクォータがまだ利用できない場合は、[上限引き上げ] フォームを使用してください。

AWS IoT Wireless には、次のクォータがあります。

- デバイス間で送信されるデバイスデータに適用される AWS IoT Core for LoRaWAN クォータ
- LoRaWAN デバイスと Sidewalk デバイスの両方に適用される AWS IoT Wireless API オペレーション。

詳細については、「AWS 全般のリファレンス」の「AWS IoT Core for LoRaWAN クォータ」を参照してください。

AWS IoT Wireless リソースのタグ付け

デバイス、ゲートウェイ、送信先およびプロファイルの管理と整理を支援するために、オプションで、タグの形式でこれらの各リソースに独自のメタデータを割り当てることができます。このセクションでは、タグとその作成方法について説明します。AWS IoT Wireless には請求グループがなく、AWS IoT Core と同じ請求グループを使用します。詳しくは、AWS IoT Core ドキュメントの「請求グループ」を参照してください。

タグの基本

同じタイプの 複数の AWS IoT Wireless リソースがある場合、タグを使って目的、所有者、環境など、さまざまな方法でリソースを分類することができます。これにより、リソースに割り当てたタグに基づいてリソースをすばやく特定できます。

タグはそれぞれ、1 つのキーとオプションの値で設定され、どちらもユーザーが定義します。例えば、デバイスのファームウェアが更新されている LoRaWAN デバイスのグループに一連のタグを定義できます。リソースをさらに管理しやすくするために、リソースの種類ごとのニーズを満たす、一貫した一連のタグキーを作成することをお勧めします。

追加または適用したタグに基づいて、リソースを検索およびフィルター処理できます。タグを使用して、IAM ポリシーと請求グループタグを使用してコストを分類および追跡することで、リソースへのアクセスを制御することもできます。

タグを作成して管理する

AWS Management Console、AWS IoT Wireless、または AWS CLI でタグエディタを使用してタグを作成および管理することができます。

コンソールを使用する場合

使いやすさのために、のタグエディタは、タグを作成および管理するための AWS Management Console の中央の統一された方法を提供します。詳細については、「 の操作 AWS Management Console」の「タグエディタの操作」を参照してください。

API または CLI の使用

API または CLI を使用して、ワイヤレスデバイス、ゲートウェイ、プロファイル、および送信先にタグを関連付けることもできます。その場合は、次のコマンドの Tags フィールドを使用します。

- AssociateAwsAccountWithPartnerAccount
- CreateDestination
- CreateDeviceProfile
- CreateFuotaTask
- CreateMulticastGroup
- CreateServiceProfile
- CreateWirelessGateway
- CreateWirelessGatewayTaskDefinition
- CreateWirelessDevice
- API_StartBulkAssociateWirelessDeviceWithMulticastGroup

リソースのタグを更新するか、またはタグを一覧表示する

以下のコマンドを使用して、タグ付けがサポートされている既存のリソースに対してタグを追加、変更、または削除できます。

- TagResource
- ListTagsForResource
- UntagResource

タグのキーと値は編集でき、タグはリソースからいつでも削除できます。タグの値を空の文字列に設定することはできますが、タグの値を null に設定することはできません。特定のリソースについて既存のタグと同じキーを持つタグを追加した場合、古い値は新しい値によってオーバーライドされます。リソースを削除すると、リソースに関連付けられているすべてのタグも削除されます。

タグの制約と制限

次のベーシックな制限がタグに適用されます。

- リソースあたりのタグの最大数は 50 です。
- キーの最大長: 127 Unicode 文字 (UTF-8)。
- 値の最大長: 255 Unicode 文字 (UTF-8)。

- タグのキーと値では、大文字と小文字が区別されます。
- タグの名前または値に「aws:」プレフィックスは使用しないでください。AWS の利用のために予約されています。このプレフィックスが含まれるタグの名前または値は編集または削除できません。このプレフィックスを持つタグは、リソースあたりのタグ数の制限にはカウントされません。
- 複数のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスでも許可される文字に制限が適用されることがあるため注意ください。使用できる文字は、UTF-8 で表現できる文字、スペース、および数字と、特殊文字 +、-、=、.、_、:、/、@ です。

IAM ポリシーでのタグの使用

ユーザーが作成、変更、または使用できるリソースを指定するには、AWS IoT Wireless API アクションに使用する IAM ポリシーでタグベースのリソースレベルのアクセス許可を適用できます。リソースのタグに基づいてユーザーアクセス (アクセス許可) を制御するには、IAM ポリシーの以下の条件コンテキストのキーと値とともに Condition 要素 (Condition ブロックとも呼ばれる) を使用して、リソースのタグに基づいてユーザーアクセス (アクセス許可) を制御できます。

- 特定のタグを持つリソースに対してユーザーアクションを許可または拒否するには、aws:ResourceTag/*tag-key*: *tag-value* を使用します。
- タグが許可されているリソースを作成または変更する API リクエストを作成する場合に、特定のタグが使用されている (または、使用されていない) ことを要求するには、aws:RequestTag/*tag-key*: *tag-value* を使用します。
- タグが許可されているリソースを作成または変更する API リクエストを作成する場合に、特定の一連のタグが使用されている (または、使用されていない) ことを要求するには、aws:TagKeys: [*tag-key*, ...] を使用します。

> **Note**
>
> IAM ポリシーの条件コンテキストのキーと値は、タグ付け可能なリソースの ID が必須パラメータである AWS IoT アクションにのみ適用されます。例えば、タグ付け可能なリソースはこのリクエスト内で参照されないため、DescribeEndpoint の使用は条件コンテキストキーおよび値に基づいて許可または拒否されません。

タグの使用の詳細については、[AWS Identity and Access ManagementUser Guide](ユーザーガイド)の[Controlling Access Using Tags](タグを使用したアクセスの制御)を参照してください。そのガイド

の「IAM JSON ポリシーリファレンス」セクションには、IAM での JSON ポリシーの要素、変数、および評価ロジックの詳細な構文、説明、および例が記載されています。

次のポリシー例では、タグベースの 2 つの制約が適用されています。このポリシーによって制限されている IAM ユーザーは、次のように制限されます。

- リソースにタグ "env=prod" を付与できません (この例の `"aws:RequestTag/env" : "prod"` の行を参照)。
- 既存のタグ "env=prod" を持つリソースに対しては変更またはアクセスできません (この例の `"aws:ResourceTag/env" : "prod"` の行を参照)

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

次のようにタグ値を 1 つのリストとして指定して、1 つのタグキーに対して複数のタグ値を指定することもできます。

```
"StringEquals" : {
              "aws:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> タグに基づいてリソースへのユーザーのアクセスを許可または拒否する場合は、ユーザーが同じリソースに対してそれらのタグを追加または削除することを明示的に拒否することを検討する必要があります。そうしないと、ユーザーはそのリソースのタグを変更することで、制限を回避してリソースにアクセスできてしまいます。

AWS IoT Wireless ユーザーガイドのドキュメント履歴

次の表は、AWS IoT Wireless のドキュメントリリースの内容をまとめたものです。

変更	説明	日付
初回リリース	AWS IoT Wireless ユーザーガイドの初回リリース	2020 年 12 月 31 日