aws

개발자 가이드

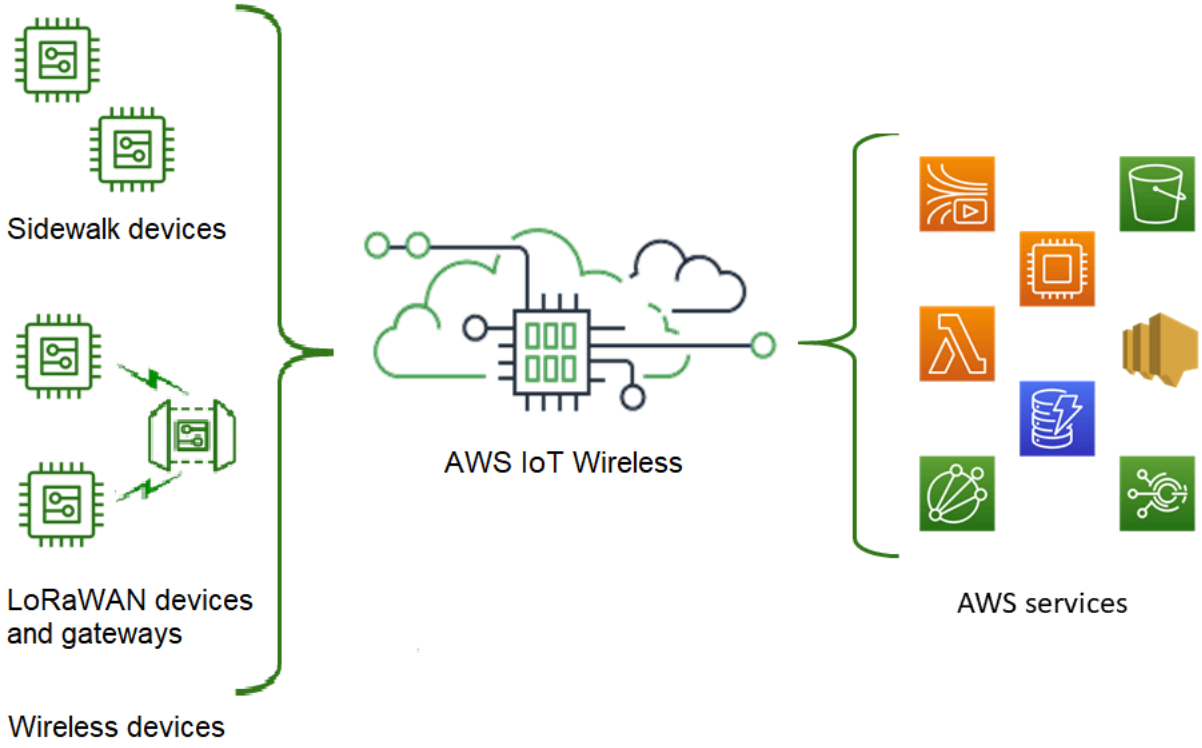
AWS IoT Wireless

AWS IoT Wireless: 개발자 가이드

Table of Contents

AWS IoT Wireless란?

AWS IoT Wireless는 사용자의 무선 디바이스를 다른 디바이스 및 AWS 클라우드 서비스에 연결하는 클라우드 서비스를 제공합니다. 디바이스를 AWS IoT 무선에 연결하여 디바이스를 AWS IoT 기반 솔루션에 통합할 수 있습니다. AWS IoT Wireless를 사용하여 LoRaWAN 및 Sidewalk 디바이스를 모두 AWS IoT에 온보딩할 수 있습니다. 이러한 무선 디바이스는 저전력 광역 네트워크(LPWAN) 통신 프로토콜을 사용하여 AWS IoT와 통신합니다.



AWS IoT 무선의 기능

AWS IoT 무선는 다음의 기능을 제공합니다.

LoRaWAN 및 Sidewalk 디바이스 온보딩

LoRaWAN 및 Sidewalk 디바이스를 모두 AWS IoT 무선에 온보딩할 수 있습니다.

- AWS IoT Core for LoRaWAN

  LoRaWAN 디바이스 및 게이트웨이를 AWS IoT Wireless에 온보딩하려면 AWS IoT Core for LoRaWAN을 사용하세요. 이는 프라이빗 LNS를 설정하고 운영하지 않아도 되는 LoRaWAN Network Server(LNS)입니다. AWS IoT Core for LoRaWAN은 구성 및 업데이트 서버(CUPS) 및 펌

웨어 무선 업데이트(FUOTA) 기능을 사용하여 게이트웨이 관리를 제공합니다. 자세한 내용은 AWS IoT Core for LoRaWAN(이)란 무엇인가요? 단원을 참조하십시오.

- Amazon Sidewalk용 AWS IoT Core

  Sidewalk 디바이스를 AWS IoT Wireless에 온보딩하려면 Amazon Sidewalk용 AWS IoT Core에서 제공하는 기능을 사용할 수 있습니다. Amazon Sidewalk는 Amazon Echo, Ring 보안 캠코더, 옥외 조명 등의 디바이스를 연결하는 공유 네트워크로, 커뮤니티 내 다른 Sidewalk 디바이스를 지원할 수 있습니다. 자세한 내용은 Amazon Sidewalk용 AWS IoT Core란? 단원을 참조하십시오.

AWS IoT Core와의 통합

AWS IoT Wireless 통합에서 제공하는 다음 기능을 AWS IoT Core와 함께 사용할 수 있습니다.

- 디바이스를 AWS IoT 사물과 연결

  무선 디바이스와 게이트웨이를 AWS IoT 사물과 연결하면 클라우드에 디바이스의 표현을 저장하는 데 도움이 됩니다. AWS IoT에서 사물을 사용하여 디바이스를 더 쉽게 검색 및 관리하고 다른 AWS IoT Core 기능에 액세스할 수 있습니다. 자세한 내용은 AWS IoT Core 개발자 설명서의 AWS IoT로 디바이스 관리를 참조하세요.

- AWS IoT 규칙을 사용하여 메시지 라우팅

  AWS IoT의 규칙 기능을 사용하여 다른 AWS 서비스 및 애플리케이션과 상호 작용할 수 있습니다. 디바이스에서 클라우드로 전송되는 업링크 메시지는 이러한 서비스 및 기타 애플리케이션으로 라우팅될 수 있습니다. 자세한 내용은 AWS IoT Core 개발자 안내서의 AWS IoT 규칙을 참조하세요.

AWS IoT Wireless를 처음 사용하는 경우

AWS IoT Wireless를 처음 사용할 경우 먼저 다음 단원을 읽을 것을 권장합니다.

- AWS IoT Core for LoRaWAN(이)란 무엇인가요?

  이 단원에서는 LoRaWAN 기술에 대한 개요와 AWS IoT Core for LoRaWAN의 작동 방식을 설명합니다. 또한 자세히 알아보는 데 도움이 되는 리소스도 제공합니다.

- Amazon Sidewalk용 AWS IoT Core란?

  이 단원에서는 Amazon Sidewalk 기술에 대한 개요와 Amazon Sidewalk용 AWS IoT Core의 작동 방식에 대해 설명합니다. 또한 자세히 알아보는 데 도움이 되는 리소스도 제공합니다.

- Amazon Sidewalk용 AWS IoT Core 시작하기

  이 단원을 읽고 Amazon Sidewalk용 AWS IoT Core를 사용하는 방법과 Amazon Sidewalk 디바이스를 온보딩하는 방법에 대해 알아보세요.

- AWS IoT Core for LoRaWAN에 게이트웨이 및 디바이스 연결

  다음으로 콘솔과 API를 사용하여 LoRaWAN 디바이스를 온보딩하는 방법에 대해 자세히 알아볼 수 있습니다.

관련 서비스

- Amazon CloudWatch

  LoRaWAN 또는 Sidewalk 디바이스를 AWS IoT 무선에 온보딩한 후 Amazon CloudWatch를 사용하여 무선 디바이스 및 게이트웨이를 실시간으로 로깅하고 모니터링할 수 있습니다. LoRaWAN 디바이스와 게이트웨이를 모니터링하려면 연결을 설정하고 추적 메시지 수신을 시작하는 데 걸리는 시간을 크게 줄이는 네트워크 분석기를 사용할 수도 있습니다.

- AWS IoT Core

  AWS IoT Core 통합을 사용하여 규칙 엔진에서 액세스할 수 있는 AWS 서비스에 연결할 수도 있습니다. 자세한 내용은 규칙 엔진에서 사용하는 AWS 서비스를 참조하세요.

AWS IoT Wireless에 액세스

콘솔, API 또는 CLI를 사용하여 LoRaWAN 및 Sidewalk 디바이스를 모두 온보딩할 수 있습니다.

- AWS IoT 콘솔 사용

  무선 디바이스를 온보딩하려면 AWS Management Console의 AWS IoT 무선 페이지를 사용하세요.

- AWS IoT 무선 API 사용

  AWS IoT 무선 API를 사용하여 Sidewalk 및 LoRaWAN 디바이스를 모두 온보딩할 수 있습니다. AWS IoT Core가 내장되어 있는 AWS IoT 무선 API는 AWS SDK에서 지원됩니다. 자세한 내용은 AWS SDK 및 도구 단원을 참조하세요.

- AWS CLI 사용

AWS CLI를 사용하여 LoRaWAN 및 Amazon Sidewalk 디바이스를 온보딩하고 관리하기 위한 명령을 실행할 수 있습니다. 자세한 내용은 AWS IoT 무선 CLI 참조를 참조하세요.

AWS IoT Wireless 시작하기

AWS 계정에 가입하고 IAM 사용자를 생성하는 단계를 따라 AWS IoT 무선를 시작할 수 있습니다. 가입한 후에는 AWS Management Console, AWS IoT 무선 API 또는 AWS CLI를 사용하여 Sidewalk 및 LoRaWAN 디바이스와 게이트웨이를 온보딩할 수 있습니다. 장치를 온보딩할 때 리소스를 더 쉽게 식별할 수 있도록 리소스를 설명하고 태그를 지정하는 방법을 고려하세요.

다음 주제에서는 AWS IoT 무선를 시작하는 방법을 보여 줍니다.

주제

- AWS IoT Wireless 설정
- AWS IoT 무선 리소스 설명

AWS IoT Wireless 설정

AWS에 가입하면 AWS IoT 무선를 포함하여 AWS의 모든 서비스에 AWS 계정 계정이 자동으로 등록됩니다. 사용자에게는 사용한 서비스에 대해서만 요금이 청구됩니다.

AWS IoT 무선를 설정하려면 다음 단원의 단계를 수행합니다.

주제

- AWS 계정 설정
- Python 및 AWS CLI 설치

AWS 계정 설정

AWS IoT Core for LoRaWAN 또는 Amazon Sidewalk용 AWS IoT Core를 처음 사용한다면 먼저 다음 태스크를 완료하여 AWS 계정을 설정합니다.

주제

- AWS 계정에 가입
- IAM 사용자를 생성합니다.
- IAM 사용자로 로그인

AWS 계정에 가입

AWS 계정가 없는 경우 다음 절차에 따라 계정을 생성합니다.

AWS 계정에 가입하려면

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   가입 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   AWS 계정에 가입하면 AWS 계정 루트 사용자이(가) 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스및 리소스에 액세스하는 권한이 있습니다. 보안 모범 사례는 관리 사용자에게 관리자 액세스 권한을 할당하고, 루트 사용자만 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

IAM 사용자를 생성합니다.

다음 옵션 중 하나를 선택하여 관리 사용자를 생성합니다.

관리자를 관리하는 방법 한 가지 선택	목적	By	다른 방법
IAM Identity Center에서 (권장)	단기 보안 인증 정보를 사용하여 AWS에 액세스합니다. 이는 보안 모범 사례와 일치합니다. 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.	AWS IAM Identity Center 사용 설명서의 시작하기 지침을 따르세요.	AWS Command Line Interface 사용 설명서의 AWS IAM Identity Center 사용할 AWS CLI 구성을 통해 프로그래밍 방식의 액세스를 구성합니다.

관리자를 관리하는 방법 한 가지 선택	목적	By	다른 방법
IAM에서 (권장되지 않음)	장기 보안 인증 정보를 사용하여 AWS에 액세스합니다.	IAM 사용 설명서의 첫 IAM 관리 사용자 및 사용자 그룹 만들기에 나온 지침을 따릅니다.	IAM 사용 설명서에 나온 IAM 사용자의 액세스 키 관리에서 프로그래밍 방식 액세스를 구성합니다.

IAM 사용자로 로그인

IAM 사용자를 생성한 후 IAM 사용자 이름과 암호를 사용하여 AWS에 로그인할 수 있습니다.

IAM 사용자로 로그인하기 전에 IAM 콘솔에서 IAM 사용자의 로그인 링크를 확인할 수 있습니다. IAM 대시보드의 IAM 사용자 로그인 링크에서 AWS 계정의 로그인 링크를 볼 수 있습니다. 로그인 링크의 URL에는 대시(-)가 없는 AWS 계정 ID가 포함되어 있습니다.

로그인 링크의 URL에 AWS 계정 ID가 포함되지 않게 하려면 계정 별칭을 생성합니다. 자세한 내용은 IAM 사용 설명서의 AWS 계정 별칭 생성, 삭제 및 나열을 참조하세요.

IAM 사용자로 로그인하기

1. 에서 로그아웃합니다AWS Management Console
2. 로그인 링크를 입력합니다. 로그인 링크에는 AWS 계정 ID(대시 제외) 또는 AWS 계정 별칭이 포함됩니다.

   ```
   https://aws_account_id_or_alias.signin.aws.amazon.com/console
   ```
3. 방금 생성한 IAM 사용자 이름과 암호를 입력합니다.

로그인하면 탐색 모음에 *'your_user_name @ your_aws_account_id'*가 표시됩니다.

Python 및 AWS CLI 설치

LoRaWAN 또는 Sidewalk 엔드 디바이스를 연결하기 전에 Python을 설치하고 AWS CLI를 구성해야 합니다.

> **Important**
>
> Sidewalk 엔드 디바이스의 프로비저닝 및 등록을 위한 전체 온보딩 워크플로를 수행하려면 Sidewalk 게이트웨이와 HDK도 설정해야 합니다. 지침은 Amazon Sidewalk 설명서의 Hardware Development Kit(HDK) 설정 및 Sidewalk 게이트웨이 설정을 참조하세요.

주제

- Python 및 Python3-pip 설치
- AWS CLI 설정

Python 및 Python3-pip 설치

다음 섹션에 설명된 대로 AWS CLI 및 boto3를 사용하려면 Python 버전 3.6 이상을 사용해야 합니다. AWS IoT 콘솔을 사용하여 엔드 디바이스를 온보딩하려는 경우 이 섹션을 건너뛰고 AWS 계정 설정을 계속할 수 있습니다. Python과 Python3-pip을 이미 설치했는지 확인하려면 다음 명령을 실행하세요. 이 명령을 실행하여 버전이 반환되면 Python과 Python3-pip이 올바르게 설치되었다는 뜻입니다.

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

이 명령으로 오류가 발생하는 경우 Python이 설치되지 않았거나 운영 체제가 Python v3.x 실행 파일을 Python3로 호출하기 때문일 수 있습니다. 이 경우 명령을 실행할 때 `python`의 모든 인스턴스를 `python3`로 바꾸세요. 그래도 오류가 발생하면 Python 설치 프로그램을 다운로드하여 실행하거나 아래에 설명된 대로 운영 체제에 따라 Python을 설치하세요.

Windows

Windows 시스템의 경우 Python 웹 사이트에서 Python을 다운로드한 다음 설치 프로그램을 실행하여 시스템에 Python을 설치합니다.

Linux

Ubuntu에 Python을 설치하려면 다음 `sudo` 명령을 실행합니다.

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

macOS

Mac 시스템에서는 Homebrew를 사용하여 Python을 설치합니다. Homebrew는 pip도 설치하는데, pip은 설치된 Python3 버전을 가리킵니다.

```
$ brew install python
```

AWS CLI 설정

다음 단계는 AWS CLI 및 boto3(Python용 AWS SDK)를 구성하는 방법을 보여 줍니다. 이러한 단계를 따르기 전에 먼저 AWS 계정에 가입하고 관리자 사용자를 생성해야 합니다. 지침은 AWS IoT Wireless 설정을(을) 참조하십시오.

1. AWS CLI 설치 및 구성

   AWS CLI를 사용하여 Sidewalk 엔드 디바이스를 프로그래밍 방식으로 Amazon Sidewalk용 AWS IoT Core에 온보딩할 수 있습니다. AWS IoT 콘솔을 사용하여 엔드 디바이스를 온보딩하려는 경우 이 섹션을 건너뛰세요. AWS IoT Core 콘솔을 열고 다음 단원으로 계속 진행하여 Amazon Sidewalk용 AWS IoT Core에 디바이스 연결을 시작하세요. AWS CLI 구성에 대한 지침은 AWS CLI 설치 및 구성을 참조하세요.

2. boto3(AWS SDK for Python) 설치

   다음 명령을 사용하여 boto3(AWS SDK for Python)와 AWS CLI를 설치합니다. boto3를 실행하는 데 필요한 botocore도 설치합니다. 자세한 지침은 Boto3 설명서 가이드의 Boto3 설치를 참조하세요.

   > **Note**
   >
   > `awscli` 버전 `1.26.6`에는 PyYAML 3.10 이상, 5.5 이하 버전이 필요합니다.

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

3. 보안 인증 정보 및 기본 리전 구성

   `~/.aws/credentials` 및 `~/.aws/config` 파일에서 보안 인증 정보와 기본 리전을 구성합니다. boto3 라이브러리는 이러한 보안 인증 정보를 사용하여 AWS 계정을 식별하고 API 호출을 승인합니다. 구성 지침은 다음을 참조하세요.

   - Boto3 설명서 가이드의 구성
   - AWS CLI 설명서 가이드의 구성 및 보안 인증 정보 파일 설정

AWS IoT 무선 리소스 설명

LoRaWAN 또는 Sidewalk 디바이스 온보딩을 시작하기 전에 디바이스, 게이트웨이 및 대상의 명명 규칙을 고려하세요. AWS IoT 무선는 사용자가 생성하는 리소스를 식별하는 몇 가지 옵션을 제공합니다. AWS IoT 무선 리소스는 생성될 때 고유 ID가 부여되는데, 이 ID는 설명적이지 않으며 리소스 생성 후 변경할 수도 없습니다. 또한 이름을 지정하고, 설명을 추가하고, 태그와 태그 값을 대부분의 AWS IoT 무선 리소스에 연결하여 리소스를 더욱 편리하게 선택, 식별 및 관리할 수 있습니다.

- 리소스 이름 및 설명

  디바이스, 게이트웨이, 및 프로필의 경우 리소스 이름은 리소스를 만든 후 변경할 수 있는 선택적 필드입니다. 리소스 허브 페이지에 표시된 목록에 이름이 나타납니다.

  대상에 대해 AWS 계정 및 AWS 리전에 고유한 이름을 제공합니다. 대상 리소스를 생성한 후에는 대상 이름을 변경할 수 없습니다.

  이름은 최대 256자까지 사용할 수 있지만 리소스 허브의 표시 공간은 제한됩니다. 가능한 경우 이름의 구별 부분이 처음 20~30자에 나타나도록 하세요.

- 리소스 태그

  태그는 AWS 리소스에 연결될 수 있는 메타데이터의 키-값 페어입니다. 태그 키와 해당 값을 모두 선택합니다.

  게이트웨이, 대상 및 프로파일에 최대 50개의 태그를 연결할 수 있습니다. 디바이스는 태그를 지원하지 않습니다.

리소스 이름 및 설명

이름에 대한 AWS IoT 무선 리소스 지원

Resource	이름 필드 지원	
대상	이름은 리소스의 고유 ID이며 변경할 수 없습니다.	
무선 디바이스	이름은 리소스의 선택적 설명자이며 변경할 수 있습니다.	
LoRaWAN 게이트웨이	이름은 리소스의 선택적 설명자이며 변경할 수 있습니다.	
프로필	이름은 리소스의 선택적 설명자이며 변경할 수 있습니다.	

이름 필드는 리소스 허브 목록에 나타나지만 공간이 제한되어 있으므로 이름의 처음 15~30자만 표시될 수 있습니다. 리소스의 이름을 선택할 때 리소스를 식별하는 방식과 콘솔에 표시되는 방식을 고려하세요.

설명

대상, 디바이스 및 게이트웨이 리소스는 최대 2,048자를 수용할 수 있는 설명 필드도 지원합니다. 설명 필드는 개별 리소스의 세부 정보 페이지에만 나타납니다. 설명 필드에는 많은 정보가 포함될 수 있지만 리소스의 세부 정보 페이지에만 표시되기 때문에 여러 리소스의 컨텍스트에서 검색하는 데 편리하지는 않습니다.

리소스 태그

AWS 태그에 대한 AWS IoT 무선 리소스 지원

Resource	AWS 태그 지원	
대상	최대 50개의 AWS 태그를 리소스에 추가할 수 있습니다.	
무선 디바이스	이 리소스는 AWS 태그를 지원하지 않습니다.	

Resource	AWS 태그 지원
LoRaWAN 게이트웨이	최대 50개의 AWS 태그를 리소스에 추가할 수 있습니다.
프로파일	최대 50개의 AWS 태그를 리소스에 추가할 수 있습니다.

태그란 AWS 리소스를 식별하고 정리할 때 사용할 수 있는 메타데이터 역할을 하는 단어 또는 문구입니다. 태그 키는 정보 범주로 간주하고 태그 값은 해당 범주의 특정 값으로 간주할 수 있습니다. 예를 들어, 태그 값이 색상(color)인 경우 어떤 리소스에는 해당 태그에 대해 파란색(blue) 값을 부여하고 다른 리소스에는 빨간색(red) 값을 부여할 수 있습니다. 이를 통해 AWS 콘솔의 태그 편집기를 사용하여 색상(color) 태그 값이 파란색(blue)인 리소스를 찾을 수 있습니다.

AWS IoT 무선의 태깅에 대한 자세한 내용은 AWS IoT 무선 리소스에 태그 지정 단원을 참조하세요.

태그 지정 및 태그 지정 전략에 대한 자세한 내용은 태그 편집기를 참조하세요.

AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWAN은 구성, 업데이트 서버(CUPS) 및 펌웨어 무선 업데이트(FUOTA) 기능을 사용하여 게이트웨이 관리를 제공하는 완전관리형 LoRaWAN Network Server(LNS)입니다. 프라이빗 LNS를 AWS IoT Core for LoRaWAN으로 바꾸고 장거리 광역 네트워크(LoRaWAN) 디바이스 및 게이트웨이를 AWS IoT Core에 연결할 수 있습니다. 이렇게 하면 유지 관리, 운영 비용, 설정 시간 및 간접비를 줄일 수 있습니다.

> **Note**
>
> AWS IoT Core for LoRaWAN은 IPv4 주소 형식만 지원합니다. IPv6 또는 듀얼 스택 구성(IPv4 및 IPv6)은 지원하지 않습니다. 자세한 내용은 IPv6를 지원하는 AWS 서비스를 참조하세요.

소개

LoRaWAN 디바이스는 LoRaWAN 프로토콜을 사용하여 라이선스가 없는 무선 스펙트럼에서 작동하는 장거리, 저전력, 배터리 작동 디바이스입니다. LoRaWAN은 LoRa에 구축된 저전력 광역 네트워크(LPWAN) 통신 프로토콜입니다. LoRa는 디바이스 간 저전력, 광역 통신을 가능하게 하는 물리적 계층 프로토콜입니다.

LoRaWAN 디바이스를 AWS IoT에 연결하려면 LoRaWAN 게이트웨이를 사용해야 합니다. 게이트웨이는 AWS IoT Core for LoRaWAN에 디바이스를 연결하고 메시지를 교환하는 브리지 역할을 합니다. AWS IoT Core for LoRaWAN은 AWS IoT 규칙 엔진을 사용하여 LoRaWAN 디바이스에서 다른 AWS IoT 서비스로 메시지를 라우팅합니다.

개발 노력을 줄이고 AWS IoT Core for LoRaWAN에 신속하게 디바이스를 온보딩하려면 LoRaWAN 인증 엔드 디바이스를 사용하는 것이 좋습니다. 자세한 내용은 AWS IoT Core for LoRaWAN 제품 개요 페이지를 참조하세요. 디바이스에 대해 LoRaWAN 인증을 받는 방법에 대한 자세한 내용은 LoRaWAN 제품 인증을 참조하세요.

AWS IoT Core for LoRaWAN 액세스

콘솔 또는 AWS IoT 무선 API를 사용하여 LoRaWAN 디바이스 및 게이트웨이를 AWS IoT Core for LoRaWAN에 빠르게 온보딩할 수 있습니다.

콘솔 사용

AWS Management Console을 사용하여 LoRaWAN 디바이스 및 게이트웨이를 온보딩하려면 AWS Management Console에 로그인하고 AWS IoT 콘솔에서 AWS IoT Core for LoRaWAN 페이지로 이동합니다. 그런 다음 소개 단원을 사용하여 AWS IoT Core for LoRaWAN에 게이트웨이와 디바이스를 추가할 수 있습니다. 자세한 내용은 콘솔을 사용하여 디바이스 및 게이트웨이를 AWS IoT Core for LoRaWAN에 온보딩 단원을 참조하십시오.

API 또는 CLI 사용

AWS IoT 무선 API를 사용하여 LoRaWAN 및 Sidewalk 디바이스를 모두 온보딩할 수 있습니다. AWS IoT Core for LoRaWAN가 내장되어 있는 AWS IoT 무선 API는 AWS SDK에서 지원됩니다. 자세한 내용은 AWS SDK 및 도구 단원을 참조하세요.

AWS CLI를 사용하여 LoRaWAN 게이트웨이 및 디바이스를 온보딩하고 관리하기 위한 명령을 실행할 수 있습니다. 자세한 내용은 AWS IoT 무선 CLI 참조를 참조하세요.

AWS IoT Core for LoRaWAN 리전 및 엔드포인트

AWS IoT Core for LoRaWAN은 AWS 리전에 특정한 컨트롤 플레인과 데이터 영역 API 엔드포인트에 대한 지원을 제공합니다. 데이터 영역 API 엔드포인트는 AWS 계정 및 AWS 리전에 한정됩니다. AWS IoT Core for LoRaWAN 엔드포인트에 대한 자세한 내용은 AWS 일반 참조의 AWS IoT Core for LoRaWAN 엔드포인트를 참조하세요.

디바이스와 AWS IoT 간의 보다 안전한 통신을 위해 디바이스를 퍼블릭 인터넷을 통해 연결하는 대신 Virtual Private Cloud(VPC)에 있는 AWS PrivateLink를 통해 AWS IoT Core for LoRaWAN에 연결할 수 있습니다. 자세한 내용은 AWS IoT Core for LoRaWAN 및 인터페이스 VPC 엔드포인트(AWS PrivateLink) 단원을 참조하십시오.

AWS IoT Core for LoRaWAN은 디바이스 간에 전송되는 디바이스 데이터에 적용되는 할당량과 AWS IoT 무선 API 작업을 위한 최대 TPS가 있습니다. 자세한 정보는 AWS 일반 참조의 AWS IoT Core for LoRaWAN 할당량을 참조하세요.

AWS IoT Core for LoRaWAN 요금

신규 고객이 AWS에 가입하면 무상으로 AWS IoT Core for LoRaWAN를 시작할 수 있는 AWS 프리 티어를 제공합니다. AWS IoT Core for LoRaWAN는 사용한 만큼만 비용을 지불하면 됩니다. 일반 제품 개요 및 요금에 대한 자세한 내용은 AWS IoT Core 요금을 참조하세요.

AWS IoT Core for LoRaWAN(이)란 무엇인가요?

AWS IoT Core for LoRaWAN은 LoRaWAN 디바이스 및 게이트웨이를 AWS에 연결하여 프라이빗 LoRaWAN Network Server(LNS)를 대체합니다. AWS IoT 규칙 엔진을 사용하여 LoRaWAN 디바이스에서 받은 메시지를 라우팅할 수 있으며, 여기에서 형식이 지정되어 다른 AWS IoT 서비스로 전송할 수 있습니다. AWS IoT와의 통신 보호를 위해 AWS IoT Core for LoRaWAN은 X.509 인증서를 사용합니다.

AWS IoT Core for LoRaWAN은 AWS IoT Core가 LoRaWAN 게이트웨이 및 디바이스와 통신하는 데 필요한 서비스 및 디바이스 정책을 관리합니다. AWS IoT Core for LoRaWAN은 디바이스 데이터를 다른 서비스에 전송하는 AWS IoT 규칙을 설명하는 대상도 관리합니다.

AWS IoT Core for LoRaWAN의 기능

AWS IoT Core for LoRaWAN를 사용하여 다음을 수행할 수 있습니다.

- 프라이빗 LNS를 설정하고 관리할 필요 없이 LoRaWAN 디바이스와 게이트웨이를 AWS IoT에 온보딩하고 연결합니다.
- LoRa Alliance에 의해 표준화된 1.0.x 또는 1.1 LoRaWAN 사양을 준수하는 LoRaWAN 디바이스를 연결합니다. 이러한 디바이스는 클래스 A, 클래스 B 또는 클래스 C 모드에서 작동할 수 있습니다.
- LoRa Basics Station 버전 2.0.4 이상을 지원하는 LoRaWAN 게이트웨이를 사용합니다. AWS IoT Core for LoRaWAN에 적합한 모든 게이트웨이는 호환 가능한 버전의 LoRa Basics Station을 실행합니다.
- 공개적으로 사용 가능한 LoRaWAN 네트워크를 사용하여 LoRaWAN 디바이스를 클라우드에 연결하면 배포 시간이 단축되고 프라이빗 LoRaWAN 네트워크를 관리할 필요가 없으므로 시간과 비용이 절약됩니다.
- AWS IoT Core for LoRaWAN의 적응형 데이터 속도를 사용하여 신호 강도, 대역폭 및 확산 요인을 모니터링하고 필요한 경우 데이터 속도를 최적화합니다. 네트워크 분석기를 사용하면 LoRaWAN 리소스를 실시간으로 모니터링할 수 있습니다.
- CUPS 서비스를 사용하여 LoRaWAN 게이트웨이의 펌웨어를 업데이트하고 펌웨어 무선 업데이트(FUOTA)를 사용하여 LoRaWAN 디바이스의 펌웨어를 업데이트합니다.

다음 주제에서는 LoRaWAN 기술 및 AWS IoT Core for LoRaWAN에 대한 자세한 정보를 제공합니다.

주제

- LoRaWAN이란 무엇입니까?

- AWS IoT Core for LoRaWAN 작동 방식

LoRaWAN이란 무엇입니까?

LoRa Alliance는 LoRaWAN을 "배터리로 작동되는 '사물'을 지역, 국가 또는 글로벌 네트워크의 인터넷에 무선으로 연결하도록 설계된 저전력, 광역(LPWA) 네트워킹 프로토콜이라고 설명하며 양방향 통신, 엔드 투 엔드 보안, 이동성 및 로컬라이제이션 서비스와 같은 주요 사물 인터넷(IoT) 요구 사항을 대상으로 합니다."

LoRa 및 LoRaWAN

LoRaWAN 프로토콜은 LoRa에서 작동하는 저전력 광역 네트워크(LPWAN) 통신 프로토콜입니다.

LoRaWAN은 저전력 광역 네트워킹의 국제 표준으로 인정받고 있습니다. 자세한 내용은 공식적으로 ITU 국제 표준으로 인정받은 LoRaWAN을 참조하세요. LoRaWAN 사양은 누구나 LoRa 네트워크를 설정하고 운영할 수 있도록 개방되어 있습니다.

LoRa는 라이선스가 필요 없는 무선 주파수 스펙트럼에서 작동하는 무선 오디오 주파수 기술입니다. LoRa는 확산 스펙트럼 변조를 사용하고 좁은 대역폭의 비용으로 장거리 통신을 지원하는 물리 계층 프로토콜입니다. 중앙 주파수가 있는 협대역 파형을 사용하여 데이터를 전송하므로 간섭에 강력합니다.

LoRWAN 기술의 특징

- 시야가 최대 10마일 떨어진 장거리 통신.
- 최대 10년의 긴 배터리 수명. 배터리 수명을 늘리려면 클래스 A 또는 클래스 B 모드에서 디바이스를 작동할 수 있으며, 이로 인해 다운링크 지연 시간이 늘어납니다.
- 저렴한 디바이스 및 유지 보수 비용.
- 라이선스가 없는 무선 스펙트럼이지만 리전별 규정이 적용됩니다.
- 저전력이지만 데이터 전송률에 따라 51바이트~241바이트로 페이로드 크기가 제한됩니다. 데이터 전송률은 222 최대 페이로드 크기로 0.3Kbit/s~27Kbit/s일 수 있습니다.

LoRaWAN 프로토콜 버전

LoRa Alliance는 LoRaWAN 사양 문서를 사용하여 LoRaWAN 프로토콜을 지정합니다. 지역별 규정을 설명하기 위해 LoRa Alliance는 지역 파라미터 문서도 게시합니다. 자세한 내용은 LoRaWAN 지역 파라미터 및 사양을 참조하세요.

LoRaWAN의 초기 릴리스는 버전 1.0입니다. 릴리스된 추가 버전은 1.0.1, 1.0.2, 1.0.3, 1.0.4, 1.1입니다. 버전 1.0.1~1.0.4는 일반적으로 1.0.x라고 지칭합니다.

LoRaWAN에 대해 자세히 알아보기

다음 링크에는 LoRaWAN 기술 및 LoRa Basic Station에 대한 유용한 정보가 포함되어 있습니다. LoRa Basic Station은 엔드 디바이스를 AWS IoT Core for LoRaWAN에 연결하기 위해 LoraWAN 게이트웨이에서 실행되는 소프트웨어입니다.

- ITU 국제 표준으로 인정받은 LoRaWAN

  LoRaWAN은 저전력 광역 네트워킹의 국제 표준으로 공식 문서화되었습니다. 이 표준의 제목은 권장 사항 ITU-T Y.4480 '광역 무선 네트워크를 위한 저전력 프로토콜'입니다.

- LoRaWAN의 사물 기본 사항

  LoRaWAN의 사물 기본 사항에는 LoRaWAN의 기본 사항을 다루는 소개 동영상 및 LoRA 및 LoraWAN에 대해 알아볼 수 있는 일련의 장이 포함되어 있습니다.

- LoRaWAN이란 무엇입니까?

  LoRa Alliance는 여러 리전의 LoRaWAN 사양에 대한 요약을 포함하여 LoRA 및 LoRaWAN의 기술 개요를 제공합니다.

- LoRa Basics Station

  Semtech Corporation은 게이트웨이 및 엔드 노드의 LoRA 기본 사항에 대한 유용한 개념을 제공합니다. LoRaWAN 게이트웨이에서 실행되는 오픈 소스 소프트웨어인 LoRa Basic Station은 Semtech Corporation의 GitHub 리포지토리를 통해 유지 관리되고 배포됩니다. LoraWAN 데이터를 교환하고 구성 업데이트를 수행하는 방법을 설명하는 LNS 및 CUPS 프로토콜에 대해서도 알아볼 수 있습니다.

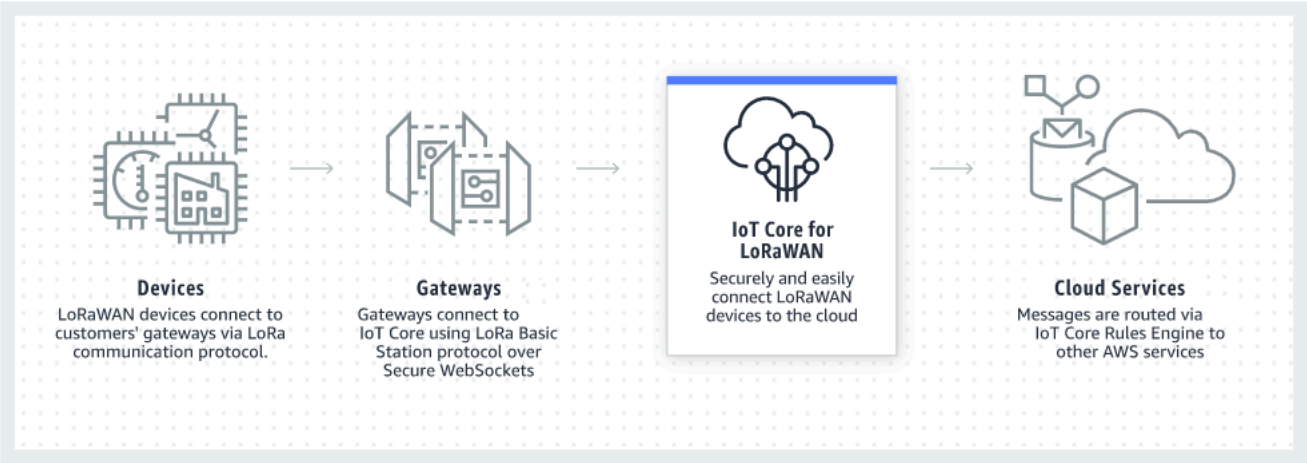
- LoRaWAN 지역 파라미터 및 사양

  RP002-1.0.2 문서에는 모든 버전의 LoRaWAN 계층 2 사양에 대한 지원이 포함됩니다. 여기에는 LoRaWAN 사양 및 지역 파라미터, 다양한 LoRaWAN 버전에 대한 정보가 포함됩니다.

AWS IoT Core for LoRaWAN 작동 방식

LoRaWAN 네트워크 아키텍처는 게이트웨이가 엔드 디바이스와 LoRaWAN 네트워크 서버(LNS) 간에 정보를 릴레이하는 대표적인 스타형 토폴로지에 배포됩니다. 다음은 LoRaWAN 디바이스가 AWS IoT

Core for LoRaWAN과 상호 작용하는 방법을 보여 줍니다. AWS IoT Core for LoRaWAN이 LNS 역할을 하고 AWS 클라우드에서 다른 AWS 서비스와 통신하는 방법도 보여 줍니다.



LoRaWAN 디바이스는 LoRaWAN 게이트웨이를 통해 AWS IoT Core와 통신합니다. AWS IoT Core for LoRaWAN은 AWS IoT Core가 LoRaWAN 게이트웨이 및 디바이스를 관리하고 이와 통신하는 데 필요한 서비스 및 디바이스 정책을 관리합니다. AWS IoT Core for LoRaWAN은 디바이스 데이터를 다른 서비스에 전송하는 AWS IoT 규칙을 설명하는 대상도 관리합니다.

AWS IoT Core for LoRaWAN 사용 시작

다음 단계는 AWS IoT Core for LoRaWAN 사용을 시작할 수 있는 방법에 대한 개요를 보여 줍니다.

1. 필요한 무선 디바이스 및 LoRaWAN 게이트웨이를 선택합니다.

   AWS 파트너 디바이스 카탈로그에는 AWS IoT Core for LoRaWAN에 사용할 수 있는 게이트웨이 및 개발자 키트가 포함되어 있습니다. 자세한 내용은 AWS 파트너 디바이스 카탈로그에서 정규화된 게이트웨이 사용 단원을 참조하십시오.
2. 무선 디바이스 및 LoRaWAN 게이트웨이를 AWS IoT Core for LoRaWAN에 추가합니다.

   AWS IoT Core for LoRaWAN에 게이트웨이 및 디바이스 연결에서는 리소스를 설명하고 AWS IoT Core for LoRaWAN에 무선 디바이스 및 LoRaWAN 게이트웨이를 추가하는 방법에 대한 정보를 제공합니다. 또한 이러한 디바이스를 관리하고 해당 데이터를 AWS 서비스에 전송하는 데 필요한 다른 AWS IoT Core for LoRaWAN 리소스를 구성하는 방법에 대해 알아봅니다.
3. AWS IoT Core for LoRaWAN 솔루션을 완성합니다.

   샘플 AWS IoT Core for LoRaWAN 솔루션을 시작해서 본인의 것으로 만드세요.

AWS IoT Core for LoRaWAN 리소스

다음 리소스는 AWS IoT Core for LoRaWAN에 대한 자세한 내용과 시작 방법에 대해 학습하는 데 도움이 됩니다.

- AWS IoT Core for LoRaWAN 시작하기

  다음 동영상은 AWS IoT Core for LoRaWAN의 작동 방법에 대해 설명하며 AWS Management Console에서 LoraWan 게이트웨이를 추가하는 프로세스를 안내합니다.

- AWS IoT Core for LoRaWAN 워크숍

  이 워크숍에서는 LoRaWAN 기술의 기본 사항과 AWS IoT Core for LoRaWAN을 통한 구현에 대해 다룹니다. 또한 워크숍을 통해 샘플 IoT 솔루션을 구축하기 위해 AWS IoT Core for LoRaWAN에 게이트웨이와 디바이스를 연결하는 방법을 보여 주는 실습을 진행할 수 있습니다.

- AWS IoT를 사용하여 저전력 광역 네트워크(LPWAN) 솔루션 구현

  이 문서는 LPWAN이 IoT 사용 사례에 적합한 선택인지 결정하는 데 도움이 되는 의사 결정 프레임워크를 제공하고, LPWAN 연결 기술 및 기능에 대한 개요를 제공하고, 구현 지침을 제공합니다.

AWS IoT Core for LoRaWAN에 게이트웨이 및 디바이스 연결

AWS IoT Core for LoRaWAN은 LoRaWAN(저전력 장거리 광역 네트워크) 디바이스를 연결하고 관리할 수 있도록 지원하며 LNS 개발 및 운영의 필요성을 대체합니다. 장거리 WAN(LoRaWAN) 디바이스 및 게이트웨이는 AWS IoT Core for LoRaWAN을 사용하여 AWS IoT Core에 연결할 수 있습니다.

디바이스, 게이트웨이, 프로파일 및 대상에 대한 명명 규칙

AWS IoT Core for LoRaWAN을 시작하고 리소스를 생성하기 전에 디바이스, 게이트웨이 및 대상의 명명 규칙을 고려하세요.

AWS IoT Core for LoRaWAN은 무선 디바이스, 게이트웨이 및 프로파일에 대해 생성한 리소스에 고유 ID를 할당합니다. 그러나 리소스를 보다 쉽게 식별할 수 있도록 리소스에 보다 구체적인 이름을 지정할 수도 있습니다. 디바이스, 게이트웨이, 프로파일 및 대상을 AWS IoT Core for LoRaWAN에 추가하기 전에 더 쉽게 관리할 수 있도록 이름을 지정하는 방법을 고려해 보세요.

리소스를 만들 때 태그를 리소스에 추가할 수 있습니다. LoRaWAN 디바이스를 추가하기 전에 태그를 사용하여 AWS IoT Core for LoRaWAN 리소스를 식별하고 관리하는 방법을 고려하세요. 태그를 추가한 후 수정할 수 있습니다.

이름 및 태그 지정에 대한 자세한 내용은 AWS IoT 무선 리소스 설명 단원을 참조하세요.

디바이스 데이터를 서비스 데이터에 매핑

LoRaWAN 무선 디바이스의 데이터는 대역폭을 최적화하기 위해 인코딩되는 경우가 많습니다. 이러한 인코딩된 메시지는 다른 AWS 서비스에 의해 쉽게 사용되지 않을 형식으로 AWS IoT Core for LoRaWAN에 도달합니다. AWS IoT Core for LoRaWAN은 AWS Lambda 함수를 사용할 수 있는 AWS IoT 규칙을 사용하여 다른 AWS 서비스가 사용할 수 있는 형식으로 디바이스 메시지를 처리 및 디코딩합니다.

디바이스 데이터를 변환하여 다른 AWS 서비스에 전송하려면 다음 사항을 알아야 합니다.

- 무선 디바이스가 보내는 데이터의 형식 및 내용입니다.
- 데이터를 전송하려는 서비스입니다.
- 서비스에 필요한 형식입니다.

이 정보를 사용하여 변환을 수행하고 변환된 데이터를 사용할 AWS 서비스에 전송하는 AWS IoT 규칙을 사용할 수 있습니다.

콘솔을 사용하여 디바이스 및 게이트웨이를 AWS IoT Core for LoRaWAN에 온보딩

콘솔 인터페이스 또는 API를 사용하여 LoraWAN 게이트웨이 및 디바이스를 추가할 수 있습니다. AWS IoT Core for LoRaWAN을 처음 사용하는 경우 콘솔을 사용하는 것이 좋습니다. 콘솔 인터페이스는 한 번에 몇 가지 AWS IoT Core for LoRaWAN 리소스를 관리할 때 가장 실용적입니다. 많은 수의 AWS IoT Core for LoRaWAN 리소스를 관리할 때 AWS IoT 무선 API를 사용하여 보다 자동화된 솔루션을 만드는 것이 좋습니다.

AWS IoT Core for LoRaWAN 리소스를 구성할 때 입력하는 대부분의 데이터는 디바이스 공급 업체에서 제공하며 지원하는 LoRaWAN 사양에 따라 다릅니다. 다음 주제에서는 AWS IoT Core for LoRaWAN 리소스를 설명하고 게이트웨이와 디바이스를 추가할 콘솔 또는 API를 사용하는 방법에 대해 설명합니다.

> **Note**
>
> 퍼블릭 네트워크를 사용하여 LoRaWAN 디바이스를 클라우드에 연결하는 경우 게이트웨이 온보딩을 건너뛸 수 있습니다. 자세한 내용은 퍼블릭 LoRaWAN 디바이스 네트워크에서 LoRaWAN 트래픽 관리(Everynet) 단원을 참조하십시오.

주제

- AWS IoT Core for LoRaWAN에 게이트웨이 온보딩
- AWS IoT Core for LoRaWAN에 디바이스 온보딩

AWS IoT Core for LoRaWAN에 게이트웨이 온보딩

AWS IoT Core for LoRaWAN을 처음 사용하려는 경우 콘솔을 사용하여 첫 번째 LoRaWAN 게이트웨이 및 디바이스를 추가할 수 있습니다.

> **Note**
>
> 퍼블릭 네트워크를 사용하여 LoRaWAN 디바이스를 클라우드에 연결하는 경우 게이트웨이 온보딩을 건너뛸 수 있습니다. 자세한 내용은 퍼블릭 LoRaWAN 디바이스 네트워크에서 LoRaWAN 트래픽 관리(Everynet) 단원을 참조하십시오.

게이트웨이를 온보딩하기 전에

AWS IoT Core for LoRaWAN에 게이트웨이를 온보딩하기 전에 다음을 수행할 것을 권장합니다.

- AWS IoT Core for LoRaWAN에서 사용할 수 있는 게이트웨이를 사용합니다. 이러한 게이트웨이는 추가 구성 설정 없이 AWS IoT Core에 연결되며 LoRa Basics Station 소프트웨어의 2.0.4 이상 버전을 실행합니다. 자세한 내용은 AWS IoT 무선를 통한 게이트웨이 관리 단원을 참조하십시오.
- 리소스를 보다 쉽게 관리할 수 있도록 만드는 리소스의 명명 규칙을 고려하세요. 자세한 내용은 AWS IoT 무선 리소스 설명 단원을 참조하십시오.
- 각 게이트웨이에 고유한 구성 파라미터를 미리 입력할 준비가 되어 있으므로 콘솔에 데이터를 보다 원활하게 입력할 수 있습니다. AWS IoT이(가) 게이트웨이와 통신하고 게이트웨이를 관리하는 데 필요한 무선 게이트웨이 구성 파라미터에는 게이트웨이의 EUI 및 LoRa 주파수 대역이 포함됩니다.

AWS IoT Core for LoRaWAN에 게이트웨이 온보딩:

- 주파수 대역 선택을 고려하고 필요한 IAM 역할 추가
- AWS IoT Core for LoRaWAN에 게이트웨이 추가
- LoRaWAN 게이트웨이를 연결하고 연결 상태를 확인합니다.

주파수 대역 선택을 고려하고 필요한 IAM 역할 추가

AWS IoT Core for LoRaWAN에 게이트웨이를 추가하기 전에 게이트웨이가 작동할 주파수 대역을 고려하고 게이트웨이를 AWS IoT Core for LoRaWAN에 연결하는 데 필요한 IAM 역할을 추가하는 것이 좋습니다.

> **Note**
>
> 콘솔을 사용하여 게이트웨이를 추가하는 경우 콘솔에서 역할 생성을 클릭하여 필요한 IAM 역할을 생성한 다음 이 단계를 건너뛸 수 있습니다. CLI를 사용하여 게이트웨이를 만드는 경우에만 이러한 단계를 수행해야 합니다.

게이트웨이 및 디바이스 연결을 위한 LoRa 주파수 대역 선택 고려

AWS IoT Core for LoRaWAN은 EU863-870, US902-928, AU915 및 AS923-1 주파수 대역을 지원하는데, 이 대역의 주파수 범위 및 특성을 지원하는 국가에 물리적으로 존재하는 게이트웨이 및 디바이스를 연결하는 데 사용할 수 있습니다. EU863-870 및 US902-928 대역은 일반적으로 유럽과 북미에서 각각 사용됩니다. AS923-1 대역은 일반적으로 호주, 뉴질랜드, 일본, 싱가포르에서 사용됩니다. AU915는 호주와 아르헨티나에서 사용됩니다. 해당 리전 또는 국가에서 사용할 주파수 대역에 대한 자세한 내용은 LoRaWAN® 리전 파라미터를 참조하세요.

LoRa Alliance는 LoRa Alliance 웹사이트에서 다운로드할 수 있는 LoRaWAN 사양 및 리전 파라미터 문서를 게시합니다. LoRa Alliance 리전 파라미터는 회사가 해당 리전 또는 국가에서 사용할 주파수 대역을 결정하는 데 도움이 됩니다. AWS IoT Core for LoRaWAN 주파수 대역 구현은 리전 파라미터 사양 문서의 권장 사항을 따릅니다. 이러한 리전 파라미터는 산업, 과학 및 의료(ISM) 대역에 맞게 조정되는 주파수 할당과 함께 무선 파라미터 집합으로 그룹화됩니다. 규정 준수 팀과 협력하여 적용 가능한 규정 요구 사항을 충족하는지 확인하는 것이 좋습니다.

CUPS(Configuration and Update Server)가 게이트웨이 자격 증명을 관리할 수 있도록 IAM 역할을 추가합니다.

이 절차에서는 CUPS(Configuration and Update Server)가 게이트웨이 자격 증명을 관리할 수 있도록 IAM 역할을 추가하는 방법에 대해 설명합니다. LoRaWAN 게이트웨이가 AWS IoT Core for LoRaWAN과의 연결을 시도하기 전에 이 절차를 수행합니다. 그러나 이 작업은 한 번만 하면 됩니다.

CUPS(Configuration and Update Server)가 게이트웨이 자격 증명을 관리할 수 있도록 IAM 역할을 추가합니다.

1. IAM 콘솔의 역할 허브를 열고 역할 생성을 선택합니다.

2. IoTWirelessGatewayCertManagerRole 역할을 이미 추가했다고 생각되면 검색 창에 **`IoTWirelessGatewayCertManagerRole`**을 입력합니다.

   검색 결과에 IoTWirelessGatewayCertManagerRole 역할이 있으면 필요한 IAM 역할을 갖게 됩니다. 절차를 지금 종료할 수 있습니다.

   검색 결과가 비어 있으면 필요한 IAM 역할이 없는 것입니다. 절차를 계속하여 항목을 추가합니다.
3. 신뢰할 수 있는 엔터티 유형 선택(Select type of trusted entity)에서 다른 AWS 계정을 선택합니다.
4. 계정 ID에 AWS 계정 ID를 입력한 후 다음: 권한을 선택합니다.
5. 검색 상자에 **`AWSIoTWirelessGatewayCertManager`**을(를) 입력합니다.
6. 검색 결과 목록에서 AWSIoTWirelessGatewayCertManager라는 정책을 선택합니다.
7. Next: Tags(다음: 태그)를 선택한 후 Next: Review(다음: 검토)를 선택합니다.
8. 역할 이름에 **`IoTWirelessGatewayCertManagerRole`**을 입력한 다음 역할 생성을 선택합니다.
9. 새 역할을 편집하려면 확인 메시지에서 IoTWirelessGatewayCertManagerRole을 선택합니다.
10. 요약(Summary)에서 신뢰 관계(Trust relationships) 탭을 선택한 다음 신뢰 관계 편집(Edit trust relationship)을 선택합니다.
11. 정책 문서에서 `Principal` 속성을 다음 예시처럼 변경합니다.

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

`Principal` 속성을 변경한 후 전체 정책 문서가 다음 예시와 같은 형식이어야 합니다.

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

```
}
```

12. 변경 사항을 저장하고 종료하려면 신뢰 정책 업데이트(Update Trust Policy)를 선택합니다.

이제 IoTWirelessGatewayCertManagerRole이 생성됐습니다. 이 작업을 다시 수행할 필요가 없습니다.

게이트웨이를 추가하는 동안 이 절차를 수행한 경우 이 창과 IAM 콘솔을 닫고 AWS IoT 콘솔로 이동하여 게이트웨이 추가를 완료합니다.

AWS IoT Core for LoRaWAN에 게이트웨이 추가

콘솔 또는 CLI를 사용하여 AWS IoT Core for LoRaWAN에 게이트웨이를 추가할 수 있습니다.

게이트웨이를 추가하기 전에 AWS IoT Core for LoRaWAN에 게이트웨이 온보딩의 게이트웨이를 온보딩하기 전에 단원에서 언급한 요소를 고려하는 것이 좋습니다.

게이트웨이를 처음 추가하는 경우 콘솔을 사용하는 것이 좋습니다. 대신 CLI를 사용하여 게이트웨이를 추가하려면 게이트웨이가 AWS IoT Core for LoRaWAN에 연결할 수 있도록 필요한 IAM 역할을 생성해야 합니다. 역할 생성 방법에 대한 자세한 내용은 CUPS(Configuration and Update Server)가 게이트웨이 자격 증명을 관리할 수 있도록 IAM 역할을 추가합니다. 단원을 참조하세요.

콘솔을 사용하여 게이트웨이 추가

AWS IoT 콘솔의 AWS IoT Core for LoRaWAN 소개(Intro) 페이지로 이동하여 시작하기(Get started)를 선택한 다음 게이트웨이 추가(Add gateway)를 선택합니다. 게이트웨이를 이미 추가한 경우 게이트웨이 보기(View gateway)을 선택하여 추가한 게이트웨이를 봅니다. 게이트웨이를 더 추가하려면 게이트웨이 추가(Add gateway)를 선택합니다.

1. 게이트웨이 세부 정보 및 주파수 대역 정보 제공

   게이트웨이 세부 정보(Gateway details) 단원을 참조하여 게이트웨이의 EUI 및 주파수 대역 구성과 같은 디바이스 구성 데이터에 대한 정보를 제공합니다.

   - 게이트웨이 EUI

     개별 게이트웨이 디바이스의 EUI(확장 고유 식별자)입니다. EUI는 LoRaWAN 네트워크에서 게이트웨이를 고유하게 식별하는 `c0ee40ffff29df10`과 같은 16자리 영숫자 코드입니다. 이 정보는 게이트웨이 모델에 따라 다르며 게이트웨이 디바이스 또는 사용 설명서에서 찾을 수 있습니다.

> **Note**
>
> 게이트웨이의 EUI는 게이트웨이 디바이스에 인쇄되어 있을 수 있는 Wi-Fi MAC 주소와 다릅니다. EUI는 게이트웨이를 고유하게 식별하는 EUI-64 표준을 따르기 때문에 다른 AWS 계정 및 리전에서 사용할 수 없습니다.

- 주파수 대역(RFRegion)

  게이트웨이의 주파수 대역입니다. 게이트웨이가 지원하는 것과 게이트웨이가 물리적으로 연결하는 국가 또는 리전에 따라 US915, EU868, AU915 또는 AS923-1에서 선택할 수 있습니다. 대역에 대한 자세한 내용은 게이트웨이 및 디바이스 연결을 위한 LoRa 주파수 대역 선택 고려 단원을 참조하세요.

2. 무선 게이트웨이 구성 데이터 지정(선택 사항)

   이러한 필드는 선택 사항이며 게이트웨이 및 해당 구성에 대한 추가 정보를 제공하는 데 사용할 수 있습니다.

   - 게이트웨이의 이름, 설명 및 태그

     이러한 선택적 필드의 정보는 무선 시스템의 요소를 구성하고 설명하는 방법에 따라 제공됩니다. 이름을 게이트웨이에 지정하고, 설명 필드를 사용하여 게이트웨이에 대한 정보를 제공하고, 태그를 사용하여 게이트웨이에 대한 메타데이터의 키-값 페어를 추가할 수 있습니다. 리소스 이름 지정 및 설명에 대한 자세한 내용은 AWS IoT 무선 리소스 설명 단원을 참조하세요.

   - 하위 밴드 및 필터를 사용한 LoRaWAN 구성

     선택적으로, LoRaWAN 구성 데이터(예: 사용하려는 하위 밴드, 트래픽 흐름을 제어할 수 있는 필터)를 지정할 수도 있습니다. 이 자습서에서는 이 필드를 건너뛸 수 있습니다. 자세한 내용은 게이트웨이의 하위 밴드 및 필터링 기능 구성 단원을 참조하십시오.

3. AWS IoT 사물과 게이트웨이 연결

   AWS IoT 사물의 생성 여부를 지정하고 이를 게이트웨이와 연결합니다. AWS IoT의 사물을 사용하면 디바이스를 더 쉽게 검색하고 관리할 수 있습니다. 게이트웨이와 사물을 연결하면 게이트웨이가 다른 AWS IoT Core 기능에 액세스할 수 있습니다.

4. 게이트웨이 인증서 만들기 및 다운로드

   게이트웨이가 AWS IoT와 안전하게 통신할 수 있도록 게이트웨이를 인증하려면 LoRaWAN 게이트웨이가 프라이빗 키와 인증서를 AWS IoT Core for LoRaWAN에 제시해야 합니다. AWS IoT이(가)

X.509 표준을 사용하여 게이트웨이의 자격 증명을 확인할 수 있도록 게이트웨이 인증서를 생성합니다.

인증서 생성(Create certificate) 버튼을 클릭하고 인증서 파일을 다운로드합니다. 이것은 나중에 게이트웨이를 구성하는 데 사용합니다.

5. CUPS 및 LNS 엔드포인트 복사 및 인증서 다운로드

   AWS IoT Core for LoRaWAN에 연결을 설정할 때 LoraWAN 게이트웨이는 CUPS 또는 LNS 엔드포인트에 연결해야 합니다. CUPS 엔드포인트는 구성 관리도 제공하므로 CUPS 엔드포인트를 사용하는 것이 좋습니다. AWS IoT Core for LoRaWAN 엔드포인트의 신뢰성을 확인하기 위해 게이트웨이는 각 CUPS 및 LNS 엔드포인트에 대해 신뢰 인증서를 사용합니다.

   복사 버튼을 클릭하여 CUPS 및 LNS 엔드포인트를 복사해 둡니다. 나중에 게이트웨이를 구성할 때 이 정보가 필요합니다. 서버 신뢰 인증서 다운로드 버튼을 클릭하여 CUPS 및 LNS 엔드포인트에 대한 신뢰 인증서를 다운로드합니다.

6. 게이트웨이 권한에 대한 IAM 역할 생성

   CUPS(Configuration and Update Server)가 게이트웨이 자격 증명을 관리할 수 있도록 허용하는 IAM 역할을 추가해야 합니다.

   > **Note**
   >
   > 이 단계에서는 IoTWirelessGatewayCertManager 역할을 생성합니다. 이 역할을 이미 생성했다면 이 단계를 건너뛸 수 있습니다. LoRaWAN 게이트웨이가 AWS IoT Core for LoRaWAN과 연결을 시도하기 전에 이 작업을 수행해야 하며, 한 번만 수행하면 됩니다.

   계정에 대한 IoTWirelessGatewayCertManager IAM 역할을 생성하려면 역할 생성 버튼을 클릭합니다. 역할이 이미 존재할 경우 드롭다운 목록에서 선택합니다.

   제출을 클릭하여 게이트웨이 생성을 완료합니다.

API를 사용하여 게이트웨이 추가

API 또는 CLI를 사용하여 처음으로 게이트웨이를 추가하는 경우 IoTWirelessGatewayCertManager IAM 역할을 추가해야 게이트웨이가 AWS IoT Core for LoRaWAN과 연결할 수 있습니다. 역할 생성 방법에 대한 자세한 내용은 CUPS(Configuration and Update Server)가 게이트웨이 자격 증명을 관리할 수 있도록 IAM 역할을 추가합니다. 단원을 참조하세요.

다음 목록에서는 LoraWAN 게이트웨이 추가, 업데이트 또는 삭제와 관련된 작업을 수행하는 API 작업에 대해 설명합니다.

AWS IoT 무선 게이트웨이에 대한 AWS IoT Core for LoRaWAN API 작업

- CreateWirelessGateway
- GetWirelessGateway
- ListWirelessGateways
- UpdateWirelessGateway
- DeleteWirelessGateway

AWS IoT Core for LoRaWAN 리소스 생성 및 관리에 사용할 수 있는 작업 및 데이터 유형의 전체 목록은 AWS IoT 무선 API 참조를 참조하세요.

AWS CLI를 사용하여 게이트웨이를 추가하는 방법

AWS CLI를 사용하여 create-wireless-gateway 명령으로 무선 게이트웨이를 만들 수 있습니다. 다음 예제에서는 무선 LoRaWAN 디바이스 게이트웨이를 만듭니다. 게이트웨이 인증서 및 프로비저닝 자격 증명과 같은 추가 세부 정보가 포함된 `input.json` 파일을 제공할 수도 있습니다.

> **Note**
>
> 여기에 표시된 CLI 명령에 해당하는 AWS API의 메서드를 사용하여 API로 이 절차를 수행할 수도 있습니다.

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

사용할 수 있는 CLI에 대한 자세한 내용은 AWS CLI 참조를 참조하세요.

LoRaWAN 게이트웨이를 연결하고 연결 상태를 확인합니다.

게이트웨이 연결 상태를 확인하려면 게이트웨이를 이미 추가하고 AWS IoT Core for LoRaWAN에 연결되어 있어야 합니다. 게이트웨이를 추가하는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 추가 단원을 참조하세요.

AWS IoT Core for LoRaWAN에 게이트웨이 연결

게이트웨이를 추가한 후 게이트웨이의 구성 인터페이스에 연결하여 구성 정보 및 신뢰 인증서를 입력합니다.

AWS IoT Core for LoRaWAN에 게이트웨이 정보를 추가한 후 일부 AWS IoT Core for LoRaWAN 정보를 게이트웨이 디바이스에 추가합니다. 게이트웨이 공급 업체에서 제공하는 문서에는 인증서 파일을 게이트웨이에 업로드하고 게이트웨이 디바이스가 AWS IoT Core for LoRaWAN과 통신하도록 구성하는 프로세스가 설명되어 있어야 합니다.

AWS IoT Core for LoRaWAN에서 사용할 수 있는 게이트웨이

LoraWAN 게이트웨이를 구성하는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN 워크숍의 게이트웨이 디바이스 구성 단원을 참조하세요. 여기에서 AWS IoT Core for LoRaWAN에서 사용할 수 있는 게이트웨이를 연결하기 위한 지침에 대한 정보를 찾을 수 있습니다.

CUPS 프로토콜을 지원하는 게이트웨이

다음 지침은 CUPS 프로토콜을 지원하는 게이트웨이를 연결하는 방법을 보여줍니다.

1. 게이트웨이를 추가할 때 얻은 다음 파일을 업로드합니다.
   - 게이트웨이 디바이스 인증서 및 프라이빗 키 파일.
   - CUPS 엔드포인트에 대한 신뢰 인증서 파일 `cups.trust`.
2. 이전에 얻은 CUPS 엔드포인트 URL을 지정합니다. 엔드포인트 형식은 *`prefix`*`.cups.lorawan.`*`region`*`.amazonaws.com:443`이(가) 될 것입니다.

이 정보를 얻는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 추가 단원을 참조하세요.

LNS 프로토콜을 지원하는 게이트웨이

다음 지침은 LNS 프로토콜을 지원하는 게이트웨이를 연결하는 방법을 보여 줍니다.

1. 게이트웨이를 추가할 때 얻은 다음 파일을 업로드합니다.
   - 게이트웨이 디바이스 인증서 및 프라이빗 키 파일.
   - LNS 엔드포인트에 대한 신뢰 인증서 파일 `lns.trust`.
2. 이전에 가져온 LNS 엔드포인트 URL을 지정합니다. 엔드포인트 형식은 https://*`prefix`*`.lns.lorawan.`*`region`*`.amazonaws.com:443`입니다.

이 정보를 얻는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 추가 단원을 참조하세요.

게이트웨이를 AWS IoT Core for LoRaWAN에 연결한 후 연결 상태를 확인하고 콘솔이나 API를 사용하여 마지막 업링크를 받은 시기에 대한 정보를 얻을 수 있습니다.

콘솔을 사용하여 게이트웨이 연결 상태 확인

콘솔을 사용하여 연결 상태를 확인하려면 AWS IoT 콘솔의 게이트웨이 페이지로 이동하여 추가한 게이트웨이를 선택합니다. 게이트웨이 세부 정보 페이지의 LoRaWAN 세부 정보 섹션에서 연결 상태와 마지막 업링크 수신 날짜 및 시간이 표시됩니다.

API를 사용하여 게이트웨이 연결 상태 확인

API를 사용해 연결 상태를 확인하려면 `GetWirelessGatewayStatistics` API를 사용합니다. 이 API에는 요청 본문이 없으며 게이트웨이가 연결되었는지 여부와 마지막 업링크가 수신된 시점을 보여주는 응답 본문만 포함합니다.

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

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

AWS IoT Core for LoRaWAN에 디바이스 온보딩

게이트웨이를 AWS IoT Core for LoRaWAN에 온보딩하고 연결 상태를 확인한 후 무선 디바이스를 온보딩할 수 있습니다. 게이트웨이를 온보딩하는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 온보딩 단원을 참조하세요.

LoRaWAN 디바이스는 LoRaWAN 프로토콜을 사용하여 데이터를 클라우드 호스팅 애플리케이션과 교환합니다. AWS IoT Core for LoRaWAN은 LoRa Alliance에 의해 표준화된 1.0.x 또는 1.1 LoRaWAN 사양을 준수하는 디바이스를 지원합니다.

LoRaWAN 디바이스는 일반적으로 하나 이상의 센서 및 액터를 포함합니다. 디바이스는 LoRaWAN 게이트웨이를 통해 업링크 원격 분석 데이터를 AWS IoT Core for LoRaWAN에 전송합니다. 클라우드 호스팅 애플리케이션은 LoRaWAN 게이트웨이를 통해 LoRaWAN 디바이스로 다운링크 명령을 전송하여 센서를 제어할 수 있습니다.

무선 디바이스를 온보딩하기 전에

무선 디바이스를 AWS IoT Core for LoRaWAN에 온보딩하는 경우 사전에 다음 정보를 준비해야 합니다.

- LoRaWAN 사양 및 무선 디바이스 구성

  각 디바이스 고유의 구성 파라미터를 미리 입력할 준비가 되어 있으므로 콘솔에 데이터를 보다 원활하게 입력할 수 있습니다. 입력해야 하는 특정 파라미터는 디바이스에서 사용하는 LoRaWAN 사양에 따라 다릅니다. 사양 및 구성 파라미터의 전체 목록은 각 디바이스의 설명서를 참조하세요.

- 디바이스 이름 및 설명(선택 사항)

  이러한 선택적 필드의 정보는 무선 시스템의 요소를 구성하고 설명하는 방법에 따라 제공됩니다. 리소스 이름 지정 및 설명에 대한 자세한 내용은 AWS IoT 무선 리소스 설명 단원을 참조하세요.

- 디바이스 및 서비스 프로파일

  많은 디바이스에서 공유하고 AWS IoT Core for LoRaWAN에 디바이스 및 서비스 프로파일로서 저장될 수 있는 일부 무선 디바이스 구성 파라미터를 준비합니다. 구성 파라미터는 디바이스 설명서 또는 디바이스 자체에서 찾을 수 있습니다. 디바이스를 추가하기 전에 디바이스의 구성 파라미터와 일치하는 디바이스 프로파일을 식별하거나 필요한 경우 만들어야 합니다. 자세한 내용은 AWS IoT Core for LoRaWAN에 프로파일 추가 단원을 참조하십시오.

- AWS IoT Core for LoRaWAN 대상

  각 디바이스는 AWS IoT 및 기타 서비스에 보낼 메시지를 처리할 대상에 할당되어야 합니다. 디바이스 메시지를 처리하고 보내는 AWS IoT 규칙은 디바이스의 메시지 형식에 따라 다릅니다. 디바이스에서 메시지를 처리하고 올바른 서비스로 보내려면 디바이스의 메시지와 함께 사용할 대상을 식별하고 디바이스에 할당합니다.

AWS IoT Core for LoRaWAN에 무선 디바이스를 온보딩하려면

- AWS IoT Core for LoRaWAN에 무선 디바이스 추가
- AWS IoT Core for LoRaWAN에 프로파일 추가
- AWS IoT Core for LoRaWAN에 대상 추가
- LoRaWAN 디바이스 메시지를 처리하는 규칙 만들기
- LoRaWAN 디바이스를 연결하고 연결 상태를 확인합니다.

AWS IoT Core for LoRaWAN에 무선 디바이스 추가

무선 디바이스를 처음으로 추가하는 경우 콘솔을 사용하는 것이 좋습니다. AWS IoT 콘솔의 AWS IoT Core for LoRaWAN 소개(Intro) 페이지로 이동하여 시작하기(Get started)를 선택한 다음 디바이스 추가(Add device)를 선택합니다. 디바이스를 이미 추가한 경우 디바이스 보기를 선택하여 추가한 게이트웨이를 봅니다. 디바이스를 더 추가하려면 디바이스 추가를 선택합니다.

또는 AWS IoT 콘솔의 디바이스 페이지에서 무선 디바이스를 추가할 수도 있습니다.

콘솔을 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가

활성화 방법과 LoRaWAN 버전을 기반으로 무선 디바이스 사양을 선택합니다. 선택한 데이터는 사용자를 위해 AWS가 소유하고 관리하는 키로 암호화됩니다.

OTAA 및 ABP 활성화 모드

LoRaWAN 디바이스에서 업링크 데이터를 전송하기 전에 활성화 또는 조인 프로시저라는 프로세스를 완료해야 합니다. 디바이스를 활성화하려면 OTAA(무선 업데이트 활성화) 또는 ABP(개인 설정으로 활성화)를 사용할 수 있습니다.

ABP는 조인 프로시저를 필요로하지 않으며 정적 키를 사용합니다. OTAA를 사용하면 LoRaWAN 디바이스가 조인 요청을 전송하고 네트워크 서버에서 요청을 허용할 수 있습니다. 각 활성화에 대해 새 세션 키가 생성되므로 OTAA를 사용하여 디바이스를 활성화하는 것이 좋습니다.

LoRaWAN 버전

OTAA를 사용하면 LoRaWAN 디바이스와 클라우드 호스팅 애플리케이션이 루트 키를 공유합니다. 이러한 루트 키는 버전 v1.0.x 또는 v1.1을 사용하는지 여부에 따라 다릅니다. v1.0.x에는 루트 키(AppKey(애플리케이션 키)가 하나만 있는 반면 v1.1에는 두 개의 루트 키(AppKey(애플리케이션 키) 및NwkKey(네트워크 키))가 있습니다. 세션 키는 각 활성화에 대한 루트 키를 기반으로 파생됩니다. NwkKey 및 AppKey는 모두 무선 공급 업체에서 제공한 32자리 16진수 값입니다.

무선 디바이스 EUI

무선 디바이스 사양을 선택하면 콘솔에 표시되는 무선 디바이스의 EUI(확장 고유 식별자) 파라미터가 표시됩니다. 이 정보는 디바이스 또는 무선 공급 업체에 대한 설명서에서 찾을 수 있습니다.

- DevEUI: 디바이스에 고유하며 디바이스 레이블 또는 해당 설명서에 있는 16자리 16진수 값입니다.
- AppEui: 조인 서버에 고유하며 디바이스 설명서에서 찾을 수 있는 16자리 16진수 값입니다. LoRaWAN v1.1 버전에서 AppEui가 JoinEui로 호출됩니다.

고유 식별자, 세션 키 및 루트 키에 대한 자세한 내용은 LoRa Alliance 설명서를 참조하세요.

API를 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가

API를 사용하여 무선 디바이스를 추가하는 경우 무선 디바이스를 만들기 전에 먼저 디바이스 프로파일과 서비스 프로파일을 만들어야 합니다. 무선 디바이스를 만들 때 디바이스 프로파일과 서비스 프로파일 ID를 사용합니다. API를 사용해 프로파일을 생성하는 방법에 대한 자세한 내용은 API를 사용하여 디바이스 프로파일 추가 단원을 참조하세요.

다음 목록에서는 서비스 프로파일 추가, 업데이트 또는 삭제와 관련된 작업을 수행하는 API 작업에 대해 설명합니다.

서비스 프로파일에 대한 AWS IoT 무선 API 작업

- CreateWirelessDevice
- GetWirelessDevice
- ListWirelessDevices
- UpdateWirelessDevice
- DeleteWirelessDevice

AWS IoT Core for LoRaWAN 리소스 생성 및 관리에 사용할 수 있는 작업 및 데이터 유형의 전체 목록은 AWS IoT 무선 API 참조를 참조하세요.

AWS CLI를 사용하여 무선 디바이스를 생성하는 방법

AWS CLI를 사용하여 create-wireless-device 명령으로 무선 디바이스를 만들 수 있습니다. 다음 예제에서는 input.json 파일을 사용하여 파라미터를 입력하여 무선 디바이스를 만듭니다.

> **Note**
>
> 여기에 표시된 CLI 명령에 해당하는 AWS API의 메서드를 사용하여 API로 이 절차를 수행할 수도 있습니다.

input.json 내용

```
{
    "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
}
```

이 파일을 `create-wireless-device` 명령에 대한 입력으로 제공할 수 있습니다.

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

사용할 수 있는 CLI에 대한 자세한 내용은 AWS CLI 참조를 참조하세요.

AWS IoT Core for LoRaWAN에 프로파일 추가

디바이스 및 서비스 프로파일을 정의하여 일반적인 디바이스 구성을 설명할 수 있습니다. 이러한 프로파일은 디바이스를 더 쉽게 추가할 수 있도록 디바이스에서 공유되는 구성 파라미터를 설명합니다. AWS IoT Core for LoRaWAN은 디바이스 프로파일 및 서비스 프로파일을 지원합니다.

이러한 프로파일에 입력할 구성 파라미터 및 값은 디바이스 제조업체에서 제공합니다.

디바이스 프로파일 추가

디바이스 프로파일은 네트워크 서버가 LoRaWAN 무선 액세스 서비스를 설정하는 데 사용하는 디바이스 기능과 부팅 파라미터를 정의합니다. 여기에는 LoRa 주파수 대역, LoRa 리전 파라미터 버전 및 디바이스의 MAC 버전과 같은 파라미터 선택이 포함됩니다. 다른 주파수 대역에 대한 자세한 내용은 게이트웨이 및 디바이스 연결을 위한 LoRa 주파수 대역 선택 고려 단원을 참조하세요.

콘솔을 사용하여 디바이스 프로파일 추가

콘솔을 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가에 설명된 대로 콘솔을 사용하여 무선 디바이스를 추가하는 경우, 무선 디바이스 사양을 추가한 후 디바이스 프로파일을 추가할 수

있습니다. 또는 AWS IoT 콘솔의 프로파일 페이지에서 LoRaWAN 탭을 통해 무선 디바이스를 추가할 수도 있습니다.

기본 디바이스 프로파일 중에서 선택하거나 새 디바이스 프로파일을 만들 수 있습니다. 기본 디바이스 프로파일을 사용하는 것이 좋습니다. 애플리케이션에서 디바이스 프로파일을 만들어야 하는 경우 디바이스 프로파일 이름을 제공하고 디바이스 및 게이트웨이에 사용 중인 주파수 대역(RFRegion)를 선택하고 디바이스 설명서에 달리 지정되지 않는 한 다른 설정을 기본값으로 유지합니다.

API를 사용하여 디바이스 프로파일 추가

API를 사용하여 무선 디바이스를 추가하는 경우 무선 디바이스를 만들기 전에 먼저 디바이스 프로파일을 만들어야 합니다.

다음 목록에서는 서비스 프로파일 추가, 업데이트 또는 삭제와 관련된 작업을 수행하는 API 작업에 대해 설명합니다.

서비스 프로파일에 대한 AWS IoT 무선 API 작업

- CreateDeviceProfile
- GetDeviceProfile
- ListDeviceProfiles
- UpdateDeviceProfile
- DeleteDeviceProfile

AWS IoT Core for LoRaWAN 리소스 생성 및 관리에 사용할 수 있는 작업 및 데이터 유형의 전체 목록은 AWS IoT 무선 API 참조를 참조하세요.

AWS CLI를 사용하여 디바이스 프로파일을 생성하는 방법

AWS CLI를 사용하여 create-device-profile 명령으로 무선 프로파일을 만들 수 있습니다. 다음 예제에서는 디바이스 프로파일을 만듭니다.

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

이 명령을 실행하면 무선 디바이스를 만들 때 사용할 수 있는 ID로 디바이스 프로파일을 자동으로 생성합니다. 이제 다음 API를 사용하여 서비스 프로파일을 만든 다음 디바이스 및 서비스 프로파일을 사용하여 무선 디바이스를 만들 수 있습니다.

```
{
```

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

사용할 수 있는 CLI에 대한 자세한 내용은 AWS CLI 참조를 참조하세요.

서비스 프로파일 추가

서비스 프로파일은 디바이스가 애플리케이션 서버와 통신하는 데 필요한 통신 파라미터를 설명합니다.

콘솔을 사용하여 서비스 프로파일 추가

콘솔을 사용하여 콘솔을 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가에서 설명된 대로 무선 디바이스를 추가하는 경우 디바이스 프로파일을 추가한 후 서비스 프로파일을 추가할 수 있습니다. 또는 AWS IoT 콘솔의 프로파일 페이지에서 LoRaWAN 탭을 통해 무선 디바이스를 추가할 수도 있습니다.

각 페이로드에 대한 추가 게이트웨이 메타데이터(예: 데이터 전송을 위한 RSSI 및 SNR)를 수신하도록 AddGWMetaData 설정을 활성화된 상태로 유지하는 것이 좋습니다.

API를 사용하여 서비스 프로파일 추가

API를 사용하여 무선 디바이스를 추가하는 경우 무선 디바이스를 만들기 전에 먼저 서비스 프로파일을 만들어야 합니다.

다음 목록에서는 서비스 프로파일 추가, 업데이트 또는 삭제와 관련된 작업을 수행하는 API 작업에 대해 설명합니다.

서비스 프로파일에 대한 AWS IoT 무선 API 작업

- CreateServiceProfile
- GetServiceProfile
- ListServiceProfiles
- UpdateServiceProfile
- DeleteServiceProfile

AWS IoT Core for LoRaWAN 리소스 생성 및 관리에 사용할 수 있는 작업 및 데이터 유형의 전체 목록은 AWS IoT 무선 API 참조를 참조하세요.

AWS CLI를 사용하여 서비스 프로파일을 만드는 방법

AWS CLI를 사용하여 create-service-profile 명령으로 서비스를 생성할 수 있습니다. 다음 예제에서는 서비스 프로파일을 만듭니다.

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

이 명령을 실행하면 무선 디바이스를 만들 때 사용할 수 있는 ID로 서비스 프로파일이 자동으로 만들어집니다. 이제 디바이스 및 서비스 프로파일을 사용하여 무선 디바이스를 만들 수 있습니다.

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

AWS IoT Core for LoRaWAN에 대상 추가

AWS IoT Core for LoRaWAN 대상은 AWS 서비스에서 사용할 디바이스의 데이터를 처리하는 AWS IoT 규칙을 설명합니다.

대부분의 LoRaWAN 디바이스는 데이터를 AWS 서비스에서 사용되는 형식으로 AWS IoT Core for LoRaWAN에 전송하지 않기 때문에 AWS IoT 규칙이 이를 먼저 처리해야 합니다. AWS IoT 규칙은 디바이스의 데이터를 해석하는 SQL 문과, SQL 문의 결과를 사용할 서비스로 보내는 주제 규칙 작업을 포함합니다.

대상을 처음으로 추가하는 경우 콘솔을 사용하는 것이 좋습니다.

콘솔을 사용하여 대상 추가

콘솔을 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가에 설명된 대로 콘솔을 사용하여 무선 디바이스를 추가하는 경우, 앞서 설명한 대로 무선 디바이스 사양 및 프로파일을 AWS IoT Core for LoRaWAN에 이미 추가했다면 다음 단계로 진행하여 대상을 추가할 수 있습니다.

또는 AWS IoT 콘솔의 대상(Destinations) 페이지에서 AWS IoT Core for LoRaWAN 대상을 추가할 수도 있습니다.

디바이스의 데이터를 처리하려면, AWS IoT Core for LoRaWAN 대상을 생성할 때 다음 필드를 지정한 다음 대상 추가를 선택합니다.

- 대상 세부 사항

  대상 이름을 입력하고 대상의 설명(선택 사항)을 입력합니다.

- 규칙 이름

  디바이스에서 보낸 메시지를 평가하고 디바이스의 데이터를 처리하도록 구성된 AWS IoT 규칙입니다. 규칙 이름이 대상에 매핑됩니다. 대상은 수신하는 메시지를 처리하기 위한 규칙이 필요합니다. AWS IoT 규칙을 호출하거나 AWS IoT 메시지 브로커에 게시하여 메시지를 처리하도록 선택할 수 있습니다.

  - 규칙 이름 입력(Enter a rule name)을 선택하는 경우 이름을 입력한 다음 복사(Copy)를 선택하여 규칙 이름(AWS IoT 규칙을 생성할 때 입력)을 복사합니다. 규칙 생성을 선택하여 지금 규칙을 생성하거나, AWS IoT 콘솔의 규칙 허브로 이동하여 해당 이름으로 규칙을 생성할 수 있습니다.

    규칙을 입력하고 고급(Advanced) 설정을 사용하여 주제 이름을 지정할 수도 있습니다. 주제 이름은 규칙 호출 중에 제공되며 규칙 내에서 `topic` 표현식을 사용하여 액세스됩니다. AWS IoT 규칙에 대한 자세한 내용은 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html 섹션을 참조하세요.

  - AWS IoT 메시지 브로커에 게시를 선택하는 경우 주제 이름을 입력합니다. 그런 다음 MQTT 주제 이름을 복사하면 여러 구독자가 이 주제를 구독하여 해당 주제에 게시된 메시지를 받을 수 있습니다. 자세한 내용은 https://docs.aws.amazon.com/iot/latest/developerguide/topics.html 단원을 참조하십시오.

  대상의 AWS IoT 규칙에 대한 자세한 내용은 LoRaWAN 디바이스 메시지를 처리하는 규칙 만들기 단원을 참조하세요.

- 역할 이름

  규칙 이름(Rule name)에서 명명된 규칙에 액세스할 수 있는 디바이스의 데이터 권한을 부여하는 IAM 역할입니다. 콘솔에서 새 서비스 역할을 생성하거나 기존 서비스 역할을 선택합니다. 새 서비스 역할을 생성하는 경우 역할 이름(예: **`IoTWirelessDestinationRole`**)을 입력하거나 AWS IoT Core for LoRaWAN이 새 역할 이름을 생성할 수 있도록 비워 둘 수 있습니다. 그러면 AWS IoT Core for LoRaWAN이 사용자를 대신하여 적절한 권한이 있는 IAM 역할을 자동으로 생성합니다.

  IAM 역할에 대한 자세한 내용은 IAM 역할 사용을 참조하세요.

API를 사용하여 대상 추가

대신 CLI를 사용하여 대상을 추가하려면 이미 대상에 대한 규칙 및 IAM 역할을 생성했어야 합니다. 역할에서 대상에 필요한 세부 정보에 대한 자세한 내용은 대상에 대한 IAM 역할 생성 섹션을 참조하세요.

다음 목록에는 대상 추가, 업데이트 또는 삭제와 관련된 작업을 수행하는 API 작업이 포함되어 있습니다.

대상에 대한 AWS IoT 무선 API 작업

- CreateDestination
- GetDestination
- ListDestinations
- UpdateDestination
- DeleteDestination

AWS IoT Core for LoRaWAN 리소스 생성 및 관리에 사용할 수 있는 작업 및 데이터 유형의 전체 목록은 AWS IoT 무선 API 참조를 참조하세요.

AWS CLI를 사용하여 대상을 추가하는 방법

AWS CLI를 사용하여 create-destination 명령으로 대상을 추가할 수 있습니다. 다음 예에서는 `RuleName`을 `expression-type` 파라미터 값으로 사용하여 규칙 이름을 입력하여 대상을 생성하는 방법을 보여줍니다. 메시지 브로커에 게시하거나 구독하기 위한 주제 이름을 지정하려면 `expression-type` 파라미터의 값을 `MqttTopic`으로 변경합니다.

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

이 명령을 실행하면 지정된 대상 이름, 규칙 이름 및 역할 이름을 가진 대상이 만들어집니다. 대상의 규칙 및 역할 이름에 대한 자세한 내용은 LoRaWAN 디바이스 메시지를 처리하는 규칙 만들기 및 대상에 대한 IAM 역할 생성 단원을 참조하세요.

사용할 수 있는 CLI에 대한 자세한 내용은 AWS CLI참조를 참조하세요.

대상에 대한 IAM 역할 생성

AWS IoT Core for LoRaWAN 대상은 AWS IoT 규칙으로 데이터를 전송하는 데 필요한 권한을 AWS IoT Core for LoRaWAN에 부여하는 IAM 역할이 필요합니다. 이러한 역할이 아직 정의되지 않은 경우 역할 목록에 표시되도록 정의해야 합니다.

콘솔을 사용하여 대상을 추가하면 이 주제의 앞부분에서 설명한 대로 AWS IoT Core for LoRaWAN에서 자동으로 IAM 역할을 생성합니다. API 또는 CLI를 사용하여 대상을 추가하는 경우 대상에 대한 IAM 역할을 생성해야 합니다.

AWS IoT Core for LoRaWAN 대상 역할에 대한 IAM 정책을 생성하는 방법

1. IAM 콘솔의 정책 허브를 엽니다.
2. 정책 생성을 선택한 후 JSON 탭을 선택합니다.
3. 편집기에서 편집기의 모든 내용을 삭제하고 이 정책 문서를 붙여 넣습니다.

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

4. 정책 검토를 선택하고 이름에 이 정책의 이름을 입력합니다. 다음 절차에서 이 이름을 사용해야 합니다.

   또한 이 정책을 설명에 기술할 수도 있습니다(원하는 경우).
5. 정책 생성을 선택합니다.

AWS IoT Core for LoRaWAN 대상에 대한 IAM 역할을 생성하는 방법

1. IAM 콘솔의 역할 허브를 열고 역할 생성을 선택합니다.
2. 신뢰할 수 있는 엔터티 유형 선택(Select type of trusted entity)에서 다른 AWS 계정을 선택합니다.
3. 계정 ID에 AWS 계정 ID를 입력한 후 다음: 권한을 선택합니다.
4. 검색 상자에 이전 절차에서 생성한 IAM 정책의 이름을 입력합니다.
5. 검색 결과에서 이전 절차에서 생성한 IAM 정책의 이름을 선택합니다.
6. Next: Tags(다음: 태그)를 선택한 후 Next: Review(다음: 검토)를 선택합니다.

7. 역할 이름에 이 역할의 이름을 입력한 후 역할 생성을 선택합니다.
8. 확인 메시지에서 생성한 역할의 이름을 선택하여 새 역할을 편집합니다.
9. 요약(Summary)에서 신뢰 관계(Trust relationships) 탭을 선택한 다음 신뢰 관계 편집(Edit trust relationship)을 선택합니다.
10. 정책 문서에서 `Principal` 속성을 다음 예시처럼 변경합니다.

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

`Principal` 속성을 변경한 후 전체 정책 문서가 다음 예시와 같은 형식이어야 합니다.

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

11. 변경 사항을 저장하고 종료하려면 신뢰 정책 업데이트(Update Trust Policy)를 선택합니다.

이 역할이 정의되어 있으면 AWS IoT Core for LoRaWAN 대상을 구성할 때 역할 목록에서 해당 역할을 찾을 수 있습니다.

LoRaWAN 디바이스 메시지를 처리하는 규칙 만들기

AWS IoT 규칙은 디바이스 메시지를 다른 서비스로 보냅니다. AWS IoT 규칙은 또한 LoRaWAN 디바이스에서 받은 이진 메시지를 처리하여 다른 서비스에서 보다 쉽게 사용할 수 있도록 다른 형식으로 메시지를 변환할 수 있습니다.

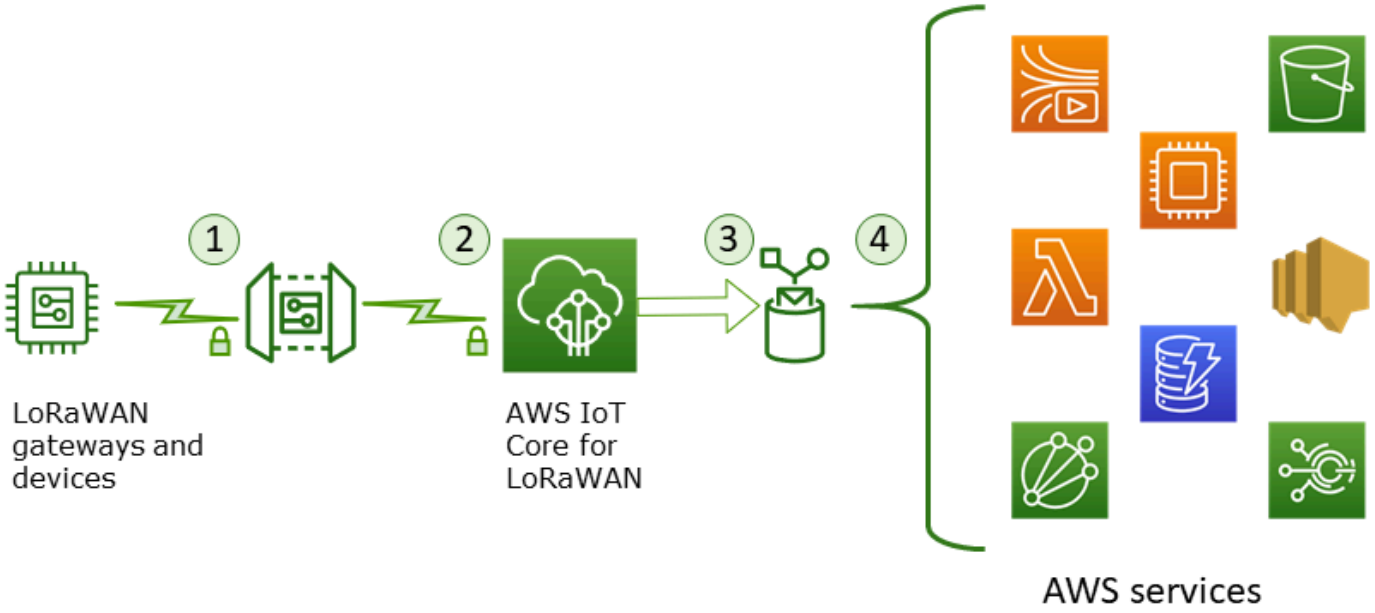
AWS IoT Core for LoRaWAN 대상은 무선 디바이스를 디바이스의 메시지 데이터를 처리하여 다른 서비스로 보내는 규칙과 연결합니다. 이 규칙은 AWS IoT Core for LoRaWAN이 수신하는 즉시 디바이스

의 데이터에 적용됩니다. AWS IoT Core for LoRaWAN 대상은 메시지가 동일한 데이터 형식을 가지며 데이터를 동일한 서비스로 보내는 모든 디바이스에서 공유할 수 있습니다.

AWS IoT 규칙이 디바이스 메시지를 처리하는 방식

AWS IoT 규칙은 데이터를 수신할 서비스, 디바이스의 메시지 데이터 형식 및 서비스에 필요한 데이터 형식에 따라 다릅니다. 일반적으로 이 규칙은 AWS Lambda 함수를 호출하여 디바이스의 메시지 데이터를 서비스에 필요한 형식으로 변환한 다음 결과를 서비스로 보냅니다.

다음 그림에서는 메시지 데이터가 무선 디바이스에서 AWS 서비스로 이동할 때 보호 및 처리되는 방식을 보여줍니다.



1. LoRaWAN 무선 디바이스는 전송하기 전에 AES128 CTR 모드를 사용하여 이진 메시지를 암호화합니다.
2. AWS IoT Core for LoRaWAN은 이진 메시지를 복호화하고 복호화된 이진 메시지 페이로드를 base64 문자열로 인코딩합니다.
3. 이에 따라 base64로 인코딩된 메시지는 디바이스에 할당된 대상에 설명되어 있는 AWS IoT 규칙에 JSON 문서로 형식이 지정되지 않은 메시지 페이로드로 전송됩니다.
4. AWS IoT 규칙은 규칙 구성에 설명된 서비스에 메시지 데이터를 보냅니다.

무선 디바이스에서 받은 암호화된 이진 페이로드는 AWS IoT Core for LoRaWAN에 의해 변경 또는 해석되지 않습니다. 보호화된 이진 메시지 페이로드는 base64 문자열로만 인코딩됩니다. 서비스가 이진 메시지 페이로드의 데이터 요소에 액세스하려면 규칙에 의해 호출된 함수에서 페이로드로부터 데이터

요소를 구문 분석해야 합니다. base64로 인코딩된 메시지 페이로드는 ASCII 문자열이므로 나중에 구문 분석할 수 있습니다.

LoRaWAN 디바이스 규칙 생성

AWS IoT Core for LoRaWAN은 메시지 브로커를 사용할 필요없이 AWS IoT 규칙을 사용하여 디바이스 메시지를 다른 AWS 서비스에 직접 안전하게 전송합니다. 수집 경로에서 메시지 브로커를 제거하면 비용이 절감되고 데이터 흐름이 최적화됩니다.

AWS IoT Core for LoRaWAN 규칙을 통해 디바이스 메시지를 다른 AWS 서비스에 전송하려면 AWS IoT Core for LoRaWAN 대상과, 해당 대상에 할당된 AWS IoT 규칙이 필요합니다. AWS IoT 규칙은 SQL 쿼리 문과 하나 이상의 규칙 작업을 포함해야 합니다.

일반적으로 AWS IoT 규칙 쿼리 문은 다음으로 구성됩니다.

- 메시지 페이로드에서 데이터를 선택하고 형식을 지정하는 SQL SELECT 절
- 사용할 메시지를 식별하는 항목 필터(규칙 쿼리 문의 FROM 객체)
- 작동할 조건을 지정하는 선택적 조건문(SQL WHERE 절)

다음은 규칙 쿼리 문의 예입니다.

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

LoRaWAN 디바이스의 페이로드를 처리하는 AWS IoT 규칙을 빌드할 때 규칙 쿼리 객체의 일부로 FROM 절을 지정할 필요가 없습니다. 규칙 쿼리 문에는 SQL SELECT 절이 있어야 하며 선택적으로 WHERE 절을 가질 수 있습니다. 쿼리 문에서 FROM 절을 사용하는 경우 해당 절은 무시됩니다.

다음은 LoraWAN 디바이스에서 페이로드를 처리할 수 있는 규칙 쿼리 문의 예입니다.

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

이 예제에서 `PayloadData`는 LoRaWAN 디바이스에서 전송한 base64로 인코딩된 이진 페이로드입니다.

다음은 들어오는 페이로드의 이진 디코딩을 수행하여 JSON 등의 다른 형식으로 변환할 수 있는 규칙 쿼리 문 예제입니다.

```
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 및 WHERE 절을 사용하는 방법에 대한 자세한 내용은 https://docs.aws.amazon.com/iot/latest/developerguide/iot-sql-reference.html 단원을 참조하세요.

AWS IoT 규칙과, 이 규칙을 생성 및 사용하는 방법에 대한 자세한 내용은 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html 및 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules-tutorial.html 단원을 참조하세요.

AWS IoT Core for LoRaWAN 대상의 생성 및 사용에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 대상 추가 단원을 참조하세요.

규칙에서 이진 메시지 페이로드를 사용하는 방법에 대한 자세한 내용은 https://docs.aws.amazon.com/iot/latest/developerguide/binary-payloads.html 단원을 참조하세요.

메시지 페이로드를 보호하는 데 사용되는 데이터 보안 및 암호화에 대한 자세한 내용은 AWS IoT Wireless의 데이터 보호 단원을 참조하세요.

IoT 규칙에 대한 이진 디코딩 및 구현 예제를 보여 주는 참조 아키텍처는 GitHub의 AWS IoT Core for LoRaWAN 솔루션 샘플을 참조하세요.

LoRaWAN 디바이스를 연결하고 연결 상태를 확인합니다.

디바이스 연결 상태를 확인하려면 디바이스를 이미 추가하고 AWS IoT Core for LoRaWAN에 연결되어 있어야 합니다. 디바이스 추가 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 무선 디바이스 추가 단원을 참조하세요.

디바이스를 추가한 후 LoRaWAN 디바이스에서 업링크 메시지 전송을 시작하는 방법을 알아보려면 디바이스의 사용 설명서를 참조하세요.

콘솔을 사용하여 디바이스 연결 상태 확인

콘솔을 사용하여 연결 상태를 확인하려면 AWS IoT 콘솔의 디바이스 페이지로 이동하여 추가한 디바이스를 선택합니다. 무선 디바이스 세부 정보 페이지의 세부 정보 섹션에서 마지막 업링크가 수신된 날짜와 시간이 표시됩니다.

API를 사용하여 디바이스 연결 상태 확인

API를 사용해 연결 상태를 확인하려면 `GetWirelessDeviceStatistics` API를 사용합니다. 이 API는 요청 본문이 없으며 마지막 업링크가 수신된 시점을 표시하는 응답 본문만 포함합니다.

```
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"
}
```

다음 단계

이제 디바이스를 연결하고 연결 상태를 확인했으므로 AWS IoT 콘솔의 테스트 페이지에 있는 MQTT 테스트 클라이언트를 사용하여 디바이스에서 수신한 업링크 메타데이터의 형식을 확인할 수 있습니다. 자세한 내용은 LoRaWAN 디바이스에서 전송된 업링크 메시지 형식 보기 단원을 참조하십시오.

AWS IoT Core for LoRaWAN으로 무선 리소스 위치 구성

이 기능을 사용하기 전에 LoRaWAN 디바이스의 위치 정보를 확인하기 위해 선택한 서드 파티 공급자는 IGS(International GNSS Service), NASA를 통한 EarthData 또는 기타 서드 파티가 제공하거나 유지 관리하는 데이터 피드 및 데이터 세트를 사용한다는 점에 유의하세요. 이러한 데이터 피드 및 데이터 세트는 서드 파티 콘텐츠(고객 계약에 정의됨)이며 있는 그대로 제공됩니다. 자세한 내용은 AWS서비스 약관을 참조하세요.

AWS IoT Core for LoRaWAN을 사용하여 정적 위치 데이터를 지정하거나 위치 확인을 활성화하여 서드 파티 솔버를 통해 디바이스의 위치를 실시간으로 식별할 수 있습니다. LoRaWAN 디바이스나 게이트웨이 또는 둘 다에 대한 위치 정보를 추가하거나 업데이트할 수 있습니다.

AWS IoT Core for LoRaWAN에 디바이스 또는 게이트웨이를 추가하거나 디바이스 또는 게이트웨이의 구성 세부 정보를 편집할 때 위치 정보를 지정합니다. 위치 정보는 GeoJSON 페이로드로 지정됩니다. GeoJSON 형식은 지리적 데이터 구조를 인코딩하는 데 사용되는 형식입니다. 페이로드에는 세계 측지 시스템 좌표계(WGS84)를 기반으로 하는 디바이스 위치의 위도 및 경도 좌표가 포함됩니다.

솔버가 리소스의 위치를 계산한 후 Amazon Location Service가 있는 경우, 리소스 위치가 표시되는 Amazon Location 맵을 활성화할 수 있습니다. 위치 데이터를 사용하여 다음을 수행할 수 있습니다.

- 위치 확인을 활성화하여 LoRaWAN 디바이스의 위치를 식별하고 가져옵니다.
- 게이트웨이와 디바이스의 위치를 추적하고 모니터링합니다.
- 위치 데이터에 대한 업데이트를 처리하고 이를 다른 AWS 서비스로 라우팅하는 AWS IoT 규칙을 정의합니다. 규칙 작업 목록은 AWS IoT 개발자 안내서의 AWS IoT 규칙 작업을 참조하세요.
- 위치 데이터와 Amazon SNS를 사용하여, 비정상적인 활동이 발생할 경우 알림을 생성하고 디바이스로 알림을 수신할 수 있습니다.

LoRaWAN 디바이스의 위치 확인이 작동하는 방식

위치 확인을 활성화하여 서드 파티 Wi-Fi 및 GNSS 솔버를 통해 디바이스의 위치를 식별할 수 있습니다. 이 정보를 사용하여 디바이스를 추적하고 모니터링할 수 있습니다. 다음 단계에서는 위치 확인을 활성화하고 LoRaWAN 디바이스의 위치 정보를 보는 방법을 보여줍니다.

> **Note**
>
> 서드 파티 솔버는 LoRa Edge 칩이 있는 LoRaWAN 디바이스에서만 사용할 수 있습니다. LoRaWAN 게이트웨이와 함께 사용할 수 없습니다. 게이트웨이의 경우, 여전히 정적 위치 정보를 지정하고 Amazon Location 맵에서 위치를 식별할 수 있습니다.

1. 디바이스 추가

   위치 확인을 활성화하려면 먼저 AWS IoT Core for LoRaWAN에 디바이스를 추가합니다. LoRaWAN 디바이스에는 LoRa Edge 칩셋이 있어야 합니다. 이 칩셋은 장거리 LoRa 송수신장치, 다중 위성군 GNSS 스캐너 및 패시브 Wi-Fi MAC 스캐너를 통합한 초저전력 플랫폼으로, 지리적 위치 애플리케이션을 대상으로 합니다.

2. 위치 확인 활성화

   디바이스의 실시간 위치를 확인하려면 위치 확인을 활성화합니다. LoRaWAN 디바이스에서 업링크 메시지를 보내면 메시지에 포함된 Wi-Fi 및 GNSS 스캔 데이터가 지리적 위치 프레임 포트를 사용하여 AWS IoT Core for LoRaWAN으로 전송됩니다.

3. 위치 정보 검색

   송수신장치의 스캔 결과를 바탕으로 계산된 예상 디바이스 위치를 솔버에서 검색합니다. 위치 정보가 Wi-Fi와 GNSS 스캔 결과를 모두 사용하여 계산된 경우 AWS IoT Core for LoRaWAN은 정확도가 더 높은 예상 위치를 선택합니다.

4. 위치 정보 보기

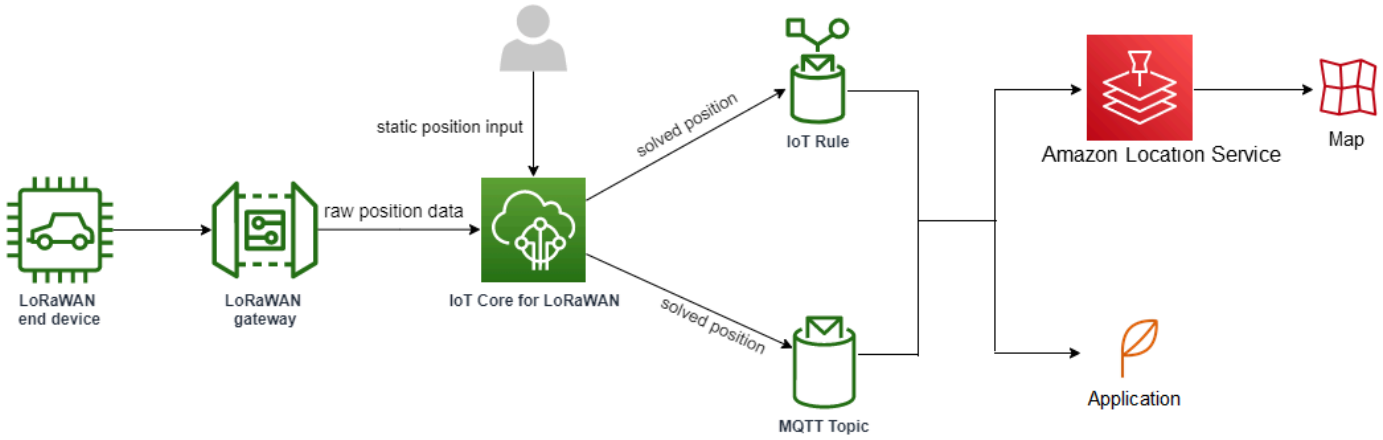
   솔버가 위치 정보를 계산한 후 솔버에서 계산한 위치와 사용자가 입력한 정적 위치 정보 간의 차이를 나타내는 정확도 정보가 표시됩니다. Amazon Location 맵에서 디바이스 위치를 확인할 수도 있습니다.

> **Note**
>
> LoRaWAN 게이트웨이에는 솔버를 사용할 수 없으므로 정확도 정보는 `0.0`과 같이 보고됩니다.

위치 확인 솔버에 사용되는 업링크 메시지 형식 및 주파수 포트에 대한 자세한 정보는 AWS IoT Core for LoRaWAN에서 규칙 엔진으로 전송되는 업링크 메시지 섹션을 참조하세요.

위치 확인 워크플로 개요

다음 다이어그램은 AWS IoT Core for LoRaWAN이 디바이스 및 게이트웨이의 위치 정보를 저장하고 업데이트하는 방법을 보여 줍니다.



1. 리소스의 정적 위치 지정

   위도 및 경도 좌표를 사용하여 디바이스 또는 게이트웨이의 정적 위치 정보를 GeoJSON 페이로드로 지정합니다. 고도 좌표를 지정할 수도 있습니다. 이러한 좌표는 WGS84 좌표계를 기반으로 합니다. 자세한 정보는 World Geodetic System(WGS84)을 참조하세요.
2. 디바이스에 대한 위치 확인 활성화

   LoRa Edge 칩이 장착된 LoRaWAN 디바이스를 사용하는 경우, 선택적으로 위치 확인을 활성화하여 디바이스 위치를 실시간으로 추적할 수 있습니다. 디바이스에서 업링크 메시지를 보내면 GNSS 및 Wi-Fi 스캔 데이터가 지리적 위치 프레임 포트를 사용하여 AWS IoT Core for LoRaWAN으로 전송됩니다. 그런 다음 솔버는 이 정보를 사용하여 디바이스 위치를 확인합니다.
3. 위치 데이터를 라우팅할 대상 추가

   디바이스 데이터를 처리하기 위한 IoT 규칙을 설명하는 대상을 추가하고 업데이트된 위치 정보를 AWS IoT Core for LoRaWAN으로 라우팅할 수 있습니다. Amazon Location 맵에서 리소스의 마지막으로 알려진 위치를 볼 수도 있습니다.

리소스 위치 구성

AWS Management Console, AWS IoT 무선 API 또는 AWS CLI를 사용하여 리소스 위치를 구성할 수 있습니다.

디바이스에 LoRa Edge 칩이 있는 경우, 위치 확인을 활성화하여 실시간 위치 정보를 계산할 수도 있습니다. 게이트웨이의 경우에도 정적 위치 좌표를 입력하고 Amazon Location을 사용하여 Amazon Location 맵에서 게이트웨이 위치를 추적할 수 있습니다.

주제

LoRaWAN 게이트웨이의 위치 구성

게이트웨이를 AWS IoT Core for LoRaWAN에 추가할 때 정적 위치 데이터를 지정할 수 있습니다. Amazon Location Service 맵을 활성화하면 위치 데이터가 Amazon Location 맵에 표시됩니다.

> **Note**
>
> 서드 파티 솔버는 LoRaWAN 게이트웨이와 함께 사용할 수 없습니다. 게이트웨이의 경우, 여전히 정적 위치 좌표를 지정할 수 있습니다. 게이트웨이의 경우처럼 솔버를 사용하여 위치를 계산하지 않는 경우, 정확도 정보는 `0.0`과 같이 보고됩니다.

AWS Management Console, AWS IoT 무선 API 또는 AWS CLI를 사용하여 게이트웨이 위치를 구성할 수 있습니다.

콘솔을 사용한 게이트웨이 위치 구성

AWS Management Console을 사용하여 게이트웨이 리소스의 위치를 구성하려면 먼저 콘솔에 로그인한 다음 AWS IoT 콘솔의 게이트웨이(Gateways) 허브 페이지로 이동합니다.

위치 정보 추가

게이트웨이에 대한 위치 구성 추가

1. 게이트웨이(Gateways) 허브 페이지에서 게이트웨이 추가(Add gateway)를 선택합니다.
2. 게이트웨이의 EUI, 주파수 대역(RFRegion) 및 추가 게이트웨이 세부 정보와 LoRaWAN 구성 정보를 입력합니다. 자세한 내용은 콘솔을 사용하여 게이트웨이 추가 단원을 참조하십시오.
3. Position information - Optional(위치 정보 - 선택 사항) 섹션에서 위도 및 경도 좌표와 선택적 고도 좌표를 사용하여 게이트웨이의 위치 정보를 입력합니다. 위치 정보는 WGS84 좌표계를 기반으로 합니다.

게이트웨이의 위치 보기

게이트웨이의 위치를 구성한 후 AWS IoT Core for LoRaWAN은 `iotwireless.map`이라고 불리는 Amazon Location 맵을 생성합니다. 이 맵은 위치(Position) 탭의 게이트웨이 세부 정보 페이지에서 확

인할 수 있습니다. 지정한 위치 좌표에 따라 게이트웨이의 위치가 맵에 마커로 표시됩니다. 확대 또는 축소하여 맵에서 게이트웨이의 위치를 명확하게 볼 수 있습니다. Position(위치) 탭에는 정확도 정보와 게이트웨이의 위치가 결정된 타임스탬프도 표시됩니다.

> **Note**
>
> Amazon Location Service 맵이 설치되어 있지 않은 경우, 맵에 액세스하고 게이트웨이 위치를 확인하려면 Amazon Location Service를 사용해야 한다는 메시지가 표시됩니다. Amazon Location Service 맵을 사용하면 AWS 계정에 추가 요금이 발생할 수 있습니다. 자세한 내용은 AWS IoT Core 요금을 참조하십시오.

`iotwireless.map` 맵은 `GetMapTile`과 같은 `Get` API 작업을 사용하여 액세스되는 지도 데이터의 소스 역할을 합니다. 맵과 함께 사용되는 `Get` API에 대한 자세한 정보는 Amazon Location Service API 참조를 참고하세요.

이 맵에 대한 추가 세부 정보를 보려면 Amazon Location Service 콘솔로 이동하여 맵(Maps)을 선택한 다음 iotwireless.map을 선택합니다. 자세한 정보는 Amazon Location Service 개발자 안내서의 맵을 참조하세요.

게이트웨이의 위치 구성 업데이트

게이트웨이의 위치 구성을 변경하려면 게이트웨이 세부 정보 페이지에서 편집(Edit)을 선택한 다음 위치 정보와 대상을 업데이트합니다.

> **Note**
>
> 과거 위치 데이터에 대한 정보는 제공되지 않습니다. 게이트웨이의 위치 좌표를 업데이트하면 이전에 보고된 위치 데이터를 덮어씁니다. 위치를 업데이트한 후 게이트웨이 세부 정보의 위치(Position) 탭에서 새 위치 정보를 볼 수 있습니다. 타임스탬프의 변화는 새 위치 정보가 마지막으로 알려진 게이트웨이 위치와 동기화되었다는 것을 나타냅니다.

API를 사용한 게이트웨이 위치 구성

AWS IoT 무선 API 또는 AWS CLI를 사용하여 위치 정보를 지정하고 게이트웨이 위치를 구성할 수 있습니다.

> **Important**
>
> API 작업 UpdatePosition, GetPosition, PutPositionConfiguration, GetPositionConfiguration 및 ListPositionConfigurations는 더 이상 지원되지 않습니다. 위치 정보를 업데이트하고 검색하기 위한 호출에는 GetResourcePosition 및 UpdateResourcePosition API 작업을 대신 사용해야 합니다.

위치 정보 추가

특정 무선 게이트웨이에 정적 위치 정보를 추가하려면 UpdateResourcePosition API 작업 또는 update-resource-position CLI 명령을 사용하여 좌표를 지정합니다. `WirelessGateway`를 `ResourceType`으로 지정하고, 업데이트할 무선 게이트웨이의 ID를 `ResourceIdentifier`로 지정하고, 위치 정보를 GeoJSON 페이로드로 지정합니다.

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

다음은 *`gatewayposition.json`* 파일의 콘텐츠를 보여줍니다.

gatewayposition.json의 내용

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

이 명령을 실행하면 출력을 생성하지 않습니다. 지정한 위치 정보를 보려면 `GetResourcePosition` API 작업을 사용합니다.

위치 정보 가져오기

특정 무선 게이트웨이에 대한 위치 정보를 가져오려면 GetResourcePosition API 또는 get-resource-position CLI 명령을 사용합니다. `WirelessGateway`를 `resourceType`으로 지정하고 무선 게이트웨이의 ID를 `resourceIdentifier`로 제공합니다.

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

이 명령을 실행하면 무선 게이트웨이의 위치 정보가 GeoJSON 페이로드로 표시됩니다. 위치 좌표, 위치 정보 유형 및 추가 속성(예: 게이트웨이의 마지막으로 알려진 위치에 해당하는 타임스탬프)에 대한 정보가 표시됩니다.

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

LoRaWAN 디바이스의 위치 구성

디바이스를 AWS IoT Core for LoRaWAN에 추가할 때 정적 위치 정보를 지정하고, 선택적으로 위치 확인을 활성화하고, 대상을 지정할 수 있습니다. 대상은 디바이스의 위치 정보를 처리하고 업데이트된 위치를 Amazon Location Service에 라우팅하는 IoT 규칙을 설명합니다. 디바이스 위치를 구성하면 위치 데이터가 Amazon Location 맵에 정확도 정보 및 지정한 대상과 함께 표시됩니다.

AWS Management Console, AWS IoT 무선 API 또는 AWS CLI를 사용하여 디바이스 위치를 구성할 수 있습니다.

업링크 메시지의 프레임 포트 및 형식

위치 확인을 활성화하는 경우 디바이스의 Wi-Fi 및 GNSS 스캔 데이터를 AWS IoT Core for LoRaWAN에 전달하기 위한 지리적 위치 프레임 포트를 지정해야 합니다. 위치 정보는 이 프레임 포트를 사용하여 AWS IoT Core for LoRaWAN으로 전달됩니다.

LoRaWAN 사양은 서로 다른 유형의 메시지를 구별하기 위해 데이터 전송 필드(FRMPayload)와 포트 필드(FPort)를 제공합니다. 위치 정보를 전달하기 위해 프레임 포트에 1부터 223 사이의 값을 지정할 수 있습니다. FPort 0은 MAC 메시지 전용이고, FPort 224는 MAC 규정 준수 테스트 전용이며, 포트 225~255는 향후 표준화된 애플리케이션 확장 전용입니다.

AWS IoT Core for LoRaWAN에서 규칙 엔진으로 전송되는 업링크 메시지

대상을 추가하면 대상은 규칙 엔진을 사용하여 데이터를 Amazon Location Service로 라우팅하는 AWS IoT 규칙을 생성합니다. 그 후 업데이트된 위치 정보가 Amazon Location 맵에 표시됩니다. 위치 확인을 활성화하지 않은 경우, 대상은 디바이스의 정적 위치 좌표를 업데이트할 때 위치 데이터를 라우팅합니다.

다음 코드는 AWS IoT Core for LoRaWAN에서 전송된 업링크 메시지(위치 정보, 정확도, 솔버 구성 및 무선 메타데이터 포함)의 형식을 보여 줍니다. 아래에 강조 표시된 필드는 선택 사항입니다. 수직 정확도 정보가 없는 경우, 값은 null입니다.

```
{
    // 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"

        }
    }
}
```

콘솔을 사용하여 디바이스의 위치 구성

AWS Management Console을 사용하여 디바이스의 위치를 구성하고 관리하려면, 먼저 콘솔에 로그인한 다음 AWS IoT 콘솔의 Devices(디바이스) 허브 페이지로 이동합니다.

위치 정보 추가

디바이스의 위치 정보를 추가하려면:

1. 디바이스(Devices) 허브 페이지에서 무선 디바이스 추가(Add wireless device)를 선택합니다.
2. 무선 디바이스 사양, 디바이스 및 서비스 프로파일, 데이터를 다른 AWS 서비스로 라우팅하기 위한 IoT 규칙을 정의하는 대상을 입력합니다. 자세한 내용은 AWS IoT Core for LoRaWAN에 디바이스 온보딩 단원을 참조하십시오.
3. 위치 정보를 입력하고, 선택적으로 지리적 위치를 활성화하고, 메시지 라우팅에 사용할 위치 데이터 대상을 지정합니다.
   - 위치 정보

     위도 및 경도 좌표와 선택적으로 고도 좌표를 사용하여 디바이스의 위치 데이터를 지정합니다. 위치 정보는 WGS84 좌표계를 기반으로 합니다.
   - 지리적 위치

AWS IoT Core for LoRaWAN이 디바이스 위치를 계산하는 데 지리적 위치를 사용하도록 하려면 위치 확인을 활성화합니다. 서드 파티 GNSS 및 Wi-Fi 솔버를 사용하여 디바이스의 위치가 실시간으로 식별됩니다.

지리적 위치 정보를 입력하려면 위치 확인 활성화를 선택하고 GNSS 및 Wi-Fi 스캔 데이터를 AWS IoT Core for LoRaWAN에 전달하기 위한 지리적 위치 프레임 포트를 입력합니다. 참조용으로 기본 FPort가 채워져 있습니다. 그러나 1부터 223 사이에서 다른 값을 선택할 수 있습니다.

- 위치 데이터 대상

  디바이스의 위치 데이터를 처리하고 처리된 위치를 AWS IoT Core for LoRaWAN으로 전달하는 AWS IoT 규칙을 설명하기 위한 대상을 선택합니다. 이 대상은 위치 데이터의 라우팅에만 사용합니다. 디바이스 데이터를 다른 AWS 서비스로 라우팅하는 데 사용하는 대상과 달라야 합니다.

디바이스의 위치 구성 보기

디바이스의 위치를 구성한 후 AWS IoT Core for LoRaWAN은 `iotwireless.map`으로 불리는 Amazon Location 맵을 생성합니다. 이 맵은 위치(Position) 탭의 디바이스 세부 정보 페이지에서 확인할 수 있습니다. 지정한 위치 좌표 또는 서드 파티 솔버에 따라 계산된 위치를 기반으로 디바이스의 위치가 맵에 마커로 표시됩니다. 확대 또는 축소하여 맵에서 디바이스의 위치를 명확하게 볼 수 있습니다. 디바이스 세부 정보 페이지의 위치(Position) 탭에는 정확도 정보, 디바이스의 위치가 결정된 타임스탬프 및 지정한 위치 데이터 대상도 표시됩니다.

> **Note**
>
> Amazon Location Service 맵을 활성화하지 않은 경우, 맵에 액세스하고 위치를 확인하려면 Amazon Location Service를 사용해야 한다는 메시지가 표시됩니다. Amazon Location Service 맵을 사용하면 AWS 계정에 추가 요금이 발생할 수 있습니다. 자세한 내용은 AWS IoT Core 요금을 참조하십시오.

`iotwireless.map` 맵은 `GetMapTile`과 같은 `Get` API 작업을 사용하여 액세스되는 지도 데이터의 소스 역할을 합니다. 맵과 함께 사용되는 `Get` API에 대한 자세한 정보는 Amazon Location Service API 참조를 참고하세요.

이 맵에 대한 추가 세부 정보를 보려면 Amazon Location Service 콘솔로 이동하여 맵(Maps)을 선택한 다음 iotwireless.map을 선택합니다. 자세한 정보는 Amazon Location Service 개발자 안내서의 맵을 참조하세요.

디바이스의 위치 구성 업데이트

디바이스의 위치 구성을 변경하려면 디바이스 세부 정보 페이지에서 Edit(편집)을 선택한 다음 위치 정보, 지리적 위치 설정 및 대상을 업데이트합니다.

> **Note**
>
> 과거 위치 데이터에 대한 정보는 제공되지 않습니다. 디바이스의 위치 좌표를 업데이트하면 이전에 보고된 위치 데이터를 덮어씁니다. 위치를 업데이트한 후 디바이스 세부 정보의 위치(Position) 탭에서 새 위치 정보를 볼 수 있습니다. 타임스탬프의 변화는 새 위치 정보가 마지막으로 알려진 게이트웨이 위치와 동기화되었다는 것을 나타냅니다.

API를 사용한 디바이스 위치 구성

AWS IoT 무선 API 또는 AWS CLI를 사용하여 위치 정보를 지정하고, 디바이스 위치를 구성하고, 선택적 지리적 위치를 활성화할 수 있습니다.

> **Important**
>
> API 작업 UpdatePosition, GetPosition, PutPositionConfiguration, GetPositionConfiguration 및 ListPositionConfigurations는 더 이상 지원되지 않습니다. 위치 정보를 업데이트하고 검색하기 위한 호출에는 GetResourcePosition 및 UpdateResourcePosition API 작업을 대신 사용해야 합니다.

위치 정보 및 구성 추가

특정 무선 디바이스에 위치 정보를 추가하려면 UpdateResourcePosition API 작업 또는 update-resource-position CLI 명령을 사용하여 좌표를 지정합니다. `WirelessDevice`를 `ResourceType`으로 지정하고, 업데이트할 무선 디바이스의 ID를 `ResourceIdentifier`로 지정하고, 위치 정보를 지정합니다.

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

다음은 *deviceposition.json* 파일의 콘텐츠를 보여줍니다. 지리적 위치 데이터를 전송하기 위한 FPort 값을 지정하려면 위치 확인 객체를 CreateWirelessDevice 및 UpdateWirelessDevice API 작업과 함께 사용합니다.

deviceposition.json의 내용

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

이 명령을 실행하면 출력을 생성하지 않습니다. 지정한 위치 정보를 보려면 `GetResourcePosition` API 작업을 사용합니다.

위치 정보 및 구성 가져오기

특정 무선 디바이스에 대한 위치 정보를 가져오려면 GetResourcePosition API 또는 get-resource-position CLI 명령을 사용합니다. `WirelessDevice`를 `resourceType`으로 지정하고 무선 디바이스 ID를 `resourceIdentifier`로 제공합니다.

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

이 명령을 실행하면 무선 디바이스의 위치 정보가 GeoJSON 페이로드로 표시됩니다. 위치 좌표, 위치 유형 그리고 디바이스의 마지막으로 알려진 위치에 해당하는 타임스탬프 및 정확도 정보를 비롯한 속성에 대한 정보가 표시됩니다.

```
{
     "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 무선를 통한 게이트웨이 관리

AWS IoT Core for LoRaWAN에서 게이트웨이를 사용할 때 고려해야 할 몇 가지 중요한 사항은 다음과 같습니다. AWS IoT Core for LoRaWAN에 게이트웨이를 추가하는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 온보딩 섹션을 참조하세요.

LoRa Basics Station 소프트웨어 요구 사항

AWS IoT Core for LoRaWAN에 연결하려면 LoRaWAN 게이트웨이에서 LoRa Basics Station이라는 소프트웨어가 실행 중이어야 합니다. LoRa Basics Station은 Semtech Corporation에 의해 유지되는 오픈 소스 소프트웨어로, GitHub 리포지토리에 배포되어 있습니다. AWS IoT Core for LoRaWAN은 LoRa Basics Station 버전 2.0.4 이상을 지원합니다. 최신 버전은 2.0.6입니다.

AWS 파트너 디바이스 카탈로그에서 정규화된 게이트웨이 사용

AWS 파트너 디바이스 카탈로그에는 AWS IoT Core for LoRaWAN에 사용할 수 있는 게이트웨이 및 개발자 키트가 포함되어 있습니다. 게이트웨이를 AWS IoT Core에 연결하기 위해 내장 소프트웨어를 수정할 필요가 없으므로 이러한 정규화된 게이트웨이를 사용하는 것이 좋습니다. 이러한 게이트웨이에는 이미 AWS IoT Core for LoRaWAN과 호환되는 BasicStation 소프트웨어 버전이 있습니다.

> **Note**
>
> AWS IoT Core for LoRaWAN에 대한 정규화된 게이트웨이로서 파트너 카탈로그에 나열되지 않은 게이트웨이를 사용하더라도 게이트웨이가 버전 2.0.4 이상의 LoRa Basics Station 소프트웨어를 실행하고 있다면 이 게이트웨이를 계속 사용할 수 있습니다. LoRaWAN 게이트웨이 인증을 위해 TLS 서버 및 클라이언트 인증을 사용해야 합니다.

CUPS 및 LNS 프로토콜 사용

LoRa Basic Station 소프트웨어에는 게이트웨이를 네트워크 서버에 연결하기 위한 두 개의 하위 프로토콜인 LoRaWAN 네트워크 서버(LNS) 및 CUPS(Configuration and Update Server) 프로토콜이 포함되어 있습니다.

LNS 프로토콜은 LoRA Basics Station 호환 게이트웨이와 네트워크 서버 간에 데이터 연결을 설정합니다. LoRA 업링크 및 다운링크 메시지는 보안 WebSockets에서 이 데이터 연결을 통해 교환됩니다.

CUPS 프로토콜은 보안 인증 관리와, 게이트웨이의 원격 구성 및 펌웨어 업데이트를 가능하게 합니다. AWS IoT Core for LoRaWAN은 LoRaWAN 데이터 모으기 및 원격 게이트웨이 관리를 위해 LNS 및 CUPS 엔드포인트를 각각 제공합니다.

자세한 내용은 프로토콜 LNS 및 CUPS 프로토콜을 참조하세요.

주제

- LoRaWAN 게이트웨이의 비커닝 및 필터링 기능 구성
- AWS IoT Core for LoRaWAN에서 CUPS 서비스를 사용하여 게이트웨이 펌웨어 업데이트
- LoRaWAN 다운링크 데이터 트래픽을 수신할 게이트웨이 선택

LoRaWAN 게이트웨이의 비커닝 및 필터링 기능 구성

LoRaWAN 디바이스에서 작업할 때 LoRaWAN 게이트웨이에 대한 특정 선택적 파라미터를 구성할 수 있습니다. 파라미터는 다음과 같습니다.

- 비커닝

  클래스 B LoRaWAN 디바이스에 대한 브리지 역할을 하는 LoRaWAN 게이트웨이에 대한 비커닝 파라미터를 구성할 수 있습니다. 이러한 디바이스는 예약된 시간 슬롯에 다운링크 메시지를 수신하므로 게이트웨이가 이러한 시간 동기화 비콘을 전송하도록 비콘 파라미터를 구성해야 합니다.

- 필터링

  LoRaWAN 게이트웨이에 대한 `NetID` 및 `JoinEUI` 파라미터를 구성하여 디바이스 데이터 트래픽을 필터링할 수 있습니다. 트래픽을 필터링하면 대역폭 사용량을 절약하고 게이트웨이와 LNS 간의 트래픽 흐름을 줄일 수 있습니다.

- 하위 대역

  게이트웨이의 하위 대역을 구성하여 사용하려는 특정 하위 대역을 지정할 수 있습니다. 다양한 하위 대역 사이를 이동할 수 없는 무선 디바이스의 경우 이 기능을 사용하여 특정 하위 대역의 주파수 채널만 사용하는 디바이스와 통신할 수 있습니다.

다음 항목에는 이러한 파라미터 및 구성 방법에 대한 자세한 정보가 포함되어 있습니다. 비커닝 파라미터는 AWS Management Console에서 사용할 수 없으며 AWS IoT 무선 API 또는 AWS CLI를 통해서만 지정할 수 있습니다.

주제

- 비콘을 클래스 B 디바이스로 보내도록 게이트웨이 구성
- 게이트웨이의 하위 밴드 및 필터링 기능 구성

비콘을 클래스 B 디바이스로 보내도록 게이트웨이 구성

클래스 B 무선 디바이스를 AWS IoT Core for LoRaWAN에 온보딩하는 경우 디바이스는 예약된 시간 슬롯에 다운링크 메시지를 수신합니다. 디바이스는 게이트웨이에서 전송하는 시간 동기화된 비콘을 기반으로 이러한 슬롯을 엽니다. 게이트웨이가 이러한 시간 동기 비콘을 전송하려면 AWS IoT Core for LoRaWAN를 사용하여 게이트웨이에 대한 특정 비콘 관련 파라미터를 구성할 수 있습니다.

이러한 비커닝 파라미터를 구성하려면 게이트웨이에서 LoRa Basics Station 소프트웨어 버전 2.0.6을 실행해야 합니다. AWS 파트너 디바이스 카탈로그에서 정규화된 게이트웨이 사용 섹션을 참조하세요.

비커닝 파라미터를 구성하는 방법

> **Note**
>
> 클래스 B 무선 디바이스와 통신하는 경우 게이트웨이에 대한 비커닝 파라미터만 구성하면 됩니다.

`CreateWirelessGateway` API 작업을 사용하여 AWS IoT Core for LoRaWAN에 게이트웨이를 추가할 때 비커닝 파라미터를 구성합니다. API 작업을 호출할 때 게이트웨이에 대한 `Beaconing` 객체를 사용하여 다음 파라미터를 지정하세요. 파라미터를 구성한 후 게이트웨이는 128초 간격으로 디바이스에 비콘을 보냅니다.

- `DataRate`: 비컨을 전송하는 게이트웨이의 데이터 속도입니다.
- `Frequencies`: 게이트웨이가 비콘을 전송할 주파수 목록입니다.

다음은 게이트웨이에 대해 이러한 파라미터를 구성하는 방법을 보여주는 예입니다. `input.json` 파일에는 게이트웨이 인증서 및 프로비저닝 보안 인증과 같은 추가 세부 정보가 포함됩니다. `CreateWirelessGateway` API 작업을 사용하여 AWS IoT Core for LoRaWAN에 게이트웨이를 추가하는 방법에 대한 자세한 내용은 API를 사용하여 게이트웨이 추가를 참조하세요.

> **Note**
>
> AWS IoT 콘솔을 사용하여 AWS IoT Core for LoRaWAN에 게이트웨이를 추가하는 경우 비커닝 파라미터를 사용할 수 없습니다.

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

다음은 `input.json` 파일의 콘텐츠를 보여줍니다.

input.json 내용

```
{
    "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]
    }
}
```

다음 코드는 이 명령 실행의 샘플 출력을 보여줍니다.

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

비커닝 파라미터에 대한 정보 가져오기

GetWirelessGateway API 작업을 사용하여 게이트웨이의 비커닝 파라미터에 대한 정보를 얻을 수 있습니다.

> **Note**
>
> 게이트웨이가 이미 온보딩된 경우 UpdateWirelessGateway API 작업을 사용하여 비컨 파라미터를 구성할 수 없습니다. 파라미터를 구성하려면 CreateWirelessGateway API 작업을 사용하여 게이트웨이를 추가할 때 게이트웨이를 삭제한 후 파라미터를 지정해야 합니다.

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

이 명령을 실행하면 게이트웨이 및 비커닝 파라미터에 대한 정보가 반환됩니다.

게이트웨이의 하위 밴드 및 필터링 기능 구성

LoRaWAN 게이트웨이는 게이트웨이를 AWS IoT Core for LoRaWAN에 연결할 수 있는 LoRa Basics Station 소프트웨어를 실행합니다. AWS IoT Core for LoRaWAN에 연결하기 위해 LoRA 게이트웨이는 먼저 CUPS 서버에 LNS 엔드포인트를 쿼리한 다음 해당 엔드포인트와의 WebSockets 데이터 연결을 설정합니다. 연결이 설정되면 해당 연결을 통해 업링크 및 다운링크 프레임을 교환할 수 있습니다.

게이트웨이가 수신한 LoRa 데이터 프레임 필터링

LoRaWAN 게이트웨이가 엔드포인트에 대한 연결을 설정하면 AWS IoT Core for LoRaWAN은 NetID 및 JoinEui 필터링 파라미터 등 LoRA 게이트웨이의 구성에 대한 파라미터 집합을 지정하는 router_config 메시지로 응답합니다. router_config에 대한 자세한 내용 및 LoRaWAN 네트워크 서버(LNS)와의 연결 설정 방법에 대한 자세한 내용은 프로토콜 LNS를 참조하세요.

```
{
"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
}
```

게이트웨이는 일반적으로 Wi-Fi, 이더넷 또는 셀룰러와 같은 고대역폭 네트워크를 통해 LNS에서 LoRaWAN 디바이스 데이터를 송수신합니다. 게이트웨이는 일반적으로 모든 메시지를 수집해서 게이트웨이에 들어오는 트래픽을 AWS IoT Core for LoRaWAN에 패스스루합니다. 그러나 일부 디바이스 데이터 트래픽을 필터링하도록 게이트웨이를 구성할 수 있으므로 대역폭 사용량을 절약하고 게이트웨이와 LNS 간의 트래픽 흐름을 줄일 수 있습니다.

데이터 프레임을 필터링하도록 LoRa 게이트웨이를 구성하려면 `router_config` 메시지에서 `NetID` 및 `JoinEui` 파라미터를 사용할 수 있습니다. `NetID`는 허용되는 netID 값 목록입니다. 나열된 것 외의 데이터 프레임을 포함하는 모든 LoRa 데이터 프레임은 삭제됩니다. `JoinEui`는 Joineui 값의 범위를 인코딩하는 정수 값 쌍의 목록입니다. 조인 요청 프레임은 메시지의 `JoinEui` 필드가 [BegEui,EndEui] 범위 내에 있지 않은 한 게이트웨이에 의해 삭제됩니다.

주파수 채널 및 하위 대역

US915 및 AU915 RF 리전의 경우 무선 디바이스는 LoRa 게이트웨이를 사용하여 LoRaWAN 네트워크에 액세스하기 위해 64개의 125kHz 및 8개의 500kHz 업링크 채널을 선택할 수 있습니다. 업링크 주파수 채널은 각각 8개의 125kHz 채널과 1개의 500kHz 채널을 갖는 8개의 하위 대역으로 나뉩니다. AU915 리전의 각 일반 게이트웨이에 대해 하나 이상의 하위 대역이 지원됩니다.

일부 무선 디바이스는 하위 대역 사이를 이동할 수 없으며 AWS IoT Core for LoRaWAN에 연결되어 있을 때 하나의 하위 대역에서만 주파수 채널을 사용합니다. 이러한 디바이스에서 전송되는 업링크 패킷의 경우 특정 하위 대역을 사용하도록 LoRA 게이트웨이를 구성합니다. EU868 등의 다른 RF 리전의 게이트웨이에 대해서는 이 구성이 필요하지 않습니다.

콘솔을 사용하여 게이트웨이 및 하위 대역을 사용하도록 게이트웨이를 구성합니다.

특정 하위 대역을 사용하도록 게이트웨이를 구성하고 LoRA 데이터 프레임을 필터링하는 기능을 활성화할 수도 있습니다. 콘솔을 사용하여 이러한 파라미터를 지정하려면:

1. AWS IoT Core for LoRaWAN 콘솔의 AWS IoT 게이트웨이 페이지로 이동하여 게이트웨이 추가를 선택합니다.
2. 게이트웨이 세부 정보(게이트웨이의 EUI, 주파수 대역(RFRegion) 및 선택 사항인 이름과 설명)을 지정하고 AWS IoT 사물을 게이트웨이에 연결할지 여부를 선택합니다. 게이트웨이를 추가하는 방법에 대한 자세한 내용은 콘솔을 사용하여 게이트웨이 추가 단원을 참조하세요.

3. LoRaWAN 구성 섹션에서 하위 대역과 필터링 정보를 지정할 수 있습니다.
   - SubBands: 하위 대역을 추가하려면 하위 대역 추가를 선택하고 게이트웨이에서 지원하는 하위 대역을 나타내는 정수 값 목록을 지정합니다. SubBands 파라미터는 RfRegion US915 및 AU915에서만 구성할 수 있으며 지원되는 리전 중 하나에서 [1,8] 범위의 값을 가져야 합니다.
   - NetIdFilters: 업링크 프레임을 필터링하려면 NetId 추가를 선택하고 게이트웨이가 사용하는 문자열 값 목록을 지정합니다. 무선 디바이스에서 들어오는 업링크 프레임의 netID가 나열된 값 중 하나 이상과 일치해야 합니다. 그렇지 않으면 프레임이 삭제됩니다.
   - JoinEuiFilters: JoinEui 범위 추가를 선택하고 게이트웨이가 LoRa 프레임을 필터링하는 데 사용하는 문자열 값 쌍 목록을 지정합니다. 무선 디바이스의 조인 요청의 일부로 지정된 JoinEui 값은 각각 [BegeUI, EndeUI] 쌍으로 나열되는 JoinEUI 값 중 하나 이상의 범위 내에 있어야 합니다. 그렇지 않으면 프레임이 삭제됩니다.
4. 그런 다음 콘솔을 사용하여 게이트웨이 추가에 설명된 지침에 따라 게이트웨이를 계속 구성할 수 있습니다.

AWS IoT Core for LoRaWAN 콘솔의 AWS IoT 게이트웨이 페이지에서 게이트웨이를 추가한 후, 추가한 게이트웨이를 선택하면 게이트웨이 세부 정보 페이지의 LoRaWAN 세부 정보 섹션에서 SubBands와 필터 NetIdFilters 및 JoinEuiFilters를 볼 수 있습니다.

API를 사용하여 게이트웨이 및 하위 대역을 사용하도록 게이트웨이를 구성합니다.

게이트웨이를 생성하기 위해 사용하는 CreateWirelessGateway API를 통해 사용할 하위 대역을 구성하고 필터링 기능을 활성화할 수 있습니다. CreateWirelessGateway API를 사용하여, LoRaWAN 필드에 입력하는 게이트웨이 구성 정보의 일부로서 하위 대역과 필터를 지정할 수 있습니다. 다음은 이 정보를 포함하는 요청 토큰을 보여 줍니다.

```
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
}
```

또한 UpdateWirelessGateway API를 사용하여 필터를 업데이트할 수 있지만 하위 대역은 업데이트할 수 없습니다. `JoinEuiFilters` 및 `NetIdfilters` 값이 null이면 필드에 대한 업데이트가 없음을 의미합니다. 값이 null이 아니고 빈 목록이 포함되어 있으면 업데이트가 적용됩니다. 지정한 필드의 값을 가져오려면 GetWirelessGateway API를 사용하세요.

AWS IoT Core for LoRaWAN에서 CUPS 서비스를 사용하여 게이트웨이 펌웨어 업데이트

게이트웨이에서 실행되는 LoRa Basics Station 소프트웨어는 CUPS(Configuration and Update Server) 프로토콜을 사용하여 자격 증명 관리 및 펌웨어 업데이트 인터페이스를 제공합니다. CUPS 프로토콜은 ECDSA 서명과 함께 안전한 펌웨어 업데이트를 제공합니다.

게이트웨이의 펌웨어를 자주 업데이트해야 합니다. AWS IoT Core for LoRaWAN에서 CUPS 서비스를 사용하여 게이트웨이에 펌웨어 업데이트를 제공할 수 있으며 게이트웨이에서 업데이트 서명도 가능합니다. 게이트웨이의 펌웨어를 업데이트하기 위해 SDK 또는 CLI를 사용할 수 있지만 콘솔은 사용할 수 없습니다.

이 업데이트 프로세스를 완료하는 데 45분이 소요될 수 있습니다. AWS IoT Core for LoRaWAN에 대한 게이트웨이 연결을 처음 설정하는 경우 시간이 더 오래 걸릴 수 있습니다. 게이트웨이 제조업체는 일반적으로 자체 펌웨어 업데이트 파일 및 서명을 제공하므로 이를 대신 사용하고 S3 버킷에 펌웨어 파일 업로드 및 IAM 역할 추가(으)로 진행할 수 있습니다.

펌웨어 업데이트 파일이 없는 경우 펌웨어 업데이트 파일 및 서명 생성 단원에서 애플리케이션에 적용하는 데 사용할 수 있는 예제를 확인할 수 있습니다.

게이트웨이의 펌웨어 업데이트를 수행하려면:

- 펌웨어 업데이트 파일 및 서명 생성
- S3 버킷에 펌웨어 파일 업로드 및 IAM 역할 추가
- 작업 정의를 사용하여 펌웨어 업데이트 예약 및 실행

펌웨어 업데이트 파일 및 서명 생성

이 절차의 단계는 선택 사항이며 사용 중인 게이트웨이에 따라 다릅니다. 게이트웨이 제조업체는 업데이트 파일 또는 스크립트 형태로 자체 펌웨어 업데이트를 제공하며 Basic Station은 백그라운드에서 이 스크립트를 실행합니다. 이 경우 사용 중인 게이트웨이의 릴리스 정보에서 펌웨어 업데이트 파일을 찾을 수 있습니다. 그런 다음 해당 업데이트 파일이나 스크립트를 대신 사용하고 S3 버킷에 펌웨어 파일 업로드 및 IAM 역할 추가(으)로 진행할 수 있습니다.

이 스크립트가 없는 경우 다음은 펌웨어 업데이트 파일을 생성하기 위해 실행하는 명령을 보여 줍니다. 또한 코드가 변경되거나 손상되지 않았으며 신뢰할 수 있는 작성자에 의해 게시된 코드만 디바이스에서 실행된다는 것을 증명하기 위해 업데이트에 서명을 할 수도 있습니다.

이 절차에서는 다음을 수행합니다.

- 펌웨어 업데이트 파일 생성
- 펌웨어 업데이트에 대한 서명 생성
- 다음 단계 검토

펌웨어 업데이트 파일 생성

게이트웨이에서 실행되는 LoRa Basic Station 소프트웨어는 CUPS 응답에서 펌웨어 업데이트를 수신할 수 있습니다. 제조업체에서 제공한 스크립트가 없는 경우 Raspberry Pi 기반 RakWireless 게이트웨이에 대해 작성된 다음 펌웨어 업데이트 스크립트를 참조하세요. 기본 스크립트가 있으며 새로운 스테이션 이진, 버전 파일 및 `station.conf`이(가) 여기에 연결됩니다.

> **Note**
>
> 스크립트는 RAKWireless 게이트웨이에 고유하므로 사용 중인 게이트웨이에 따라 애플리케이션에 맞게 조정해야 합니다.

기본 스크립트

다음은 Raspberry Pi 기반 RAKWireless 게이트웨이에 대한 기본 스크립트 샘플을 보여줍니다. 다음 명령을 `base.sh` 파일에 저장한 다음 터미널의 Raspberry Pi의 웹 브라우저에서 스크립트를 실행합니다.

```
*#!/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
```

페이로드 스크립트 추가

기본 스크립트에 Basics Station 이진, 업데이트 버전을 식별하는 version.txt, addpayload.sh라는 스크립트의 station.conf를 추가합니다. 그런 다음 이 스크립트를 실행합니다.

```
*#!/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
```

이러한 스크립트를 실행한 후 터미널에서 다음 명령을 실행하여 펌웨어 업데이트 파일 fwstation을 생성할 수 있습니다.

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

펌웨어 업데이트에 대한 서명 생성

LoRa Basic Station 소프트웨어는 ECDSA 서명과 함께 서명된 펌웨어 업데이트를 제공합니다. 서명된 업데이트를 지원하려면 다음이 필요합니다.

- ECDSA 프라이빗 키에 의해 생성되고 128바이트 미만이어야 하는 서명.
- 서명에 사용되고 sig-%d.key 형식의 파일 이름으로 게이트웨이에 저장되어야 하는 프라이빗 키. sig-0.key 파일 이름을 사용하는 것이 좋습니다.
- 프라이빗 키를 통한 32 비트 CRC.

서명과 CRC는 AWS IoT Core for LoRaWAN API에 전달됩니다. 이전 파일을 생성하려면 GitHub 리포지토리에 있는 basicstation 예제를 활용하는 다음 스크립트 gen.sh를 사용할 수 있습니다.

```
*#!/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
```

스크립트에 의해 생성된 프라이빗 키는 게이트웨이에 저장되어야 합니다. 키 파일은 이진 형식입니다.

```
./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
```

다음 단계 검토

이제 펌웨어와 서명을 생성했으니 다음 주제로 이동하여 펌웨어 파일 `fwstation`을 Amazon S3 버킷에 업로드하세요. 버킷은 펌웨어 업데이트 파일을 객체로 저장하는 컨테이너입니다. S3 버킷의 펌웨어 업데이트 파일을 읽을 수 있는 권한을 CUPS 서버에 부여하는 IAM 역할을 추가할 수 있습니다.

S3 버킷에 펌웨어 파일 업로드 및 IAM 역할 추가

Amazon S3를 사용하여 버킷을 생성합니다. 버킷은 펌웨어 업데이트 파일을 저장할 수 있는 컨테이너입니다. 파일을 S3 버킷에 업로드하고 CUPS 서버가 버킷에서 업데이트 파일을 읽을 수 있도록 하는 IAM 역할을 추가할 수 있습니다. Amazon S3에 대한 자세한 내용은 S3 시작하기를 참조하세요.

업로드하려는 펌웨어 업데이트 파일은 사용 중인 게이트웨이에 따라 다릅니다. 펌웨어 업데이트 파일 및 서명 생성에 설명된 것과 유사한 절차를 수행한 경우 스크립트를 실행하여 생성된 `fwstation` 파일을 업로드합니다.

이 절차를 완료하는 데 약 20분이 소요됩니다.

펌웨어 파일을 업로드하려면:

- Amazon S3 버킷 생성 및 업데이트 파일 업로드
- S3 버킷을 읽을 수 있는 권한이 있는 IAM 역할 생성
- 다음 단계 검토

Amazon S3 버킷 생성 및 업데이트 파일 업로드

AWS Management Console을 사용하여 Amazon S3 버킷을 생성한 다음 펌웨어 업데이트 파일을 버킷에 업로드합니다.

S3 버킷 생성

Amazon S3 버킷을 생성하려면 Amazon S3 콘솔을 엽니다. 아직 로그인하지 않았다면 로그인하고 다음 단계를 수행하세요.

1. 버킷 생성을 선택합니다.
2. 버킷 이름에 대해 고유하고 의미 있는 이름(예: `iotwirelessfwupdate`)을 입력합니다. 버킷에 대한 권장 명명 규칙은 https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html 단원을 참조하세요.
3. AWS 리전을 LoRaWAN 게이트웨이 및 디바이스를 만드는 데 사용한 것으로 선택했는지, 버킷이 기본 권한을 사용하도록 모든 퍼블릭 액세스 차단 설정이 선택되어 있는지 확인합니다.
4. 버킷 버전 관리(Bucket versioning)에서 활성화(Enable)를 선택합니다. 그러면 여러 가지 버전의 펌웨어 업데이트 파일을 동일한 버킷에 보관하는 데 도움이 됩니다.
5. 서버 측 암호화(Server-side encryption)가 비활성화(Disable)로 설정되어 있는지 확인하고 버킷 생성(Create bucket)을 선택합니다.

펌웨어 업데이트 파일 업로드

이제 AWS Management Console에 표시된 버킷 목록에서 버킷을 볼 수 있습니다. 버킷을 선택하고 다음 단계를 완료하여 파일을 업로드합니다.

1. 버킷을 선택한 다음 업로드를 선택합니다.
2. 파일 추가를 클릭한 다음 펌웨어 업데이트 파일을 업로드합니다. 펌웨어 업데이트 파일 및 서명 생성에서 설명한 절차를 수행한 경우, `fwstation` 파일을 업로드합니다. 그렇지 않으면 게이트웨이 제조업체에서 제공한 파일을 업로드합니다.
3. 모든 설정이 기본값으로 설정되어 있는지 확인합니다. 미리 정의된 ACL(Predefined ACLs)이 프라이빗(private)으로 설정되어 있는지 확인하고 업로드를 선택하여 파일을 업로드합니다.
4. 업로드한 파일의 S3 URI를 복사합니다. 버킷을 선택하면 업로드한 파일이 객체 목록에 표시됩니다. 파일을 선택한 후 S3 URI 복사를 선택합니다. URI는 이전에 설명한 예제(`fwstation`)와 비슷한 버킷 이름을 지정한 경우 `s3://iotwirelessfwupdate/fwstation` 형태입니다. IAM 역할을 만들 때 S3 URI를 사용합니다.

S3 버킷을 읽을 수 있는 권한이 있는 IAM 역할 생성

이제 CUPS에 S3 버킷에서 펌웨어 업데이트 파일을 읽을 수 있는 권한을 부여하는 IAM 역할과 정책을 생성합니다.

역할에 대한 IAM 정책 생성

AWS IoT Core for LoRaWAN 대상 역할에 대한 IAM 정책을 생성하려면 IAM 콘솔의 정책 허브를 열고 다음 단계를 완료합니다.

1. 정책 생성을 선택한 후 JSON 탭을 선택합니다.
2. 편집기의 모든 내용을 삭제하고 이 정책 문서를 붙여 넣습니다. 정책은 `iotwireless` 버킷과 객체 내부에 저장된 펌웨어 업데이트 파일인 `fwstation`에 액세스할 수 있는 권한을 제공합니다.

```
{
    "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. 정책 검토를 선택하고 이름에 이 정책의 이름(예: `IoTWirelessFwUpdatePolicy`)을 입력합니다. 다음 절차에서 이 이름을 사용해야 합니다.
4. 정책 생성을 선택합니다.

연결된 정책으로 IAM 역할 생성

이제 IAM 역할을 생성하여 이전에 S3 버킷을 액세스하기 위해 생성한 정책을 연결합니다. IAM 콘솔의 역할 허브를 열고 다음 단계를 완료합니다.

1. 역할 생성을 선택합니다.
2. 신뢰할 수 있는 엔터티 유형 선택(Select type of trusted entity)에서 다른 AWS 계정을 선택합니다.
3. 계정 ID에 AWS 계정 ID를 입력한 후 다음: 권한을 선택합니다.
4. 검색 상자에 이전 절차에서 생성한 IAM 정책의 이름을 입력합니다. 검색 결과에서 이전에 만든 IAM 정책(예: `IoTWirelessFwUpdatePolicy`)을 확인하고 선택합니다.
5. Next: Tags(다음: 태그)를 선택한 후 Next: Review(다음: 검토)를 선택합니다.
6. 역할 이름에 이 역할의 이름(예: `IoTWirelessFwUpdateRole`)을 입력한 후 역할 생성을 선택합니다.

IAM 역할의 신뢰 관계 편집

이전 단계를 실행한 후 표시되는 확인 메시지에서 생성한 역할 이름을 선택하고 편집합니다. 다음 신뢰 관계를 추가하기 위해 역할을 편집합니다.

1. 생성한 역할에 대한 요약 페이지에서 신뢰 관계 탭을 선택한 후 신뢰 관계 편집을 선택합니다.
2. 정책 문서에서 `Principal` 속성을 다음 예시처럼 변경합니다.

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

`Principal` 속성을 변경한 후 전체 정책 문서가 다음 예시와 같은 형식이어야 합니다.

```
{
```

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

3. 변경 사항을 저장하고 종료하려면 신뢰 정책 업데이트(Update Trust Policy)를 선택합니다.
4. 사용자 역할에 대한 ARN을 획득합니다. IAM 역할을 선택하면 요약 섹션에 역할 ARN(예: `arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`)이 표시됩니다. 역할 ARN을 복사합니다.

다음 단계 검토

이제 S3 버킷과 CUPS 서버가 S3 버킷을 읽을 수 있도록 하는 IAM 역할을 만들었으므로 다음 항목으로 이동하여 펌웨어 업데이트를 예약하고 실행합니다. 펌웨어 업데이트 수행을 위해 실행될 작업 정의를 생성할 때 입력할 수 있도록, 이전에 복사한 S3 URI 및 역할 ARN을 기록해 둡니다.

작업 정의를 사용하여 펌웨어 업데이트 예약 및 실행

작업 정의를 사용하여 펌웨어 업데이트에 대한 세부 정보를 포함하고 업데이트를 정의할 수 있습니다. AWS IoT Core for LoRaWAN은 게이트웨이와 관련된 다음 세 필드의 정보를 기반으로 펌웨어 업데이트를 제공합니다.

- 스테이션

  Basic Station 소프트웨어의 버전 및 빌드 시간입니다. 이 정보를 식별하기 위해, 게이트웨이에 의해 실행되는 Basic Station 소프트웨어를 사용하여 이 정보를 생성할 수도 있습니다(예: `2.0.5(rpi/std) 2021-03-09 03:45:09`).

- PackageVersion

  게이트웨이에서 `version.txt` 파일에 의해 지정된 펌웨어 버전입니다. 이 정보는 게이트웨이에 없을 수도 있지만 펌웨어 버전을 정의하는 방법으로 사용하는 것이 좋습니다(예: `1.0.0`).

- 모델

게이트웨이에서 사용 중인 플랫폼 또는 모델(예: Linux)입니다.

이 절차를 완료하는 데 20분이 소요됩니다.

이 절차를 완료하려면:

- 게이트웨이에서 실행 중인 현재 버전 가져오기
- 무선 게이트웨이 작업 정의 생성
- 펌웨어 업데이트 작업 실행 및 진행률 추적

게이트웨이에서 실행 중인 현재 버전 가져오기

펌웨어 업데이트에 대한 게이트웨이의 적격성을 확인하기 위해 CUPS 서버는 게이트웨이가 CUPS 요청 중에 세 개의 필드 `Station`, `PackageVersion`, 및 `Model` 정보를 제시할 때 해당 정보가 일치하는지 확인합니다. 작업 정의를 사용할 때 이러한 필드는 `CurrentVersion` 필드의 일부로 저장됩니다.

AWS IoT Core for LoRaWAN API 또는 AWS CLI를 사용하여 게이트웨이를 위한 `CurrentVersion`을 얻습니다. 다음 명령은 CLI를 사용하여 이 정보를 가져오는 방법을 보여줍니다.

1. 게이트웨이를 이미 프로비저닝한 경우 get-wireless-gateway 명령을 사용하여 게이트웨이에 대한 정보를 가져올 수 있습니다.

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

다음은 이 명령에 대한 샘플 출력의 일부입니다.

```
{
    "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` 명령에 의해 보고된 무선 게이트웨이 ID를 사용하여 get-wireless-gateway-firmware-information 명령을 통해 `CurrentVersion`를 가져올 수 있습니다.

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

다음은 명령에 대한 샘플 출력이며, `CurrentVersion`에 의해 표시되는 세 필드 정보를 보여줍니다.

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

무선 게이트웨이 작업 정의 생성

작업 정의를 만들 때 AutoCreateTasks 파라미터를 사용하여 작업의 자동 생성을 지정하는 것이 좋습니다. `AutoCreateTasks`는 앞에서 언급한 세 파라미터 모두와 일치하는 게이트웨이에 적용됩니다. 이 파라미터를 비활성화하면 파라미터를 게이트웨이에 수동으로 할당해야 합니다.

AWS IoT Core for LoRaWAN API 또는 AWS CLI를 사용하여 무선 게이트웨이 작업 정의를 생성할 수 있습니다. 다음 명령은 CLI를 사용하여 작업 정의를 만드는 방법을 보여줍니다.

1. `CreateWirelessGatewayTaskDefinition` API에 전달할 정보가 포함될 `input.json` 파일을 생성합니다. `input.json` 파일에서, 이전에 얻은 다음 정보를 제공합니다.

   - UpdateDataSource

     S3 버킷에 업로드한 펌웨어 업데이트 파일이 들어 있는 객체에 대한 링크를 제공합니다(예: `s3://iotwirelessfwupdate/fwstation`).

   - UpdateDataRole

S3 버킷을 읽을 수 있는 권한을 제공하는 IAM 역할의 역할 ARN에 대한 링크를 제공합니다(예: `arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`).

- SigKeyCRC 및 UpdateSignature

  이 정보는 게이트웨이 제조업체에서 제공할 수 있지만 펌웨어 업데이트 파일 및 서명 생성에 설명된 절차를 따랐다면 서명을 생성할 때 이 정보를 찾을 수 있습니다.

- CurrentVersion

  `get-wireless-gateway-firmware-information` 명령을 실행하여 이전에 얻은 `CurrentVersion` 출력을 제공하세요.

```
cat input.json
```

다음은 `input.json` 파일의 내용을 보여줍니다.

```
{
    "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` 파일을 create-wireless-gateway-task-definition 명령에 전달하여 작업 정의를 생성합니다.

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

다음은 해당 명령 출력을 보여 줍니다.

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

펌웨어 업데이트 작업 실행 및 진행률 추적

게이트웨이는 펌웨어 업데이트를 받을 준비가 되어 있으며 전원이 켜지면 CUPS 서버에 연결됩니다. CUPS 서버가 게이트웨이 버전의 일치를 발견하면 펌웨어 업데이트를 예약합니다.

작업은 프로세스의 작업 정의입니다. `AutoCreateTasks`을(를) `True`(으)로 설정하여 자동 작업 생성을 지정했으므로 일치하는 게이트웨이가 발견되는 즉시 펌웨어 업데이트 작업이 시작됩니다.

`GetWirelessGatewayTask` API를 사용하여 작업의 진행을 추적할 수 있습니다. get-wireless-gateway-task 명령을 처음 실행하면 작업 상태가 `IN_PROGRESS`로 표시됩니다.

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

다음은 해당 명령 출력을 보여 줍니다.

```
{
    "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"
}
```

다음에 명령을 실행할 때 펌웨어 업데이트가 적용되면 업데이트된 필드 `Package`, `Version` 및 `Model`이 표시되고 작업 상태가 `COMPLETED`로 변경됩니다.

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

다음은 해당 명령 출력을 보여 줍니다.

```
{
    "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"
}
```

이 예에서는 Raspberry Pi 기반 RAKWireless 게이트웨이를 사용하여 펌웨어 업데이트를 보여 주었습니다. 펌웨어 업데이트 스크립트는 실행 중인 BasicStation을 중지하여 업데이트된 `Package`, `Version` 및 `Model` 필드를 저장하므로 BasicStation을 다시 시작해야 합니다.

```
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
```

펌웨어 업데이트가 실패하면 CUPS 서버에서 상태 `FIRST_RETRY`가 표시되고 게이트웨이가 동일한 요청을 보냅니다. CUPS 서버가 `SECOND_RETRY` 이후에 게이트웨이에 연결할 수 없으면 `FAILED` 상태를 표시합니다.

이전 작업이 `COMPLETED` 또는 `FAILED`였으면 새 작업을 시작하기 전에 delete-wireless-gateway-task 명령을 사용하여 이전 작업을 삭제합니다.

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

LoRaWAN 다운링크 데이터 트래픽을 수신할 게이트웨이 선택

AWS IoT Core for LoRaWAN에서 디바이스로 다운링크 메시지를 보낼 때 다운링크 데이터 트래픽에 사용할 게이트웨이를 선택할 수 있습니다. 개별 게이트웨이를 지정하거나 게이트웨이 목록에서 선택하여 다운링크 트래픽을 수신할 수 있습니다.

게이트웨이 목록을 지정하는 방법

`SendDataToWirelessDevice` API 작업을 사용하여 AWS IoT Core for LoRaWAN에서 디바이스로 다운링크 메시지를 보낼 때 사용할 개별 게이트웨이 또는 게이트웨이 목록을 지정할 수 있습니다. API 작업을 호출할 때 게이트웨이에 대한 `ParticipatingGateways` 객체를 사용하여 다음 파라미터를 지정하세요.

> **Note**
>
> 사용하려는 게이트웨이 목록이 AWS IoT 콘솔에 없습니다. `SendDataToWirelessDevice` API 작업 또는 CLI를 사용할 때만 사용할 게이트웨이 목록을 지정할 수 있습니다.

- `DownlinkMode`: 다운링크 메시지를 순차 모드로 보낼지 아니면 동시 모드로 보낼지를 나타냅니다. 클래스 A 장치의 경우 이전 업링크 메시지 전송에서 선택한 게이트웨이만 사용하려면 `UsingUplinkGateway`를 지정합니다.
- `GatewayList`: 다운링크 데이터 트래픽을 보내는 데 사용할 게이트웨이 목록입니다. 다운링크 페이로드는 지정된 빈도로 지정된 게이트웨이로 전송됩니다. 이는 `GatewayId` 및 `DownlinkFrequency` 쌍으로 구성된 `GatewayListItem` 객체 목록을 사용하여 표시됩니다.
- `TransmissionInterval`: AWS IoT Core for LoRaWAN가 페이로드를 다음 게이트웨이로 전송하기 전에 대기하는 시간입니다.

> **Note**
>
> 다운링크 메시지를 클래스 B 또는 클래스 C 무선 디바이스로 전송할 때만 사용할 게이트웨이 목록을 지정할 수 있습니다. 클래스 A 디바이스를 사용하는 경우 다운링크 메시지가 디바이스로 전송될 때 업링크 메시지를 보낼 때 선택한 게이트웨이가 사용됩니다.

다음은 게이트웨이에 대해 이러한 파라미터를 지정하는 방법을 보여주는 예입니다. `input.json` 파일에는 추가 세부 정보가 포함됩니다. `SendDataToWirelessDevice` API 작업을 사용하여 다운링크

메시지를 전송하는 방법에 대한 자세한 내용은 API를 사용하여 다운링크 대기열 작업 수행을 참조하세요.

> **Note**
>
> AWS IoT 콘솔을 사용하여 AWS IoT Core for LoRaWAN에서 다운링크 메시지를 보낼 때는 참여 게이트웨이 목록을 지정하기 위한 파라미터를 사용할 수 없습니다.

```
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
```

다음은 `input.json` 파일의 콘텐츠를 보여줍니다.

input.json 내용

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

이 명령을 실행하면 출력에 다운링크 메시지의 `MessageId`가 생성됩니다. 경우에 따라, `MessageId`를 수신하더라도 패킷이 삭제될 수 있습니다. 이 오류를 해결하는 방법에 대한 자세한 내용은 다운링크 메시지 대기열 오류 문제 해결 섹션을 참조하세요.

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

참여 게이트웨이 목록에 대한 정보 가져오기

다운링크 대기열에 메시지를 나열하면 다운링크 메시지를 받는 데 참여하는 게이트웨이 목록에 대한 정보를 얻을 수 있습니다. 메시지를 나열하려면 `ListQueuedMessages` API를 사용하세요.

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

이 명령을 실행하면 큐에 있는 메시지 및 해당 매개 변수에 대한 정보가 반환됩니다.

AWS IoT Core for LoRaWAN에서 디바이스 관리

다음은 AWS IoT Core for LoRaWAN에서 디바이스를 사용할 때 고려해야 할 몇 가지 중요한 사항입니다. AWS IoT Core for LoRaWAN에 디바이스 추가 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 디바이스 온보딩 섹션을 참조하세요.

디바이스 고려 사항

AWS IoT Core for LoRaWAN과 통신하기 위해 사용하려는 디바이스를 선택할 때 다음을 고려하세요.

- 사용 가능한 센서
- 배터리 용량
- 에너지 소비
- 비용
- 안테나 유형 및 전송 범위

AWS IoT Core for LoRaWAN에 적합한 게이트웨이가 있는 디바이스 사용

사용하는 디바이스를 AWS IoT Core for LoRaWAN과 함께 사용할 수 있는 무선 게이트웨이와 페어링할 수 있습니다. AWS 파트너 디바이스 카탈로그에서 이러한 게이트웨이 및 개발자 키트를 찾을 수 있

습니다. 또한 게이트웨이와 이러한 디바이스의 근접성을 고려하는 것이 좋습니다. 자세한 내용은 AWS 파트너 디바이스 카탈로그에서 정규화된 게이트웨이 사용 단원을 참조하십시오.

LoRaWAN 버전

AWS IoT Core for LoRaWAN은 LoRa Alliance에 의해 표준화된 1.0.x 또는 1.1 LoRaWAN 사양을 준수하는 모든 디바이스를 지원합니다.

활성화 모드

LoRaWAN 디바이스에서 업링크 데이터를 전송하기 전에 활성화 또는 조인 프로시저라는 프로세스를 완료해야 합니다. 디바이스를 활성화하려면 OTAA(무선 업데이트 활성화) 또는 ABP(개인 설정으로 활성화)를 사용할 수 있습니다. 각 활성화에 대해 새 세션 키가 생성되므로 OTAA를 사용하여 디바이스를 활성화하는 것이 좋습니다.

무선 디바이스 사양은 각 활성화에 대해 생성된 루트 키와 세션 키를 결정하는 LoRaWAN 버전 및 활성화 모드를 기반으로 합니다. 자세한 내용은 콘솔을 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가 단원을 참조하십시오.

디바이스 클래스

LoRaWAN 디바이스는 언제든지 업링크 메시지를 전송할 수 있습니다. 다운링크 메시지를 수신 대기하면 배터리 용량을 소모하고 배터리 지속 시간을 줄입니다. LoRaWAN 프로토콜은 세 가지 클래스의 LoRaWAN 디바이스를 지정합니다.

- 클래스 A 디바이스는 대부분의 시간을 절전 모드로 전환하고 짧은 시간 동안만 다운링크 메시지를 수신 대기합니다. 이 디바이스는 대부분 배터리 수명이 최대 10년인 배터리 전원 센서입니다.
- 클래스 B 디바이스는 예약된 다운링크 슬롯에서 메시지를 수신할 수 있습니다. 이러한 디바이스는 대부분 배터리 전원 액추에이터입니다.
- Class C 디바이스는 휴면 상태가 아니어서 계속 메시지 수신 대기하기 때문에 메시지 수신에 많은 지연이 없습니다. 이 디바이스는 대부분 주전원에 연결된 액추에이터입니다.

이러한 무선 디바이스 고려 사항에 대한 자세한 내용은 LoRaWAN에 대해 자세히 알아보기에 언급된 리소스를 참조하세요.

주제

- AWS IoT Core for LoRaWAN을 사용하여 적응형 데이터 속도(ADR) 수행

- LoRaWAN 디바이스와 AWS IoT 간의 통신 관리
- 퍼블릭 LoRaWAN 디바이스 네트워크에서 LoRaWAN 트래픽 관리(Everynet)

AWS IoT Core for LoRaWAN을 사용하여 적응형 데이터 속도(ADR) 수행

엔드 디바이스의 메시지가 게이트웨이에서 수신되도록 하면서 디바이스 전송 전력 소비를 최적화하기 위해 AWS IoT Core for LoRaWAN은 적응형 데이터 속도를 사용합니다. 적응형 데이터 속도는 게이트웨이에서 수신되는 패킷의 오류율을 줄이면서 데이터 속도, 전송 전력 및 재전송 횟수를 최적화하도록 최종 디바이스에 지시합니다. 예를 들어, 엔드 디바이스가 게이트웨이 근처에 있는 경우 적응형 데이터 속도는 전송 전력을 줄이고 데이터 전송 속도를 높입니다.

주제

- 적응형 데이터 속도(ADR) 작동 방식
- 데이터 속도 제한(CLI) 구성

적응형 데이터 속도(ADR) 작동 방식

ADR을 활성화하려면 디바이스가 프레임 헤더에 ADR 비트를 설정해야 합니다. ADR 비트가 설정되면 AWS IoT Core for LoRaWAN은 `LinkADRReq` MAC 명령을 보내고 디바이스는 ADR 명령의 ACK 상태가 포함된 `LinkADRAns` 명령으로 응답합니다. 디바이스가 ADR 명령을 ACK로 전송하면 AWS IoT Core for LoRaWAN의 ADR 지침에 따라 최적의 데이터 속도를 위해 전송 파라미터 값을 조정합니다.

AWS IoT Core for LoRaWAN ADR 알고리즘은 업링크 메타데이터 기록의 SINR 정보를 사용하여 기기에 사용할 최적의 전송 전력과 데이터 속도를 결정합니다. 알고리즘은 프레임 헤더에 ADR 비트가 설정되면 시작되는 가장 최근의 업링크 메시지 20개를 사용합니다. 재전송 횟수를 결정하기 위해 총 손실된 패킷 수의 백분율인 패킷 오류율(PER)을 사용합니다. 이 알고리즘을 사용하면 데이터 속도 범위, 즉 데이터 속도의 최소 및 최대 제한만 제어할 수 있습니다.

데이터 속도 제한(CLI) 구성

기본적으로 AWS IoT Core for LoRaWAN은 LoRaWAN 디바이스의 프레임 헤더에 ADR 비트를 설정할 때 ADR을 수행합니다. AWS IoT 무선 API 작업 `CreateServiceProfile` 또는 AWS CLI 명령 `create-service-profile`을 사용하여 LoRaWAN 디바이스에 대한 서비스 프로필을 생성할 때 데이터 속도의 최소 및 최대 제한을 제어할 수 있습니다.

> **Note**
>
> AWS Management Console에서 서비스 프로필을 생성할 때는 최대 및 최소 데이터 속도 제한을 지정할 수 없습니다. AWS IoT 무선 API 또는 AWS CLI로만 지정할 수 있습니다.

데이터 속도의 최소 및 최대 제한을 지정하려면 `CreateServiceProfile` API 작업과 함께 `DrMin` 및 `DrMax` 파라미터를 사용하세요. 기본 최소 및 최대 데이터 속도 제한은 0과 15입니다. 예를 들어, 다음 CLI 명령은 최소 데이터 속도 제한을 3으로 설정하고 최대 제한은 12로 설정합니다.

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

이 명령을 실행하면 서비스 프로필의 ID와 Amazon 리소스 이름(ARN)이 생성됩니다.

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

AWS IoT 무선 API 작업 GetServiceProfile 또는 AWS CLI 명령 get-service-profile를 사용하여 지정된 파라미터 값을 가져올 수 있습니다.

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

이 명령을 실행하면 서비스 프로필 파라미터의 값이 생성됩니다.

```
{
    "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
    }
}
```

프로필을 여러 개 만든 경우 API 작업 `ListServiceProfiles` 또는 AWS CLI 명령 `list-service-profiles`를 사용하여 AWS 계정의 서비스 프로필을 나열한 다음 `GetServiceProfile` API 또는 `get-service-profile` CLI 명령을 사용하여 데이터 속도 제한을 사용자 지정한 서비스 프로필을 검색할 수 있습니다.

LoRaWAN 디바이스와 AWS IoT 간의 통신 관리

LoRaWAN 디바이스를 AWS IoT Core for LoRaWAN에 연결한 후에는 디바이스가 클라우드로 메시지 전송을 시작할 수 있습니다. 업링크 메시지는 디바이스에서 전송하고 AWS IoT Core for LoRaWAN에서 수신하는 메시지입니다. LoRaWAN 디바이스는 언제든지 업링크 메시지를 전송할 수 있으며, 이 메시지는 다른 AWS 서비스 및 클라우드 호스팅 애플리케이션으로 전달됩니다. AWS IoT Core for LoRaWAN과 기타 AWS 서비스 및 애플리케이션에서 디바이스로 전송한 메시지를 다운링크 메시지라고 합니다.

다음은 디바이스와 클라우드 간에 전송되는 업링크 및 다운링크 메시지를 보고 관리하는 방법을 보여줍니다. 다운링크 메시지의 대기열을 유지하고 이러한 메시지를 대기열에 추가된 순서대로 디바이스로 전송할 수 있습니다.

주제

- LoRaWAN 디바이스에서 전송된 업링크 메시지 형식 보기
- 대기열에 LoRaWAN 장치로 전송할 다운링크 메시지 추가

LoRaWAN 디바이스에서 전송된 업링크 메시지 형식 보기

LoRaWAN 디바이스를 AWS IoT Core for LoRaWAN에 연결했으면 무선 디바이스에서 수신할 업링크 메시지의 형식을 확인할 수 있습니다.

업링크 메시지를 관찰하기 전에

데이터를 송수신할 수 있도록 무선 디바이스를 온보딩하고 디바이스를 AWS IoT에 연결해야 합니다. 디바이스를 AWS IoT Core for LoRaWAN에 온보딩하는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 디바이스 온보딩 섹션을 참조하세요.

업링크 메시지에는 무엇이 포함되어 있습니까?

LoRaWAN 게이트웨이를 사용하여 LoRaWAN 디바이스를 AWS IoT Core for LoRaWAN에 연결합니다. 디바이스에서 받는 업링크 메시지에는 다음 정보가 포함됩니다.

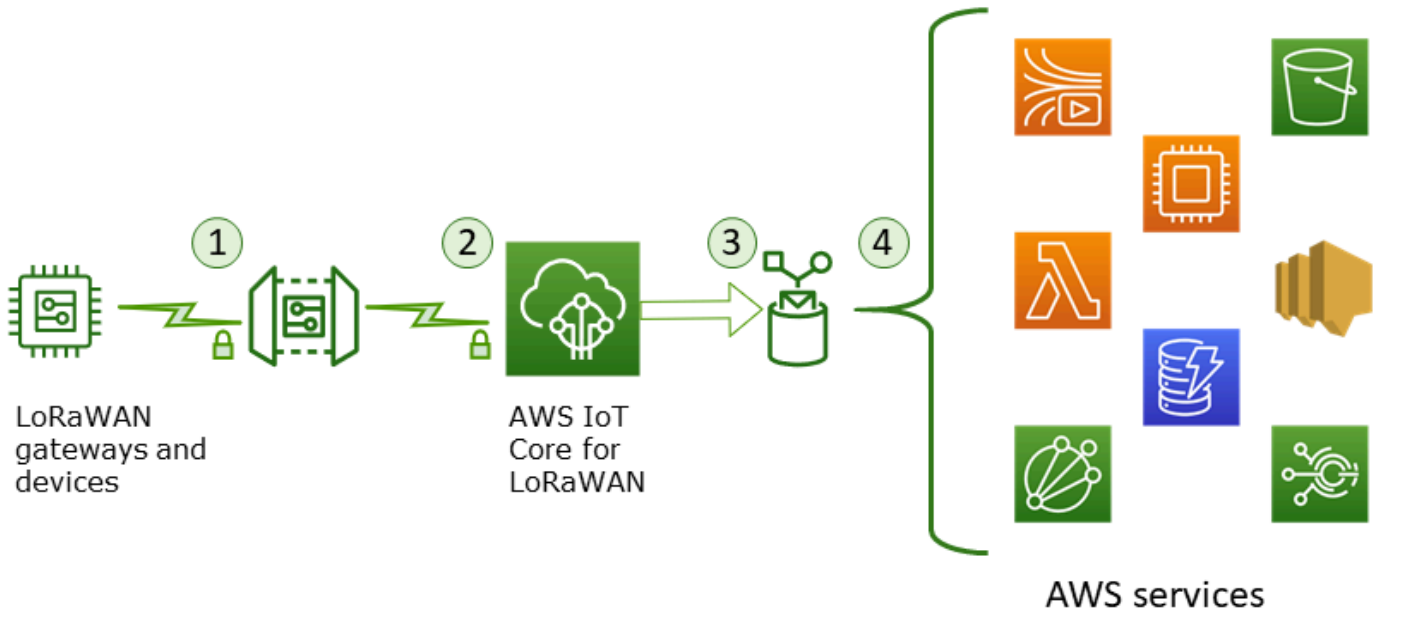
- 무선 디바이스에서 전송되는 암호화된 페이로드 메시지에 해당하는 페이로드 데이터.
- 다음을 포함하는 무선 메타데이터:
  - 디바이스 정보(예: DeveUI, 데이터 속도 및 디바이스가 작동 중인 주파수 채널).
  - 디바이스에 연결된 게이트웨이에 대한 선택적 추가 파라미터 및 게이트웨이 정보. 게이트웨이 파라미터에는 게이트웨이의 EUI, SNR 및 RSSI가 포함됩니다.

  무선 메타데이터를 사용하여 무선 디바이스에 대한 유용한 정보와 디바이스와 AWS IoT 간에 전송되는 데이터를 얻을 수 있습니다. 예를 들어, `AckedMessageId` 파라미터를 사용하여 마지막으로 확인된 다운링크 메시지가 디바이스에서 수신되었는지 여부를 확인할 수 있습니다. 필요에 따라, 게이트웨이 정보를 포함하도록 선택한 경우 디바이스에 더 가까운 더 강력한 게이트웨이 채널로 전환할지 여부를 식별할 수 있습니다.

업링크 메시지를 관찰하는 방법?

디바이스를 온보딩한 후 AWS IoT 콘솔의 테스트 페이지에서 MQTT 테스트 클라이언트를 사용하여 대상을 생성할 때 지정한 주제를 구독할 수 있습니다. 디바이스가 연결되어 페이로드 데이터 전송을 시작하면 메시지가 표시되기 시작합니다.

이 다이어그램은 AWS IoT Core for LoRaWAN에 연결된 LoRaWAN 시스템의 핵심 요소를 식별하며 기본 데이터 영역과 시스템을 통해 데이터가 흐르는 방식을 보여줍니다.



무선 디바이스가 업링크 데이터 전송을 시작하면 AWS IoT Core for LoRaWAN은 무선 메타데이터 정보를 페이로드로 래핑한 다음 이를 AWS 애플리케이션으로 전송합니다.

업링크 메시지 예제

다음 예제에서는 디바이스에서 받은 업링크 메시지의 형식을 보여 줍니다.

```
{
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",
    "PayloadData": "Cc48AAAAAAAAAA=",
    "WirelessMetadata":
    {
        "LoRaWAN":
        {
            "ADR": false,
            "Bandwidth": 125,
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "0",
            "DevAddr": "00b96cd4",
            "DevEui": "58a0cb000202c99",
            "FOptLen": 2,
            "FCnt": 1,
            "Fport": 136,
            "Frequency": "868100000",
            "Gateways": [
```

```
            {
                   "GatewayEui": "80029cfffe5cf1cc",
                   "Snr": -29,
                   "Rssi": 9.75
            }
            ],
           "MIC": "7255cb07",
           "MType": "UnconfirmedDataUp",
           "Major": "LoRaWANR1",
           "Modulation": "LORA",
           "PolarizationInversion": false,
           "SpreadingFactor": 12,
           "Timestamp": "2021-05-03T03:24:29Z"

        }
    }
}
```

업링크 메타데이터에서 게이트웨이 메타데이터 제외

업링크 메타데이터에서 게이트웨이 메타데이터 정보를 제외하려면 서비스 프로파일을 만들 때 AddGwMetadata 파라미터를 비활성화합니다. 파라미터를 비활성화하는 방법에 대한 자세한 내용은 서비스 프로파일 추가을(를) 참조하세요.

이 경우 다음 예제와 같이 업링크 메타데이터에 `Gateways` 섹션이 표시되지 않습니다.

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

대기열에 LoRaWAN 장치로 전송할 다운링크 메시지 추가

클라우드 호스팅 애플리케이션 및 기타 AWS 서비스는 무선 디바이스로 다운링크 메시지를 전송할 수 있습니다. 다운링크 메시지는 AWS IoT Core for LoRaWAN에서 무선 디바이스로 전송한 메시지입니다. AWS IoT Core for LoRaWAN에 온보딩한 각 디바이스에 대해 다운링크 메시지를 예약하고 전송할 수 있습니다.

다운링크 메시지를 전송하려는 디바이스가 여러 개인 경우 멀티캐스트 그룹을 사용할 수 있습니다. 멀티캐스트 그룹의 디바이스는 동일한 멀티캐스트 주소를 공유하며, 이 주소는 전체 수신자 디바이스 그룹에 배포됩니다. 자세한 내용은 멀티캐스트 그룹을 생성하여 여러 디바이스로 다운링크 페이로드를 전송합니다. 단원을 참조하십시오.

다운링크 메시지 대기열 작동 방식

LoRaWAN 디바이스의 디바이스 클래스에 따라 대기열에 있는 메시지가 디바이스로 전송되는 방식이 결정됩니다. 클래스 A 디바이스는 AWS IoT Core for LoRaWAN에 업링크 메시지를 전송하여 디바이스가 다운링크 메시지를 수신할 수 있음을 나타냅니다. 클래스 B 디바이스는 일반 다운링크 슬롯에서 메시지를 수신할 수 있습니다. 클래스 C 디바이스는 언제라도 다운링크 메시지를 수신할 수 있습니다. 디바이스 클래스에 대한 자세한 내용은 디바이스 클래스 섹션을 참조하세요.

다음은 메시지가 대기열에 추가되고 클래스 A 디바이스로 전송되는 방법을 보여줍니다.

1. AWS IoT Core for LoRaWAN은 대기열에 추가된 다운링크 메시지를 AWS IoT 콘솔 또는AWS IoT 무선 API를 사용하여 지정한 프레임 포트, 페이로드 데이터 및 승인 모드 파라미터와 함께 버퍼링합니다.
2. LoRaWAN 디바이스는 업링크 메시지를 전송하여 현재 온라인 상태이고 다운링크 메시지 수신을 시작할 수 있음을 나타냅니다.
3. 대기열에 다운링크 메시지를 두 개 이상 추가한 경우 AWS IoT Core for LoRaWAN은 승인(ACK) 플래그가 설정된 상태로 대기열의 첫 번째 다운링크 메시지를 디바이스로 전송합니다.
4. 디바이스는 업링크 메시지를 즉시 AWS IoT Core for LoRaWAN으로 전송하거나 다음 업링크 메시지가 나타날 때까지 대기하며, 메시지에 ACK 플래그를 포함시킵니다.
5. AWS IoT Core for LoRaWAN에서 ACK 플래그가 있는 업링크 메시지를 수신하면 대기열에서 다운링크 메시지를 삭제하여 디바이스가 다운링크 메시지를 성공적으로 수신했음을 나타냅니다. 세 번 확인한 후에도 업링크 메시지에 ACK 플래그가 없다면 메시지가 폐기됩니다.

콘솔을 사용하여 다운링크 대기열 작업 수행

AWS Management Console을 사용하여 다운링크 메시지를 대기열에 추가하고 필요에 따라 개별 메시지 또는 전체 대기열을 지울 수 있습니다. 클래스 A 디바이스의 경우 디바이스에서 업링크를 수신하여 온라인 상태임을 나타내면 대기열에 있는 메시지가 해당 디바이스로 전송됩니다. 메시지가 전송되면 대기열에서 자동으로 지워집니다.

대기열에 다운링크 메시지 추가

다운링크 메시지 대기열을 생성하려면

1. AWS IoT 콘솔의 디바이스 허브로 이동하여 다운링크 메시지를 대기열에 추가할 디바이스를 선택합니다.
2. 디바이스 세부 정보 페이지의 다운링크 메시지(Downlink messages) 섹션에서 대기열에 다운링크 메시지 추가(Queue downlink messages)를 선택합니다.
3. 다음과 같은 파라미터를 지정하여 다운링크 메시지를 구성합니다.
   - FPort: 디바이스가 AWS IoT Core for LoRaWAN과 통신할 프레임 포트를 선택합니다.
   - 페이로드(Payload): 디바이스로 전송하려는 페이로드 메시지를 지정합니다. 최대 페이로드 크기는 242바이트입니다. 적응형 데이터 속도(ADR)를 활성화한 경우 AWS IoT Core for LoRaWAN은 이 기능을 사용하여 페이로드 크기에 맞는 최적의 데이터 속도를 선택합니다. 필요에 따라 데이터 속도를 추가로 최적화할 수 있습니다.
   - 승인 모드(Acknowledge mode): 디바이스에서 다운링크 메시지를 수신했는지 여부를 확인합니다. 메시지에 이 모드가 필요한 경우 데이터 스트림에 ACK 플래그가 있는 업링크 메시지가 표시되고 메시지가 대기열에서 지워집니다.
4. 다운링크 메시지를 대기열에 추가하려면 제출(Submit)을 선택합니다.

이제 다운링크 메시지가 대기열에 추가되었습니다. 메시지가 표시되지 않거나 오류가 발생하는 경우 다운링크 메시지 대기열 오류 문제 해결에 설명된 대로 오류를 해결할 수 있습니다.

> **Note**
>
> 다운링크 메시지가 대기열에 추가된 후에는 더 이상 FPort, 페이로드(Payload) 및 승인 모드(Acknowledge mode) 파라미터를 편집할 수 없습니다. 이러한 파라미터에 다른 값을 사용하여 다운링크 메시지를 전송하려면 이 메시지를 삭제하고 업데이트된 파라미터 값으로 새 다운링크 메시지를 대기열에 추가할 수 있습니다.

대기열에는 추가한 다운링크 메시지가 나열됩니다. 디바이스와 AWS IoT Core for LoRaWAN 간에 교환되는 업링크 및 다운링크 메시지의 페이로드를 보려면 네트워크 분석기를 사용할 수 있습니다. 자세한 내용은 네트워크 분석기를 사용하여 무선 리소스 플릿 실시간 모니터링 단원을 참조하십시오.

다운링크 메시지 대기열 나열

생성한 다운링크 메시지가 대기열에 추가됩니다. 각 후속 다운링크 메시지는 대기열에서 이 메시지 다음에 추가됩니다. 디바이스 세부 정보 페이지의 다운링크 메시지(Downlink messages) 섹션에서 다운링크 메시지 목록을 볼 수 있습니다. 업링크가 수신되면 메시지가 디바이스로 전송됩니다. 디바이스에서 다운링크 메시지를 수신하면 메시지가 대기열에서 제거됩니다. 이제 다음 번 메시지가 대기열에서 위로 이동하여 디바이스로 전송됩니다.

개별 다운링크 메시지 삭제 또는 전체 대기열 지우기

각 다운링크 메시지는 디바이스로 전송된 후 대기열에서 자동으로 지워집니다. 또한 개별 메시지를 삭제하거나 전체 다운링크 대기열을 지울 수 있습니다. 이러한 작업은 실행 취소할 수 없습니다.

- 대기열에 전송하지 않으려는 메시지가 있는 경우 메시지를 선택하고 삭제(Delete)를 선택합니다.
- 대기열에서 디바이스로 메시지를 전송하지 않으려는 경우 다운링크 대기열 지우기(Clear downlink queue)를 선택하여 전체 대기열을 지울 수 있습니다.

API를 사용하여 다운링크 대기열 작업 수행

AWS IoT 무선 API를 사용하여 다운링크 메시지를 대기열로 전송하고 필요에 따라 개별 메시지 또는 전체 대기열을 지울 수 있습니다.

대기열에 다운링크 메시지 추가

다운링크 메시지 대기열을 생성하려면 `SendDataToWirelessDevice` API 작업 또는`send-data-to-wireless-device` CLI 명령을 사용합니다.

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

이 명령을 실행하면 출력에 다운링크 메시지의 `MessageId`가 생성됩니다. 경우에 따라, `MessageId`를 수신하더라도 패킷이 삭제될 수 있습니다. 이 오류를 해결하는 방법에 대한 자세한 내용은 다운링크 메시지 대기열 오류 문제 해결 섹션을 참조하세요.

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

대기열에 있는 다운링크 메시지 나열

대기열에 있는 모든 다운링크 메시지를 나열하려면 ListQueuedMessages API 작업 또는list-queued-messages CLI 명령을 사용합니다.

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

기본적으로 이 명령을 실행하면 최대 10개의 다운링크 메시지가 표시됩니다.

개별 다운링크 메시지 제거 또는 전체 대기열 지우기

대기열에서 개별 메시지를 제거하거나 전체 대기열을 지우려면 DeleteQueuedMessages API 작업 또는 delete-queued-messages CLI 명령을 사용합니다.

- 개별 메시지를 제거하려면 wirelessDeviceId로 지정된 무선 디바이스에서 제거하려는 메시지의 messageID를 제공합니다.
- 전체 다운링크 대기열을 지우려면 wirelessDeviceId로 지정된 무선 디바이스에 대해 messageID를 *로 지정합니다.

다운링크 메시지 대기열 오류 문제 해결

다음은 예상한 결과가 표시되지 않는 경우에 확인해야 할 몇 가지 사항입니다.

- AWS IoT 콘솔에 다운링크 메시지가 표시되지 않음

  콘솔을 사용하여 다운링크 대기열 작업 수행에 설명된 대로 메시지를 추가했지만 대기열에 다운링크 메시지가 표시되지 않는 경우 사용 중인 디바이스가 활성화 또는 조인 절차라는 프로세스를 완료하지 않았기 때문일 수 있습니다. 이 절차는 디바이스가 AWS IoT Core for LoRaWAN에 온보딩할 때 완료됩니다. 자세한 내용은 콘솔을 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가 단원을 참조하십시오.

  디바이스를 AWS IoT Core for LoRaWAN에 온보딩한 후 네트워크 분석기나 Amazon CloudWatch를 통해 디바이스를 모니터링하여 조인 및 리조인이 성공했는지 여부를 확인할 수 있습니다. 자세한 내용은 모니터링 도구 단원을 참조하십시오.

- API 사용 시 다운링크 메시지 패킷이 누락됨

  `SendDataToWirelessDevice` API 작업을 사용할 때 API는 고유한 `MessageId`를 반환합니다. 그러나 이것만으로는 LoRaWAN 디바이스가 다운링크 메시지를 수신했는지 여부는 확인할 수 없습니다. 예를 들어, 디바이스가 조인 절차를 완료하지 않은 경우 다운링크 패킷이 삭제될 수 있습니다. 이 오류를 해결하는 방법에 대한 자세한 내용은 이전 섹션을 참조하세요.

- 다운링크 메시지를 전송할 때 ARN 누락 오류

  대기열에서 디바이스로 다운링크 메시지를 전송할 때 Amazon 리소스 이름(ARN) 누락 오류가 나타날 수 있습니다. 이 오류는 다운링크 메시지를 수신하는 디바이스에 대해 대상이 올바르게 지정되지 않았기 때문에 발생할 수 있습니다. 이 오류를 해결하려면 디바이스에 대한 대상 세부 정보를 확인합니다.

퍼블릭 LoRaWAN 디바이스 네트워크에서 LoRaWAN 트래픽 관리(Everynet)

공개적으로 사용 가능한 LoRaWAN 네트워크를 사용하여 몇 분 만에 LoRaWAN 디바이스를 클라우드에 연결할 수 있습니다. AWS IoT Core for LoRaWAN은 이제 미국과 영국에서 Everynet의 네트워크 커버리지를 지원합니다. 퍼블릭 네트워크를 사용할 경우 매월 디바이스마다 퍼블릭 네트워크 연결 요금이 부과됩니다. 요금은 퍼블릭 네트워크 연결이 제공되는 모든 AWS 리전에 적용됩니다. 이 기능의 요금에 대한 자세한 내용은 AWS IoT Core 요금 페이지를 참조하세요.

> **Important**
>
> 퍼블릭 네트워크는 Everynet에서 직접 운영하고 서비스로 제공됩니다. 이 기능을 사용하기 전에 해당 AWS 서비스 약관을 확인하세요. 또한 AWS IoT Core for LoRaWAN을 통해 퍼블릭 네트워크를 사용하는 경우 특정 LoRaWAN 디바이스 정보(예: `DevEUI` 및 `JoinEUI`)는 AWS IoT Core for LoRaWAN을 사용할 수 있는 리전 간에 복제됩니다.

AWS IoT Core for LoRaWAN은 LoRaWAN 백엔드 인터페이스 1.0 사양에 설명된 대로 LoRa Alliance 사양에 따라 퍼블릭 LoRaWAN 네트워크를 지원합니다. 퍼블릭 네트워크 기능을 사용하여 홈 네트워크 외부의 최종 디바이스를 연결할 수 있습니다. 이 기능을 지원하기 위해 AWS IoT Core for LoRaWAN은 Everynet과 협력하여 확장된 무선 커버리지를 제공합니다.

퍼블릭 LoRaWAN 네트워크 사용의 이점

LoRaWAN 디바이스는 퍼블릭 네트워크를 사용하여 클라우드에 연결할 수 있으므로 배포 시간이 단축되고 프라이빗 LoRaWAN 네트워크를 유지 관리하는 데 필요한 시간과 비용이 절약됩니다.

퍼블릭 LoRaWAN 네트워크를 사용하면 커버리지 확장, 무선 네트워크 없이 코어 실행, 커버리지 밀도화와 같은 이점을 누릴 수 있습니다. 이 기능을 사용하여 다음을 수행할 수 있습니다.

- 퍼블릭 LoRaWAN 네트워크 지원 아키텍처 섹션에 표시된 그림의 디바이스 A와 같이 디바이스가 홈 네트워크 외부로 이동할 때 해당 디바이스에 커버리지를 제공합니다.
- 퍼블릭 LoRaWAN 네트워크 지원 아키텍처 섹션에 표시된 그림의 디바이스 B와 같이 연결할 LoRa 게이트웨이가 없는 디바이스에 커버리지를 확장합니다. 그러면 디바이스는 파트너가 제공한 게이트웨이를 사용하여 홈 네트워크에 연결할 수 있습니다.

LoRaWAN 디바이스는 퍼블릭 네트워크를 사용하여 로밍 기능을 사용하는 클라우드에 연결할 수 있으므로 배포 시간이 단축되고 프라이빗 LoRaWAN 네트워크를 유지 관리하는 데 필요한 시간과 비용이 절약됩니다.

다음 섹션에서는 퍼블릭 네트워크 지원 아키텍처, 퍼블릭 LoRaWAN 네트워크 지원의 작동 방식 및 이 기능을 사용하는 방법에 대해 설명합니다.

주제

- LoRaWAN 퍼블릭 네트워크 지원 작동 방식
- 퍼블릭 네트워크 지원 사용 방법

LoRaWAN 퍼블릭 네트워크 지원 작동 방식

AWS IoT Core for LoRaWAN은 LoRa Alliance 사양에 따라 패시브 로밍 기능을 지원합니다. 패시브 로밍을 사용하면 엔드 디바이스에 로밍 프로세스가 완전히 투명하게 진행됩니다. 홈 네트워크 외부에서 로밍하는 엔드 디바이스는 네트워크의 게이트웨이에 연결하고 애플리케이션 서버를 사용하여 업링크 및 다운링크 데이터를 교환할 수 있습니다. 디바이스는 전체 로밍 프로세스 동안 홈 네트워크에 연결된 상태를 유지합니다.

> **Note**
>
> AWS IoT Core for LoRaWAN은 패시브 로밍의 스테이트리스 기능만 지원합니다. 핸드오버 로밍은 지원되지 않습니다. 핸드오버 로밍에서는 디바이스가 홈 네트워크 외부로 이동할 때 다른 이동 통신사로 전환됩니다.

주제

- 퍼블릭 LoRaWAN 네트워크 개념

- 퍼블릭 LoRaWAN 네트워크 지원 아키텍처

퍼블릭 LoRaWAN 네트워크 개념

AWS IoT Core for LoRaWAN에서 지원되는 퍼블릭 네트워크 기능은 다음과 같은 개념을 사용합니다.

LoRaWAN network server(LNS)

LNS는 온프레미스에서 실행되거나 클라우드 기반 서비스로 사용될 수 있는 독립 실행형 프라이빗 서버입니다. AWS IoT Core for LoRaWAN은 클라우드에서 서비스를 제공하는 LNS입니다.

홈 네트워크 서버(hNS)

홈 네트워크는 디바이스가 속한 네트워크입니다. 홈 네트워크 서버(hNS)는 AWS IoT Core for LoRaWAN이 디바이스의 프로비저닝 데이터(예: `DevEUI`, `AppEUI`, 세션 키)를 저장하는 LNS입니다.

방문 네트워크 서버(vNS)

방문 네트워크는 디바이스가 홈 네트워크를 벗어날 때 디바이스에 커버리지를 제공하는 네트워크입니다. 방문 네트워크 서버(vNS)는 엔드 디바이스에 서비스를 제공할 수 있도록 hNS와 비즈니스 및 기술 계약을 체결한 LNS입니다. AWS 파트너인 Everynet은 커버리지를 제공하는 방문 네트워크 역할을 합니다.

서빙 네트워크 서버(sNS)

서빙 네트워크 서버(sNS)는 디바이스의 MAC 명령을 처리하는 LNS입니다. 하나의 LoRa 세션에는 sNS가 하나만 있을 수 있습니다.

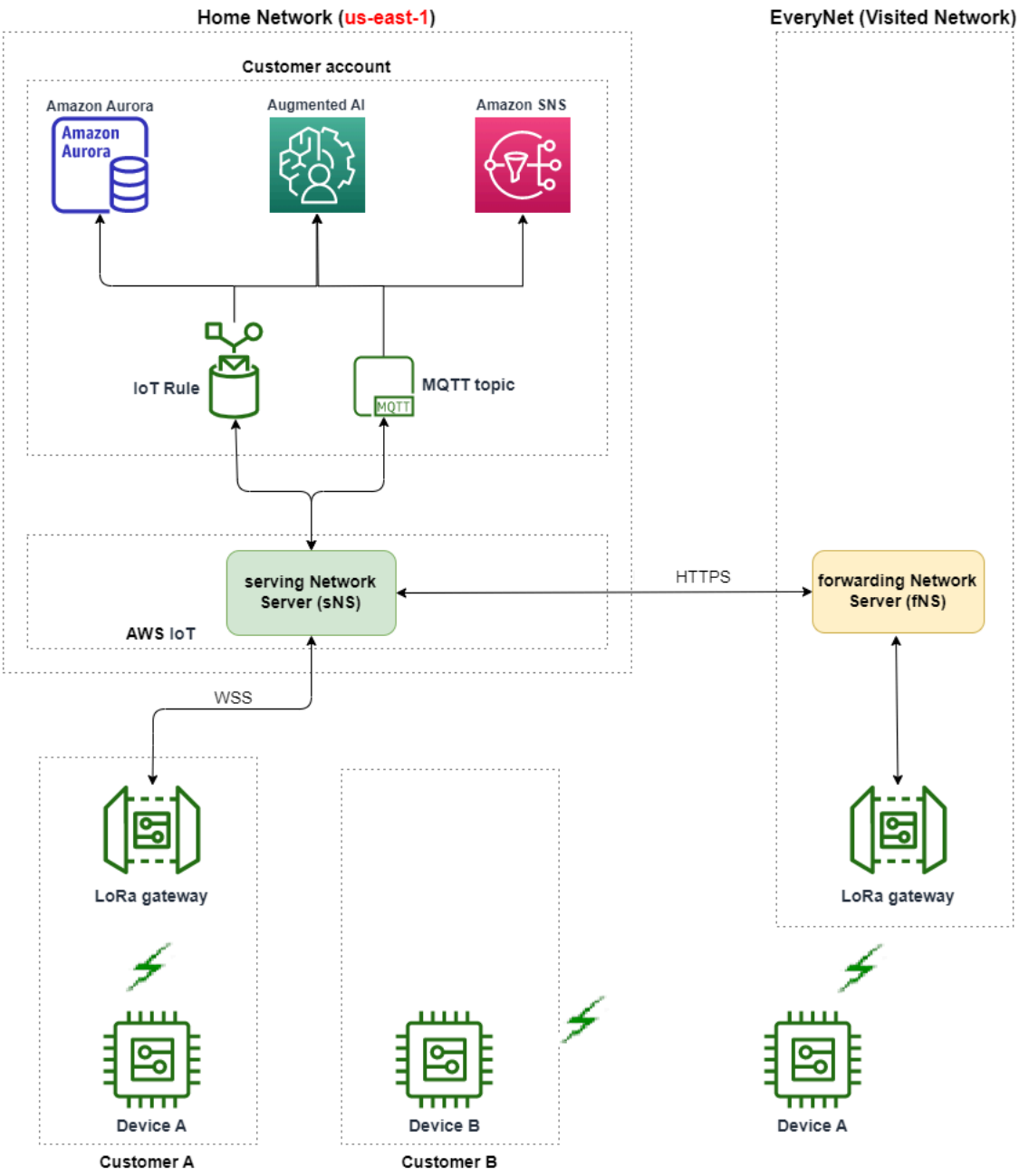
전달 네트워크 서버(fNS)

전달 네트워크 서버(fNS)는 무선 게이트웨이를 관리하는 LNS입니다. 하나의 LoRa 세션에 0개 이상의 fNS가 사용될 수 있습니다. 이 네트워크 서버는 디바이스로부터 수신한 데이터 패킷을 홈 네트워크로 전달하는 작업을 관리합니다.

퍼블릭 LoRaWAN 네트워크 지원 아키텍처

다음 아키텍처 다이어그램은 AWS IoT Core for LoRaWAN이 Everynet과 협력하여 퍼블릭 네트워크 연결을 제공하는 방법을 보여줍니다. 이 경우 디바이스 A는 LoRa 게이트웨이를 통해 AWS IoT Core for LoRaWAN에서 제공되는 홈 네트워크 서버(hNS)에 연결됩니다. 디바이스 A가 홈 네트워크 밖으로 이동하면 방문 네트워크로 들어가고 Everynet에서 제공하는 방문 네트워크 서버(vNS)에서 커버리지를 제공합니다. vNS는 연결할 LoRa 게이트웨이가 없는 디바이스 B까지 커버리지를 확장합니다.

다음 섹션에 설명된 대로 AWS IoT 콘솔에서 퍼블릭 네트워크 커버리지 정보를 볼 수 있습니다.



AWS IoT Core for LoRaWAN은 LoRa Alliance LoRaWAN 로밍 허브 기술 권장 사항에 따라 로밍 허브 기능을 사용합니다. 로밍 허브는 Everynet이 엔드 디바이스로부터 수신한 트래픽을 라우팅할 수 있도록 엔드포인트를 제공합니다. 이 경우 Everynet은 디바이스로부터 수신한 트래픽을 전달하는 전달 네트워크 서버(fNS) 역할을 합니다. LoRa Alliance 사양에 정의된 대로 HTTP RESTful API를 사용합니다.

> **Note**
>
> 디바이스가 홈 네트워크에서 벗어나 홈 네트워크와 Everynet이 모두 커버리지를 제공할 수 있는 위치에 진입하는 경우, 디바이스는 선착순 정책을 사용하여 LoRa 게이트웨이에 연결할지 아니면 Everynet의 게이트웨이에 연결할지를 결정합니다.

퍼블릭 네트워크를 방문하면 hNS와 서빙 네트워크 서버(sNS)가 분리됩니다. 그런 다음 sNS와 hNS 간에 업링크 및 다운링크 패킷이 교환됩니다.

퍼블릭 네트워크 지원 사용 방법

Everynet의 퍼블릭 네트워크 지원을 활성화하려면 서비스 프로필을 만들 때 특정 로밍 파라미터를 지정합니다. 이번 베타 릴리스에서는 AWS IoT 무선 API 또는 AWS CLI를 사용할 때 이러한 파라미터를 사용할 수 있습니다. 다음 단원에서는 활성화해야 하는 파라미터와 AWS CLI를 사용하여 공용 네트워크를 활성화하는 방법을 보여 줍니다.

> **Note**
>
> 새 서비스 프로필을 생성할 때만 퍼블릭 네트워크 지원을 활성화할 수 있습니다. 이러한 파라미터를 사용하여 퍼블릭 네트워크를 활성화하기 위해 기존 프로필을 업데이트할 수는 없습니다.

주제

- 로밍 파라미터
- 디바이스에 대한 퍼블릭 네트워크 지원 활성화

로밍 파라미터

디바이스의 서비스 프로필을 만들 때 다음 파라미터를 지정하세요. AWS IoT 콘솔의 프로필 허브에서 서비스 프로필을 추가할 때 또는 AWS IoT 무선 API 작업, `CreateServiceProfile` 또는 AWS CLI 명령 `create-service-profile`을 사용할 때 이러한 파라미터를 지정합니다.

> **Note**
>
> AWS IoT Core for LoRaWAN은 핸드오버 로밍을 지원하지 않습니다. 서비스 프로필을 만들 때는 핸드오버 로밍 사용 여부를 지정하는 `HrAllowed` 파라미터를 활성화할 수 없습니다.

- 로밍 활성화 허용(`RaAllowed`): 이 파라미터는 로밍 활성화를 사용할지를 지정합니다. 로밍 활성화를 사용하면 vNS 커버리지 내에서 엔드 디바이스를 활성화할 수 있습니다. 로밍 기능을 사용할 때는 `RaAllowed`를 `true`로 설정해야 합니다.
- 패시브 로밍 허용(`PrAllowed`): 이 파라미터는 패시브 로밍을 활성화할지를 지정합니다. 로밍 기능을 사용할 때는 `PrAllowed`를 `true`로 설정해야 합니다.

디바이스에 대한 퍼블릭 네트워크 지원 활성화

디바이스에서 퍼블릭 LoRaWAN 네트워크 지원을 활성화하려면 다음 절차를 실행하세요.

> **Note**
>
> 퍼블릭 네트워크 기능은 OTAA 디바이스에만 활성화할 수 있습니다. ABP를 활성화 방법으로 사용하는 디바이스에는 이 기능이 지원되지 않습니다.

1. 로밍 파라미터를 포함하여 서비스 프로필 생성

   로밍 파라미터를 활성화하여 서비스 프로필을 생성합니다.

   > **Note**
   >
   > 이 서비스 프로필에 연결할 디바이스에 대한 디바이스 프로필을 생성할 때는 `RxDelay1` 파라미터에 2초 이상의 큰 값을 지정하는 것이 좋습니다.

   - AWS IoT 콘솔 사용

     AWS IoT 콘솔의 프로필 허브로 이동하여 서비스 프로필 추가를 선택합니다. 프로필을 만들 때 퍼블릭 네트워크 활성화를 선택합니다.

   - AWS IoT 무선 API 사용

서비스 프로필을 생성할 때 로밍을 활성화하려면 아래 예와 같이 CreateServiceProfile API 작업 또는 create-service-profile CLI 명령을 사용합니다.

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

이 명령을 실행하면 서비스 프로필의 ARN 및 ID가 출력으로 반환됩니다.

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

2. 서비스 프로필에서 로밍 파라미터 확인

지정한 로밍 파라미터를 확인하려면 아래 예와 같이 콘솔에서 또는 get-service-profile CLI 명령을 사용하여 서비스 프로필을 볼 수 있습니다.

- AWS IoT 콘솔 사용

  AWS IoT 콘솔의 프로필 허브로 이동하여 생성한 프로필을 선택합니다. 세부 정보 페이지의 프로필 구성 탭에서 RaAllowed 및 PrAllowed가 true로 설정된 것을 볼 수 있습니다.

- AWS IoT 무선 API 사용

  활성화한 로밍 파라미터를 보려면 아래 예와 같이 GetServiceProfile API 작업 또는 get-service-profile CLI 명령을 사용합니다.

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

이 명령을 실행하면 로밍 파라미터 값인 RaAllowed 및 PrAllowed가 포함된 서비스 프로필 세부 정보가 출력으로 반환됩니다.

```
{
```

```
    "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. 서비스 프로필을 디바이스에 연결

로밍 파라미터를 포함하여 생성한 서비스 프로필을 엔드 디바이스에 연결합니다. 디바이스 프로필을 만들고 무선 디바이스의 대상을 추가할 수도 있습니다. 이 대상을 사용하여 디바이스에서 전송되는 업링크 메시지를 라우팅합니다. 디바이스 프로필 및 대상을 만드는 방법에 대한 자세한 내용은 디바이스 프로파일 추가 및 AWS IoT Core for LoRaWAN에 대상 추가 섹션을 참조하세요.

- 새 디바이스 온보딩

  아직 디바이스를 온보딩하지 않은 경우 디바이스를 AWS IoT Core for LoRaWAN에 추가할 때 이 서비스 프로필을 사용하도록 지정합니다. 다음 명령은 생성한 서비스 프로필의 ID를 사용하여 디바이스를 추가할 때 `create-wireless-device` CLI 명령을 사용하는 방법을 보여줍니다. 콘솔을 사용하여 서비스 프로필을 추가하는 방법에 대한 자세한 내용은 콘솔을 사용하여 AWS IoT Core for LoRaWAN에 무선 디바이스 사양 추가 단원을 참조하세요.

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

  다음은 *createdevice.json* 파일의 콘텐츠를 보여줍니다.

createdevice.json의 내용

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

이 명령을 실행하여 출력하면 무선 디바이스의 ARN과 ID가 출력으로 생성됩니다.

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

- 기존 디바이스 업데이트

  디바이스를 이미 온보딩한 경우 이 서비스 프로필을 사용하도록 기존 무선 디바이스를 업데이트할 수 있습니다. 다음 명령은 생성한 서비스 프로필의 ID를 사용하여 디바이스를 업데이트할 때 `update-wireless-device` CLI 명령을 사용하는 방법을 보여줍니다.

  ```
  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"
  ```

  이 명령은 출력을 생성하지 않습니다. `GetWirelessDevice` API 또는 `get-wireless-device` CLI 명령을 사용하여 업데이트된 정보를 가져올 수 있습니다.

4. Everynet을 사용하여 디바이스를 클라우드에 연결

   로밍이 활성화되었으므로 이제 디바이스에서 조인을 수행해야 새로운 `DevAddr`을 얻을 수 있습니다. OTAA를 사용하면 LoRaWAN 디바이스가 조인 요청을 전송하고 네트워크 서버에서 요청을 허용할 수 있습니다. 그러면 Everynet에서 제공하는 네트워크 커버리지를 사용하여 AWS 클라우드에 연결할 수 있습니다. 활성화 절차를 수행하거나 디바이스를 조인하는 방법에 대한 지침은 디바이스 설명서를 참조하세요.

   > **Note**
   >
   > - OTAA를 활성화 방법으로 사용하는 디바이스의 경우에만 로밍 기능을 활성화하고 퍼블릭 네트워크에 연결할 수 있습니다. ABP 디바이스는 지원되지 않습니다. 활성화 절차를 수행하거나 디바이스를 조인하는 방법에 대한 지침은 디바이스 설명서를 참조하세요. 활성화 모드 섹션을 참조하세요.
   > - 디바이스의 로밍 기능을 비활성화하려면 이 서비스 프로필에서 디바이스를 분리한 다음 로밍 파라미터가 `false`로 설정된 다른 서비스 프로필에 연결할 수 있습니다. 이 서비스 프로필로 전환한 후에는 디바이스가 퍼블릭 네트워크에서 계속 실행되지 않도록 또 다른 조인을 수행해야 합니다.

5. 업링크 및 다운링크 메시지 교환

   디바이스가 AWS IoT Core for LoRaWAN에 조인되면 디바이스와 클라우드 간에 메시지 교환을 시작할 수 있습니다.

   - 업링크 메시지 보기

     디바이스에서 업링크 메시지를 보내는 경우 AWS IoT Core for LoRaWAN은 이전에 구성된 대상을 사용하여 이 메시지를 AWS 계정에 전송합니다. 이러한 메시지는 Everynet 네트워크를 통해 디바이스에서 클라우드로 전송됩니다.

     AWS IoT 규칙 이름을 사용하여 메시지를 보거나 MQTT 클라이언트를 사용하여 대상을 만들 때 지정한 MQTT 주제를 구독할 수 있습니다. 지정하는 규칙 이름 및 기타 대상 세부 정보에 대한 자세한 내용은 콘솔을 사용하여 대상 추가 섹션을 참조하세요.

     업링크 메시지 및 형식 보기에 대한 자세한 정보는 LoRaWAN 디바이스에서 전송된 업링크 메시지 형식 보기 섹션을 참조하세요.

   - 다운링크 메시지 전송

콘솔에서 또는 AWS IoT 무선 API 명령 SendDataToWirelessDevice 또는 AWS CLI 명령 send-data-to-wireless-device를 사용하여 다운링크 메시지를 대기열에 넣고 디바이스에 전송할 수 있습니다. 다운링크 메시지를 대기열에 넣고 전송하는 방법에 대한 자세한 내용은 대기열에 LoRaWAN 장치로 전송할 다운링크 메시지 추가 섹션을 참조하세요.

다음 코드는 send-data-to-wireless-device CLI 명령을 사용하여 다운링크 메시지를 보내는 방법의 예를 보여줍니다. 데이터를 수신할 무선 디바이스의 ID, 페이로드, 승인 모드 사용 여부, 무선 메타데이터를 지정합니다.

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

이 명령을 실행하면 출력에 다운링크 메시지의 MessageId가 생성됩니다.

> **Note**
>
> 경우에 따라, MessageId를 수신하더라도 패킷이 삭제될 수 있습니다. 이러한 시나리오의 문제 해결에 대한 자세한 내용은 다운링크 메시지 대기열 오류 문제 해결 섹션을 참조하세요.

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

- 커버리지 정보 보기

퍼블릭 네트워크를 활성화한 후 AWS IoT 콘솔에서 네트워크 커버리지 정보를 볼 수 있습니다. AWS IoT 콘솔의 커버리지 허브로 이동한 다음 위치를 검색하면 지도에서 디바이스의 커버리지 정보를 확인할 수 있습니다.

> **Note**
>
> 이 기능은 Amazon Location Service를 사용하여 디바이스의 커버리지 정보를 Amazon Location 지도에 표시합니다. Amazon Location 지도를 사용하기 전에 Amazon Location

Service 이용 약관을 검토하세요. AWS는 사용자가 선택한 타사 데이터 공급자에게 사용자의 API 쿼리를 전송할 수 있으며, 이 타사 데이터 공급자는 현재 사용자가 사용하는 AWS 리전 외부에 있을 수도 있습니다. 자세한 내용은 AWS서비스 약관을 참조하세요.

LoRaWan 디바이스 및 멀티캐스트 그룹에 대한 무선 펌웨어 업데이트(FUOTA) 수행

무선 펌웨어 업데이트를 수행하여 단일 LoRaWAN 디바이스 또는 디바이스 그룹의 디바이스 펌웨어를 업데이트할 수 있습니다. 디바이스 펌웨어를 업데이트하거나 다운링크 페이로드를 여러 디바이스로 전송하려면 멀티캐스트 그룹을 생성합니다. 소스는 멀티캐스트를 사용하여 단일 멀티캐스트 그룹으로 데이터를 전송할 수 있으며, 이 데이터는 수신자 디바이스 그룹에 배포됩니다.

FUOTA 및 멀티캐스트 그룹에 대한 AWS IoT Core for LoRaWAN의 지원은 LoRa Alliance의 다음 사양을 기반으로 합니다.

- LoRaWAN 원격 멀티캐스트 설정 사양, TS005-2.0.0
- LoRaWAN 단편화된 데이터 블록 전송 사양, TS004-2.0.0
- LoRaWAN 애플리케이션 레이어 클록 동기화 사양, TS003-2.0.0

> **Note**
>
> AWS IoT Core for LoRaWAN은 LoRa Alliance 사양에 따라 클록 동기화를 자동으로 수행합니다. `AppTimeReq` 함수로 ClockSync 시그널링을 사용하여 요청하는 디바이스에 서버 측 시간을 회신합니다.

멀티캐스트 그룹을 생성하고 FUOTA를 수행하는 방법은 다음 주제에 나와 있습니다.

주제

- 멀티캐스트 및 FUOTA 구성을 위한 디바이스 준비
- 멀티캐스트 그룹을 생성하여 여러 디바이스로 다운링크 페이로드를 전송합니다.
- AWS IoT Core for LoRaWAN 디바이스의 펌웨어 무선 업데이트(FUOTA)

멀티캐스트 및 FUOTA 구성을 위한 디바이스 준비

무선 디바이스를 AWS IoT Core for LoRaWAN에 추가하면 콘솔이나 CLI를 사용하여 멀티캐스트 설정 및 FUOTA 구성을 위해 무선 디바이스를 준비할 수 있습니다. 이 구성을 처음 수행하는 경우 콘솔을 사용하는 것이 좋습니다. 멀티캐스트 그룹을 관리하고 그룹에서 여러 디바이스를 추가하거나 제거하려면 CLI를 사용하여 많은 리소스를 관리하는 것이 좋습니다.

GenAppKey 및 FPorts

무선 디바이스를 추가할 때 디바이스를 멀티캐스트 그룹에 추가하거나 FUOTA를 수행하기 전에 다음 파라미터를 구성합니다. 이러한 파라미터를 구성하기 전에 디바이스가 FUOTA 및 멀티캐스트를 지원하고 무선 디바이스 사양이 `OTAA v1.1` 또는 `OTAAv1.0.x`인지 확인합니다.

- `GenAppKey`: LoRaWAN 버전 1.0.x를 지원하고 멀티캐스트 그룹을 사용하는 디바이스의 경우 `GenAppKey`는 멀티캐스트 그룹의 세션 키가 파생되는 디바이스별 루트 키입니다.

  > **Note**
  >
  > 무선 사양 `OTAA v1.1`을 사용하는 LoRaWAN 디바이스의 경우 `AppKey`는 `GenAppKey`와 동일한 용도로 사용됩니다.

  데이터 전송을 시작하도록 파라미터를 설정하기 위해 AWS IoT Core for LoRaWAN는 최종 디바이스와 세션 키를 배포합니다. LoRaWAN 버전에 대한 자세한 정보는 LoRaWAN 버전 섹션을 참조하세요.

  > **Note**
  >
  > AWS IoT Core for LoRaWAN은 사용자가 제공하는 `GenAppKey` 정보를 암호화된 형식으로 저장합니다.

- `FPorts`: FUOTA 및 멀티캐스트 그룹에 대한 LoRaWAN 사양에 따라 AWS IoT Core for LoRaWAN은 `FPorts` 파라미터의 다음 필드에 기본값을 할당합니다. 다음 `FPort` 값 중 하나를 이미 지정한 경우 1에서 223까지 사용 가능한 다른 값을 선택할 수 있습니다.
  - `Multicast`: 200

    이 `FPort` 값은 멀티캐스트 그룹에 사용됩니다.
  - FUOTA: 201

이 FPort 값은 FUOTA에 사용됩니다.

- ClockSync: 202

  이 FPort 값은 클록 동기화에 사용됩니다.

멀티캐스트 및 FUOTA용 디바이스 프로파일

멀티캐스트 세션이 시작될 때 클래스 B 또는 클래스 C 배포 기간은 다운링크 메시지를 그룹의 디바이스로 전송하는 데 사용됩니다. 멀티캐스트 및 FUOTA용으로 추가하는 디바이스는 클래스 B 또는 클래스 C 작동 모드를 지원해야 합니다. 디바이스가 지원하는 디바이스 클래스에 따라 클래스 B 또는 클래스 C 모드 중 하나 또는 둘 모두가 사용되는 디바이스의 디바이스 프로파일을 선택합니다.

디바이스 프로파일에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 프로파일 추가 섹션을 참조하세요.

콘솔을 사용하여 멀티캐스트 및 FUOTA용 디바이스 준비

콘솔을 사용하여 멀티캐스트 설정 및 FUOTA에 대한 FPorts 및 GenAppKey 파라미터를 지정하려면

1. AWS IoT 콘솔의 디바이스 허브로 이동하고 무선 디바이스 추가(Add wireless device)를 선택합니다.
2. 무선 디바이스 사양(Wireless device specification)을 선택합니다. 디바이스 사용을 위해 디바이스에서 OTAA를 사용해야 합니다. OTAA v1.0.x 또는 OTAA v1.1을 선택하면 FUOTA 구성-옵션(FUOTA configuration-Optional) 섹션이 나타납니다.
3. 무선 디바이스의 확장 고유 식별자(EUI) 파라미터를 입력합니다.
4. FUOTA 구성 - 옵션(FUOTA configuration-Optional) 섹션을 확장한 다음 이 디바이스는 펌웨어 무선 업데이트(FUOTA)를 지원함(This device supports firmware updates over the air (FUOTA))을 선택합니다. 이제 멀티캐스트, FUOTA 및 시계 동기화에 대한 FPort 값을 입력할 수 있습니다. 무선 디바이스 사양으로 OTAA v1.0.x를 선택한 경우 GenAppKey를 입력합니다.
5. 프로파일과 메시지 라우팅 대상을 선택하여 디바이스를 AWS IoT Core for LoRaWAN에 추가합니다. 디바이스에 연결된 디바이스 프로파일의 경우 클래스 B 지원(Supports Class B) 또는 클래스 C 지원(Supports Class C) 모드 중 하나 또는 둘 다를 선택해야 합니다.

> **Note**
>
> FUOTA 구성 파라미터를 지정하려면 AWS IoT 콘솔의 디바이스 허브를 사용해야 합니다. AWS IoT 콘솔의 소개(Intro) 페이지를 사용하여 디바이스를 온보딩하는 경우 이러한 파라미터가 나타나지 않습니다.

무선 디바이스 사양 및 디바이스 온보딩에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 무선 디바이스 추가 섹션을 참조하세요.

> **Note**
>
> 무선 디바이스를 생성할 때만 이러한 파라미터를 지정할 수 있습니다. 기존 디바이스를 업데이트할 때는 파라미터를 변경하거나 지정할 수 없습니다.

API 작업을 사용하여 멀티캐스트 및 FUOTA용 디바이스 준비

멀티캐스트 그룹을 사용하거나 FUOTA를 수행하려면 `CreateWirelessDevice` API 작업 또는 `create-wireless-device` CLI 명령을 사용하여 이러한 파라미터를 구성합니다. 애플리케이션 키 및 fPort 파라미터를 지정하는 것 외에도 디바이스에 연결된 디바이스 프로파일이 클래스 B 또는 클래스 C 모드를 하나 또는 둘 다 지원하는지 확인합니다.

`input.json` 파일을 `create-wireless-device` 명령에 대한 입력으로 제공할 수 있습니다.

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

여기서 각 항목은 다음과 같습니다.

input.json 내용

```
{
    "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
}
```

사용할 수 있는 CLI 명령에 대한 자세한 내용은 AWS CLI 참조를 참조하세요.

> **Note**
>
> 이러한 파라미터의 값을 지정한 후에는 `UpdateWirelessDevice` API 작업을 사용하여 업데이트할 수 없습니다. 대신 `GenAppKey` 및 `FPorts` 파라미터 값을 사용하여 새 디바이스를 생성할 수 있습니다.

이러한 파라미터에 대해 지정된 값에 대한 정보를 얻으려면 `GetWirelessDevice` API 작업 또는 `get-wireless-device` CLI 명령을 사용합니다.

다음 단계

파라미터를 구성한 후 멀티캐스트 그룹 및 FUOTA 태스크를 생성하여 다운링크 페이로드를 전송하거나 LoRaWAN 디바이스의 펌웨어를 업데이트할 수 있습니다.

- 멀티캐스트 그룹 생성에 대한 자세한 내용은 멀티캐스트 그룹 생성 및 그룹에 디바이스 추가 섹션을 참조하세요.
- FUOTA 태스크 생성에 대한 자세한 내용은 FUOTA 태스크 생성 및 펌웨어 이미지 제공 섹션을 참조하세요.

멀티캐스트 그룹을 생성하여 여러 디바이스로 다운링크 페이로드를 전송합니다.

다운링크 페이로드를 여러 디바이스로 전송하려면 멀티캐스트 그룹을 생성합니다. 소스는 멀티캐스트를 사용하여 단일 멀티캐스트 주소로 데이터를 전송할 수 있으며, 이 주소는 전체 수신자 디바이스 그룹에 배포됩니다.

멀티캐스트 그룹의 디바이스는 동일한 멀티캐스트 주소, 세션 키 및 프레임 카운터를 공유합니다. 동일한 세션 키를 사용하여 멀티캐스트 그룹의 디바이스는 다운링크 전송이 시작될 때 메시지를 복호화할 수 있습니다. 멀티캐스트 그룹은 다운링크만 지원합니다. 디바이스에서 다운링크 페이로드가 수신했는지 여부는 확인하지 않습니다.

AWS IoT Core for LoRaWAN의 멀티캐스트 그룹으로 다음을 수행할 수 있습니다.

- 디바이스 프로파일, RFRegion 또는 디바이스 클래스를 사용하여 디바이스 목록을 필터링한 다음 이러한 디바이스를 멀티캐스트 그룹에 추가합니다.
- 48시간 배포 기간 내에 하나 이상의 다운링크 페이로드 메시지를 예약하고 멀티캐스트 그룹의 디바이스로 전송합니다.
- 다운링크 메시지를 수신하기 위해 멀티캐스트 세션이 시작될 때 디바이스가 클래스 B 또는 클래스 C 모드로 일시적으로 전환되도록 합니다.
- 멀티캐스트 그룹 설정과 해당 디바이스의 상태를 모니터링하고 문제도 해결합니다.
- 펌웨어 무선 업데이트(FUOTA)를 사용하여 멀티캐스트 그룹의 디바이스에 펌웨어 업데이트를 안전하게 배포합니다.

다음 동영상은 AWS IoT Core for LoRaWAN 멀티캐스트 그룹을 생성하는 방법에 대해 설명하며 그룹에 디바이스를 추가하고 그룹에 다운링크 메시지를 예약하는 프로세스를 안내합니다.

다음은 멀티캐스트 그룹을 생성하고 다운링크 메시지를 예약하는 방법을 보여줍니다.

주제

- 멀티캐스트 그룹 생성 및 그룹에 디바이스 추가
- 멀티캐스트 그룹 및 그룹의 디바이스 상태 모니터링 및 문제 해결
- 멀티캐스트 그룹의 디바이스로 전송하도록 다운링크 메시지 예약

멀티캐스트 그룹 생성 및 그룹에 디바이스 추가

콘솔 또는 CLI를 사용하여 멀티캐스트 그룹을 생성할 수 있습니다. 멀티캐스트 그룹을 처음 생성하는 경우 콘솔을 사용하여 멀티캐스트 그룹을 추가하는 것이 좋습니다. 멀티캐스트 그룹을 관리하고 그룹에서 디바이스를 추가하거나 제거하려는 경우 CLI를 사용합니다.

추가한 종단 디바이스와 신호를 주고받은 후 AWS IoT Core for LoRaWAN은 종단 디바이스와 공유 키를 설정하고 데이터 전송을 위한 파라미터를 설정합니다.

필수 조건

멀티캐스트 그룹을 생성하고 그룹에 디바이스를 추가하려면 다음을 수행합니다.

- FUOTA 구성 파라미터 `GenAppKey` 및 `FPorts`를 지정하여 멀티캐스트 및 FUOTA 설정을 위해 디바이스를 준비합니다. 자세한 내용은 멀티캐스트 및 FUOTA 구성을 위한 디바이스 준비 단원을 참조하십시오.
- 디바이스가 클래스 B 또는 클래스 C 작동 모드를 지원하는지 확인합니다. 디바이스가 지원하는 디바이스 클래스에 따라 클래스 B 지원(Supports Class B) 또는 클래스 C 지원(Supports Class C) 모드 중 하나 또는 둘 모두가 사용되는 디바이스 프로파일을 선택합니다. 디바이스 프로파일에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 프로파일 추가 섹션을 참조하세요.

  멀티캐스트 세션이 시작될 때 클래스 B 또는 클래스 C 배포 기간은 다운링크 메시지를 그룹의 디바이스로 전송하는 데 사용됩니다.

콘솔을 사용하여 멀티캐스트 그룹 생성

콘솔을 사용하여 멀티캐스트 그룹을 생성하려면 AWS IoT 콘솔의 멀티캐스트 그룹(Multicast groups) 페이지로 이동하고 멀티캐스트 그룹 생성(Create multicast group)을 선택합니다.

1. 멀티캐스트 그룹 생성

   멀티캐스트 그룹을 생성하려면 그룹에 대한 멀티캐스트 속성과 태그를 지정합니다.

   1. 멀티캐스트 속성 지정

      멀티캐스트 속성을 지정하려면 멀티캐스트 그룹에 대해 다음 정보를 입력합니다.

      - 이름(Name): 멀티캐스트 그룹의 고유한 이름을 입력합니다. 이름에는 문자, 숫자, 하이픈 및 밑줄만 포함되어야 합니다. 공백은 포함할 수 없습니다.
      - 설명(Description): 멀티캐스트 그룹에 대한 설명(선택 사항)을 제공할 수 있습니다. 설명 길이는 최대 2,048자입니다.

2. 멀티캐스트 그룹용 태그

   선택적으로 모든 키-값 페어를 멀티캐스트 그룹에 대한 태그(Tags)로 제공할 수 있습니다. 멀티캐스트 그룹 생성을 계속하려면 다음(Next)을 선택합니다.

2. 멀티캐스트 그룹에 디바이스 추가

   멀티캐스트 그룹에 개별 디바이스나 디바이스 그룹을 추가할 수 있습니다. 디바이스를 추가하려면

   1. RFRegion 지정

      멀티캐스트 그룹에 대한 RFRegion 또는 주파수 대역을 지정합니다. 멀티캐스트 그룹의 RFRegion은 멀티캐스트 그룹에 추가하는 디바이스의 RFRegion과 일치해야 합니다. RFRegion에 대한 자세한 내용은 게이트웨이 및 디바이스 연결을 위한 LoRa 주파수 대역 선택 고려 섹션을 참조하세요.

   2. 멀티캐스트 디바이스 클래스 선택

      멀티캐스트 세션 시작 시 멀티캐스트 그룹의 디바이스를 클래스 B 또는 클래스 C 모드로 전환할지 여부를 선택합니다. 클래스 B 세션은 일반 다운링크 슬롯에서 다운링크 메시지를 수신할 수 있으며 클래스 C 세션은 언제든지 다운링크 메시지를 수신할 수 있습니다.

   3. 그룹에 추가할 디바이스를 선택합니다.

      멀티캐스트 그룹에 디바이스를 개별적으로 추가할지 일괄적으로 추가할지 선택합니다.

      - 디바이스를 개별적으로 추가하려면 그룹에 추가할 각 디바이스의 무선 디바이스 ID를 입력합니다.
      - 디바이스를 일괄 추가하려면 디바이스 프로파일 또는 태그별로 추가하려는 디바이스를 필터링합니다. 디바이스 프로파일의 경우 클래스 B, 클래스 C 또는 두 디바이스 클래스를 모두 지원하는 프로파일이 있는 디바이스를 추가할 수 있습니다.

   4. 멀티캐스트 그룹을 생성하려면 생성(Create)을 선택합니다.

      멀티캐스트 그룹 세부 정보와 추가한 디바이스가 그룹에 나타납니다. 멀티캐스트 그룹 및 디바이스의 상태 및 문제 해결에 대한 자세한 내용은 멀티캐스트 그룹 및 그룹의 디바이스 상태 모니터링 및 문제 해결 섹션을 참조하세요.

멀티캐스트 그룹을 생성한 후 작업(Action)을 선택하여 디바이스를 편집 또는 삭제하거나 멀티캐스트 그룹에 추가할 수 있습니다. 디바이스를 추가한 후에는 다운링크 페이로드가 그룹의 디바이스로 전송되도록 세션을 예약할 수 있습니다.

API를 사용하여 멀티캐스트 그룹 생성

API를 사용하여 멀티캐스트 그룹을 생성하고 그룹에 디바이스를 추가하려면

1. 멀티캐스트 그룹 생성

   멀티캐스트 그룹을 생성하려면 CreateMulticastGroup API 작업 또는create-multicast-group CLI 명령을 사용합니다. input.json 파일을 create-multicast-group 명령에 대한 입력으로 제공할 수 있습니다.

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

   여기서 각 항목은 다음과 같습니다.

   input.json 내용

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

   멀티캐스트 그룹을 생성한 후 다음 API 작업 또는 CLI 명령을 사용하여 멀티캐스트 그룹에 대한 정보를 업데이트, 삭제 또는 가져올 수 있습니다.

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

2. 멀티캐스트 그룹에 디바이스 추가

   멀티캐스트 그룹에 디바이스를 개별적으로 또는 일괄적으로 추가할 수 있습니다.

   - 멀티캐스트 그룹에 디바이스를 일괄적으로 추가하려면 StartBulkAssociateWirelessDeviceWithMulticastGroup API 작업 또는start-

bulk-associate-wireless-device-with-multicast-group CLI 명령을 사용합니다. 멀티캐스트 그룹에 일괄적으로 연결하려는 디바이스를 필터링하려면 쿼리 문자열을 제공합니다. 다음은 지정된 ID가 연결된 디바이스 프로파일이 있는 디바이스 그룹을 추가하는 방법을 보여줍니다.

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

여기서 각 항목은 다음과 같습니다.

input.json 내용

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

여기서 `multicast-groups/d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf/bulk`는 디바이스를 그룹과 연결하는 데 사용되는 URL입니다.

- 멀티캐스트 그룹에 디바이스를 개별적으로 추가하려면 AssociateWirelessDeviceWithMulticastGroup API 작업 또는 associate-wireless-device-with-multicast-group CLI를 사용합니다. 그룹에 추가할 각 디바이스의 무선 디바이스 ID를 입력합니다.

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

멀티캐스트 그룹을 생성한 후 다음 API 작업 또는 CLI 명령을 사용하여 멀티캐스트 그룹에 대한 정보를 보거나 장치 연결을 해제할 수 있습니다.

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

> **Note**
>
> ListWirelessDevices API 작업은 일반적으로 무선 디바이스 및 멀티캐스트 그룹 또는 FUOTA 태스크와 연결된 무선 디바이스를 나열하는 데 사용할 수 있습니다.
>
> - 멀티캐스트 그룹과 연결된 무선 디바이스를 나열하려면 MulticastGroupID를 필터로 하여 ListWirelessDevices API 작업을 사용합니다.
> - FUOTA 태스크와 연결된 무선 장치를 나열하려면 ListWirelessDevices를 필터로 하여 FuotaTaskID API 작업을 사용합니다.

다음 단계

멀티캐스트 그룹을 생성하고 디바이스를 추가한 후 계속해서 디바이스를 추가하고 멀티캐스트 그룹과 디바이스의 상태를 모니터링할 수 있습니다. 디바이스가 그룹에 성공적으로 추가되었으면 다운링크 메시지를 구성하고 디바이스로 전송하도록 예약할 수 있습니다. 다운링크 메시지를 전송하려면 먼저 디바이스가 멀티캐스트 설정 준비 상태(Multicast setup ready)여야 합니다. 다운링크 메시지를 예약하면 상태가 세션 시도 중(Session attempting)으로 변경됩니다. 자세한 내용은 멀티캐스트 그룹의 디바이스로 전송하도록 다운링크 메시지 예약 단원을 참조하십시오.

멀티캐스트 그룹에 있는 디바이스의 펌웨어를 업데이트하려면 AWS IoT Core for LoRaWAN을 사용하여 펌웨어 무선 업데이트(FUOTA)를 수행할 수 있습니다. 자세한 내용은 AWS IoT Core for LoRaWAN 디바이스의 펌웨어 무선 업데이트(FUOTA) 단원을 참조하십시오.

디바이스가 추가되지 않았거나 멀티캐스트 그룹 또는 디바이스 상태에 오류가 표시되는 경우 오류 위로 마우스를 가져가 자세한 정보를 보고 해결할 수 있습니다. 여전히 오류가 표시되는 경우 문제를 해결하는 방법에 대한 자세한 내용은 멀티캐스트 그룹 및 그룹의 디바이스 상태 모니터링 및 문제 해결 섹션을 참조하세요.

멀티캐스트 그룹 및 그룹의 디바이스 상태 모니터링 및 문제 해결

디바이스를 추가하고 멀티캐스트 그룹을 만든 후 AWS Management Console을 엽니다. AWS IoT 콘솔의 멀티캐스트 그룹(Multicast groups) 페이지로 이동하고 생성한 멀티캐스트 그룹을 선택하여 세부 정보를 봅니다. 멀티캐스트 그룹 정보, 추가된 디바이스 수 및 디바이스 상태 세부 정보가 표시됩니다. 상태 정보를 사용하여 멀티캐스트 세션의 진행 상황을 추적하고 오류를 해결할 수 있습니다.

멀티캐스트 그룹 상태

AWS Management Console에 멀티캐스트 그룹에 대한 다음 상태 메시지 중 하나가 표시될 수 있습니다.

- 보류중

  이 상태는 멀티캐스트 그룹을 생성했지만 아직 멀티캐스트 세션이 없음을 나타냅니다. 그룹이 생성되면 이 상태 메시지가 표시됩니다. 이 시간 동안 멀티캐스트 그룹을 업데이트하고 디바이스를 그룹과 연결 또는 연결 해제할 수 있습니다. 상태가 보류 중에서 변경된 후에는 추가 디바이스를 그룹에 추가할 수 없습니다.

- 세션 시도 중(Session attempting)

  디바이스가 멀티캐스트 그룹에 성공적으로 추가된 후 그룹에 예약된 멀티캐스트 세션이 있으면 이 상태 메시지가 표시됩니다. 이 시간 동안에는 디바이스를 업데이트하거나 멀티캐스트 그룹에 추가할 수 없습니다. 멀티캐스트 세션을 취소하면 그룹 상태가 보류 중(Pending)으로 변경됩니다.

- 세션 중(In session)

  멀티캐스트 세션의 가장 빠른 세션 시간인 경우 이 상태 메시지가 표시됩니다. 또한 멀티캐스트 그룹은 펌웨어 업데이트 세션이 진행 중인 FUOTA 태스크와 연결될 때 이 상태를 계속 유지합니다.

  세션에 연결된 FUOTA 태스크가 없고 세션 시간이 시간 제한을 초과했거나 멀티캐스트 세션을 취소하여 멀티캐스트 세션이 취소된 경우 그룹 상태가 보류 중(Pending)으로 변경됩니다.

- 삭제 대기 중(Delete waiting)

  멀티캐스트 그룹을 삭제하면 해당 그룹 상태가 삭제 대기 중(Delete waiting)으로 변경됩니다. 삭제는 영구적이며 취소할 수 없습니다. 이 작업은 시간이 걸릴 수 있으며 멀티캐스트 그룹이 삭제될 때까지 그룹 상태는 Delete_Waiting이 됩니다. 멀티캐스트 그룹이 이 상태가 된 후에는 다른 상태 중 하나로 전환할 수 없습니다.

멀티캐스트 그룹의 디바이스 상태

AWS Management Console에 멀티캐스트 그룹의 디바이스에 대한 다음 상태 메시지 중 하나가 표시될 수 있습니다. 각 상태 메시지 위로 마우스를 가져가서 표시 내용에 대한 자세한 정보를 볼 수 있습니다.

- 패키지 시도 중(Package attempting)

  치가 멀티캐스트 그룹과 연결되면 디바이스 상태는 패키지 시도 중(Package attempting)입니다. 이 상태는 AWS IoT Core for LoRaWAN이 디바이스가 멀티캐스트 설정 및 작동을 지원하는지 여부를 아직 확인하지 않았음을 나타냅니다.

- 패키지 지원되지 않음(Package unsupported)

  디바이스가 멀티캐스트 그룹과 연결되면 AWS IoT Core for LoRaWAN은 디바이스의 펌웨어가 멀티캐스트 설정 및 작동이 가능한지 여부를 확인합니다. 디바이스에 지원되는 멀티캐스트 패키지가 없는 경우 상태는 패키지 지원되지 않음(Package unsupported)입니다. 오류를 해결하려면 디바이스의 펌웨어가 멀티캐스트 설정 및 작동이 가능한지 확인합니다.

- 멀티캐스트 설정 시도 중(Multicast setup attempting)

  멀티캐스트 그룹과 연결된 디바이스가 멀티캐스트 설정 및 작동이 가능한 경우 상태는 멀티캐스트 설정 시도 중(Multicast setup attempting)입니다. 이 상태는 디바이스가 아직 멀티캐스트 설정을 완료하지 않았음을 나타냅니다.

- 멀티캐스트 설정 준비(ulticast setup ready)

  디바이스가 멀티캐스트 설정을 완료하고 멀티캐스트 그룹에 추가되었습니다. 이 상태는 디바이스가 멀티캐스트 세션에 사용할 준비가 되었고 다운링크 메시지를 해당 디바이스로 전송할 수 있음을 나타냅니다. 또한 FUOTA를 사용하여 그룹의 디바이스 펌웨어를 업데이트할 수 있는 시기를 나타냅니다.

- 세션 시도 중(Session attempting)

  멀티캐스트 그룹의 디바이스에 대해 멀티캐스트 세션이 예약되었습니다. 멀티캐스트 그룹 세션이 시작될 때 디바이스 상태는 세션 시도 중이고 세션에 대해 클래스 B 또는 클래스 C 배포 기간을 시작할 수 있는지 여부에 대한 요청이 전송됩니다. 멀티캐스트 세션을 설정하는 데 걸리는 시간이 시간 제한을 초과하거나 멀티캐스트 세션을 취소하면 상태가 멀티캐스트 설정 완료(Multicast setup done)로 변경됩니다.

- 세션 중(In session)

이 상태는 클래스 B 또는 클래스 C 배포 기간이 시작되었고 디바이스에 진행 중인 멀티캐스트 세션이 있음을 나타냅니다. 이 시간 동안 AWS IoT Core for LoRaWAN에서 멀티캐스트 그룹의 디바이스로 다운링크 메시지를 전송할 수 있습니다. 세션 시간을 업데이트하면 현재 세션이 무시되고 상태가 세션 시도 중(Session attempting)으로 변경됩니다. 세션 시간이 종료되거나 멀티캐스트 세션을 취소하면 상태가 멀티캐스트 설정 준비(Multicast setup ready)로 변경됩니다.

다음 단계

멀티캐스트 그룹과 그룹 내 디바이스의 다양한 상태와 디바이스가 멀티캐스트 설정을 할 수 없는 경우와 같은 문제를 해결하는 방법을 배웠으므로 다운링크 메시지가 디바이스로 전송되도록 예약할 수 있으며 멀티캐스트 그룹은 세션 중(In session)이 됩니다. 다운링크 메시지 예약에 대한 자세한 내용은 멀티캐스트 그룹의 디바이스로 전송하도록 다운링크 메시지 예약 섹션을 참조하세요.

멀티캐스트 그룹의 디바이스로 전송하도록 다운링크 메시지 예약

멀티캐스트 그룹에 디바이스를 성공적으로 추가한 후 멀티캐스트 세션을 시작하고 이러한 디바이스로 전송되도록 다운링크 메시지를 구성할 수 있습니다. 다운링크 메시지는 48시간 이내에 예약되어야 하며 멀티캐스트 시작 시간은 현재 시간보다 30분 이상 늦어야 합니다.

> **Note**
>
> 멀티캐스트 그룹의 디바이스는 언제 다운링크 메시지가 수신되었는지 확인할 수 없습니다.

필수 조건

다운링크 메시지를 전송하려면 먼저 멀티캐스트 그룹을 생성하고 다운링크 메시지를 전송하려는 그룹에 디바이스를 성공적으로 추가해야 합니다. 멀티캐스트 세션의 시작 시간이 예약된 후에는 디바이스를 더 추가할 수 없습니다. 자세한 내용은 멀티캐스트 그룹 생성 및 그룹에 디바이스 추가 단원을 참조하십시오.

디바이스가 성공적으로 추가되지 않은 경우 멀티캐스트 그룹 및 디바이스 상태에 오류 해결에 도움이 되는 정보가 포함됩니다. 오류가 지속되는 경우 이러한 오류 해결에 대한 자세한 내용은 멀티캐스트 그룹 및 그룹의 디바이스 상태 모니터링 및 문제 해결 섹션을 참조하세요.

콘솔을 사용하여 다운링크 메시지 예약

콘솔을 사용하여 다운링크 메시지를 전송하려면 AWS IoT 콘솔의 멀티캐스트 그룹(Multicast groups) 페이지로 이동하고 생성한 멀티캐스트 그룹을 선택합니다. 멀티캐스트 그룹 세부 정보 페이지에서

다운링크 메시지 예약(Schedule downlink message)을 선택한 다음 다운링크 세션 예약(Schedule downlink session)을 선택합니다.

1. 다운링크 메시지 예약(Schedule downlink message) 창

   다운링크 메시지가 멀티캐스트 그룹의 디바이스로 전송될 기간을 설정할 수 있습니다. 다운링크 메시지는 48시간 이내에 예약되어야 합니다.

   멀티캐스트 세션을 예약하려면 다음 파라미터를 지정합니다.

   - 시작 날짜 및 시작 시간: 시작 날짜 및 시간은 현재 시간 이후 최소 30분 및 48시간 이전이어야 합니다.

     > **Note**
     >
     > 지정하는 시간은 UTC이므로 다운링크 기간을 예약할 때 시간대와 시차를 확인하는 것이 좋습니다.

   - 세션 시간 제한(Session timeout): 다운링크 메시지가 수신되지 않은 경우 멀티캐스트 세션이 시간 초과되기를 원하는 시간입니다. 허용되는 최소 시간 제한은 60초입니다. 최대 시간 제한 값은 클래스 B 멀티캐스트 그룹의 경우 2일, 클래스 C 멀티캐스트 그룹의 경우 18시간입니다.

2. 다운링크 메시지 구성

   다운링크 메시지를 구성하려면 다음 파라미터를 지정합니다.

   - 데이터 속도(Data rate): 다운링크 메시지의 데이터 속도를 선택합니다. 데이터 속도는 RFRegion 및 페이로드 크기에 따라 다릅니다. 기본 데이터 속도는 US915 리전의 경우 8이고 EU868 리전의 경우 0입니다.
   - 주파수(Frequency): 다운링크 메시지를 전송할 주파수를 선택합니다. 메시징 충돌을 방지하려면 RFRegion에 따라 사용 가능한 주파수를 선택합니다.
   - FPort: 다운링크 메시지를 디바이스로 전송하는 데 사용 가능한 주파수 포트를 선택합니다.
   - 페이로드(Payload): 데이터 속도에 따라 페이로드의 최대 크기를 지정합니다. 기본 데이터 속도를 사용하는 경우 최대 페이로드 크기는 US915 RfRegion에서 33바이트이고, EU868 RfRegion에서 51바이트입니다. 더 큰 데이터 속도를 사용하면 최대 페이로드 크기인 242바이트까지 전송할 수 있습니다.

   다운링크 메시지를 예약하려면 예약(Schedule)을 선택합니다.

API를 사용하여 다운링크 메시지 예약

API를 사용하여 다운링크 메시지를 예약하려면 `StartMulticastGroupSession` API 작업 또는 `start-multicast-group-session` CLI 명령을 사용합니다.

다음 API 작업 또는 CLI 명령을 사용하여 멀티캐스트 그룹에 대한 정보를 보고 멀티캐스트 그룹을 삭제할 수 있습니다.

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

세션이 시작된 후 멀티캐스트 그룹으로 데이터를 전송하려면 `SendDataToMulticastGroup` API 작업 또는 `send-data-to-multicast-group` CLI 명령을 사용합니다.

다음 단계

디바이스로 전송될 다운링크 메시지를 구성한 후 세션이 시작될 때 메시지가 전송됩니다. 멀티캐스트 그룹의 디바이스는 메시지 수신 여부를 확인할 수 없습니다.

추가 다운링크 메시지 구성

멀티캐스트 그룹의 디바이스로 전송될 추가 다운링크 메시지를 구성할 수도 있습니다.

- 콘솔에서 추가 다운링크 메시지를 구성하려면
  1. AWS IoT 콘솔의 멀티캐스트 그룹(Multicast groups) 페이지로 이동하고 생성한 멀티캐스트 그룹을 선택합니다.
  2. 멀티캐스트 그룹 세부 정보 페이지에서 다운링크 메시지 예약(Schedule downlink message)을 선택한 다음 추가 다운링크 메시지 구성(Configure additional downlink message)을 선택합니다.
  3. 데이터 속도(Data rate), 주파수(Frequency), FPort 및 페이로드(Payload)를 지정합니다. 방법은 첫 번째 다운링크 메시지에 대해 이러한 파라미터를 구성한 방법과 비슷합니다.
- API 또는 CLI를 사용하여 추가 다운링크 메시지를 구성하려면 각 추가 다운링크 메시지에 대해 `SendDataToMulticastGroup` API 작업 또는 `send-data-to-multicast-group` CLI 명령을 호출합니다.

세션 일정 업데이트

멀티캐스트 세션에 새로운 시작 날짜와 시간을 사용하도록 세션 일정을 업데이트할 수도 있습니다. 새 세션 일정은 이전에 예약된 세션보다 우선합니다.

> **Note**
>
> 필요한 경우에만 멀티캐스트 세션을 업데이트합니다. 이러한 업데이트로 인해 디바이스 그룹이 오랜 시간 동안 깨어 있어 배터리가 소모될 수 있습니다.

- 콘솔에서 세션 일정을 업데이트하려면
  1. AWS IoT 콘솔의 멀티캐스트 그룹(Multicast groups) 페이지로 이동하고 생성한 멀티캐스트 그룹을 선택합니다.
  2. 멀티캐스트 그룹 세부 정보 페이지에서 다운링크 메시지 예약(Schedule downlink message)을 선택한 다음 세션 일정 업데이트(Update session schedule)를 선택합니다.
  3. 상태 날짜(State date), 시작 시간(Start time) 및 세션 시간 제한(Session timeout) 파라미터를 지정합니다. 방법은 첫 번째 다운링크 메시지에 대해 이러한 파라미터를 지정한 방법과 비슷합니다.
- API 또는 CLI에서 세션 일정을 업데이트하려면 `StartMulticastGroupSession` API 작업 또는 `start-multicast-group-session` CLI 명령을 사용합니다.

AWS IoT Core for LoRaWAN 디바이스의 펌웨어 무선 업데이트(FUOTA)

펌웨어 무선 업데이트(FUOTA)를 사용하여 AWS IoT Core for LoRaWAN 디바이스에 펌웨어 업데이트를 배포합니다.

FUOTA를 사용하여 개별 디바이스 또는 디바이스 그룹에 펌웨어 업데이트를 전송할 수 있습니다. 멀티캐스트 그룹을 생성하여 여러 디바이스에 펌웨어 업데이트를 전송할 수도 있습니다. 먼저 디바이스를 멀티캐스트 그룹에 추가한 다음 펌웨어 업데이트 이미지를 해당하는 모든 디바이스에 전송합니다. 이미지를 수신하는 디바이스가 이미지가 올바른 소스에서 오는지 확인할 수 있도록 펌웨어 이미지에 디지털 서명을 하는 것이 좋습니다.

AWS IoT Core for LoRaWAN의 FUOTA를 사용하면 다음을 수행할 수 있습니다.

- 새 펌웨어 이미지나 델타 이미지를 단일 디바이스 또는 디바이스 그룹에 배포합니다.
- 디바이스에 배포된 이후에 새 펌웨어의 신뢰성과 무결성을 확인합니다.
- 배포 진행 상황을 모니터링하고 배포에 실패한 경우 문제를 디버그합니다.

FUOTA 및 멀티캐스트 그룹에 대한 AWS IoT Core for LoRaWAN의 지원은 LoRa Alliance의 다음 사양을 기반으로 합니다.

- LoRaWAN 원격 멀티캐스트 설정 사양, TS005-2.0.0
- LoRaWAN 단편화된 데이터 블록 전송 사양, TS004-2.0.0
- LoRaWAN 애플리케이션 레이어 클록 동기화 사양, TS003-2.0.0

> **Note**
>
> AWS IoT Core for LoRaWAN은 LoRa Alliance 사양에 따라 클록 동기화를 자동으로 수행합니다. `AppTimeReq` 함수로 ClockSync 시그널링을 사용하여 요청하는 디바이스에 서버 측 시간을 회신합니다.

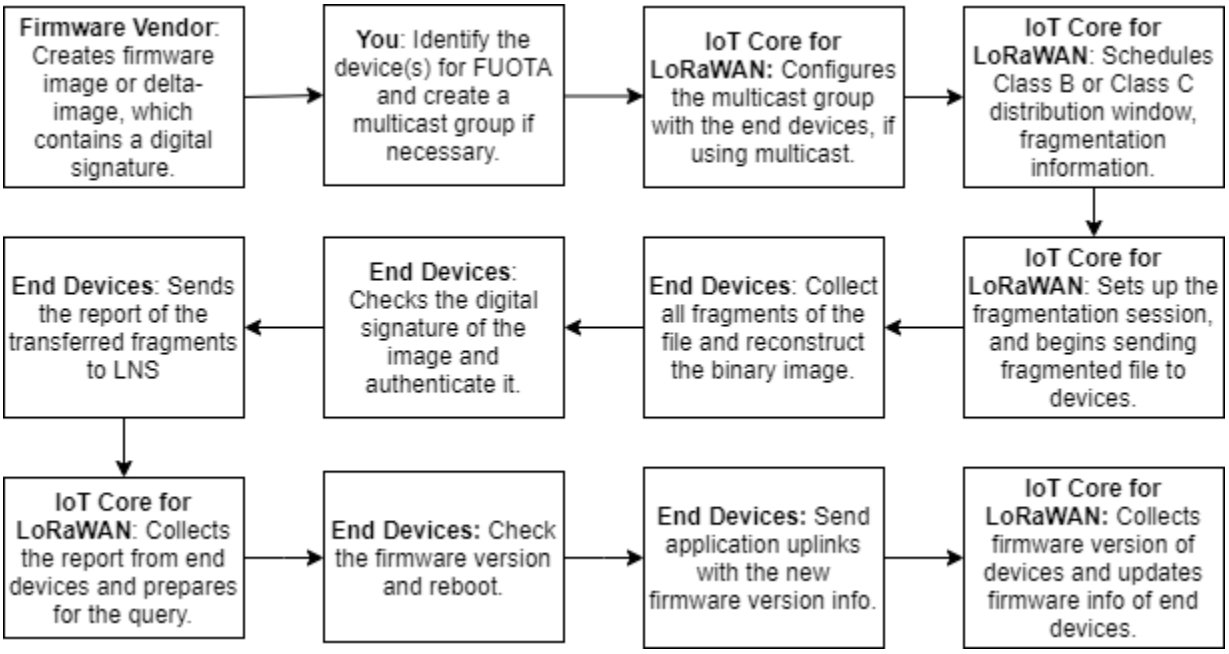
다음 동영상은 AWS IoT Core for LoRaWAN FUOTA 태스크를 생성하는 방법을 설명하고 태스크에 디바이스를 추가하고 FUOTA 태스크를 예약하는 프로세스를 안내합니다.

다음 주제는 FUOTA를 수행하는 방법을 보여 줍니다.

- FUOTA 프로세스 개요
- FUOTA 태스크 생성 및 펌웨어 이미지 제공
- FUOTA 태스크에 디바이스 및 멀티캐스트 그룹 추가 및 FUOTA 세션 예약
- FUOTA 태스크 및 태스크에 추가된 디바이스의 상태 모니터링 및 문제 해결

FUOTA 프로세스 개요

다음 다이어그램은 AWS IoT Core for LoRaWAN이 종단 디바이스에 대해 FUOTA 프로세스를 수행하는 방법을 보여줍니다. FUOTA 세션에 개별 디바이스를 추가하는 경우 멀티캐스트 그룹 생성 및 구성 단계를 건너뛸 수 있습니다. FUOTA 세션에 디바이스를 직접 추가할 수 있습니다. 그러면 AWS IoT Core for LoRaWAN이 펌웨어 업데이트 프로세스를 시작합니다.



디바이스에 대한 FUOTA를 수행하려면 먼저 디지털 서명된 펌웨어 이미지를 생성하고 FUOTA 태스크에 추가하려는 디바이스 및 멀티캐스트 그룹을 구성합니다. FUOTA 세션을 시작한 후 종단 디바이스는 모든 조각을 수집하고 조각에서 이미지를 재구성하고 상태를 AWS IoT Core for LoRaWAN에 보고한 다음 새 펌웨어 이미지를 적용합니다.

다음은 FUOTA 프로세스의 여러 단계를 보여줍니다.

1. 디지털 서명을 사용하여 펌웨어 이미지 또는 델타 이미지 생성

   AWS IoT Core for LoRaWAN이 LoRaWAN 디바이스에 대한 FUOTA를 수행하려면 무선으로 펌웨어 업데이트를 전송할 때 펌웨어 이미지 또는 델타 이미지에 디지털 서명을 하는 것이 좋습니다. 그러면 이미지를 수신하는 디바이스에서 이미지가 올바른 소스에서 왔는지 확인할 수 있습니다.

   펌웨어 이미지의 크기는 1MB를 넘지 않아야 합니다. 펌웨어 크기가 클수록 업데이트 프로세스를 완료하는 데 더 오래 걸릴 수 있습니다. 더 빠른 데이터 전송을 위해 또는 새 이미지가 1MB보다 큰 경우 새 펌웨어 이미지와 이전 이미지 사이의 델타인 새 이미지의 일부인 델타 이미지를 사용합니다.

   > **Note**
   >
   > AWS IoT Core for LoRaWAN은 디지털 서명 생성 도구와 펌웨어 버전 관리 시스템을 제공하지 않습니다. 서드 파티 도구를 사용하여 펌웨어 이미지에 대한 디지털 서명을 생성할 수 있습니다. ARM Mbed GitHub 리포지토리에 포함된 것과 같은 디지털 서명 도구를 사용

하는 것이 좋습니다. 여기에는 델타 이미지를 생성하고 디바이스에서 해당 이미지를 사용하기 위한 도구도 포함됩니다.

2. FUOTA용 디바이스 식별 및 구성

   FUOTA용 디바이스를 식별한 후 펌웨어 업데이트를 개별 또는 여러 디바이스에 전송합니다.

   - 펌웨어 업데이트를 여러 디바이스로 전송하려면 멀티캐스트 그룹을 생성하고 종단 디바이스로 멀티캐스트 그룹을 구성합니다. 자세한 내용은 멀티캐스트 그룹을 생성하여 여러 디바이스로 다운링크 페이로드를 전송합니다. 단원을 참조하십시오.
   - 개별 디바이스에 펌웨어 업데이트를 전송하려면 해당 디바이스를 FUOTA 세션에 추가한 다음 펌웨어 업데이트를 수행합니다.

3. 배포 기간 예약 및 조각화 세션 설정

   멀티캐스트 그룹을 만든 경우 클래스 B 또는 클래스 C 배포 기간을 지정하여 디바이스가 AWS IoT Core for LoRaWAN에서 조각을 수신할 수 있는 시기를 결정할 수 있습니다. 디바이스가 클래스 B 또는 클래스 C 모드로 전환하기 전에 클래스 A에서 작동 중일 수 있습니다. 세션의 시작 시간도 지정해야 합니다.

   클래스 B 또는 클래스 C 디바이스가 지정된 배포 기간에 깨어나 다운링크 패킷을 수신하기 시작합니다. 클래스 C 모드에서 작동하는 디바이스는 클래스 B 디바이스보다 더 많은 전력을 소비할 수 있습니다. 자세한 내용은 디바이스 클래스 단원을 참조하십시오.

4. 종단 디바이스가 AWS IoT Core for LoRaWAN에 상태를 보고하고 펌웨어 이미지를 업데이트합니다.

   조각화 세션을 설정한 후 종단 디바이스와 AWS IoT Core for LoRaWAN은 다음 단계를 수행하여 디바이스의 펌웨어를 업데이트합니다.

   1. LoRaWAN 디바이스는 데이터 속도가 낮기 때문에 FUOTA 프로세스를 시작하기 위해 AWS IoT Core for LoRaWAN은 조각화 세션을 설정하여 펌웨어 이미지를 조각화합니다. 그런 다음 이 조각을 종단 디바이스로 전송합니다.
   2. AWS IoT Core for LoRaWAN이 이미지 조각을 전송한 후 LoRaWAN 종단 디바이스는 다음 태스크를 수행합니다.
      a. 조각을 수집한 다음 이러한 조각에서 이진 이미지를 재구성합니다.
      b. 재구성된 이미지의 디지털 서명을 확인하여 이미지를 인증하고 올바른 소스에서 왔는지 확인합니다.
      c. AWS IoT Core for LoRaWAN의 펌웨어 버전을 현재 버전과 비교합니다.

d. AWS IoT Core for LoRaWAN으로 전송된 조각화된 이미지의 상태를 보고한 후 새 펌웨어 이미지를 적용합니다.

> **Note**
>
> 경우에 따라 종단 디바이스는 펌웨어 이미지의 디지털 서명을 확인하기 전 AWS IoT Core for LoRaWAN에 전송된 조각화된 이미지의 상태를 보고합니다.

이제 FUOTA 프로세스를 배웠으므로 FUOTA 태스크를 생성하고 펌웨어 업데이트를 위해 태스크에 디바이스를 추가할 수 있습니다. 자세한 내용은 FUOTA 태스크 생성 및 펌웨어 이미지 제공 단원을 참조하십시오.

FUOTA 태스크 생성 및 펌웨어 이미지 제공

LoRaWAN 디바이스의 펌웨어를 업데이트하려면 먼저 FUOTA 태스크를 생성하고 업데이트에 사용할 디지털 서명된 펌웨어 이미지를 제공합니다. 그런 다음 디바이스와 멀티캐스트 그룹을 태스크에 추가하고 FUOTA 세션을 예약할 수 있습니다. 세션이 시작되면 AWS IoT Core for LoRaWAN이 조각화 세션을 설정하고 종단 디바이스가 조각을 수집하고 이미지를 재구성하고 새 펌웨어를 적용합니다. FUOTA 프로세스에 대한 자세한 내용은 FUOTA 프로세스 개요 섹션을 참조하세요.

다음은 FUOTA 태스크를 생성하고 S3 버킷에 저장할 펌웨어 이미지 또는 델타 이미지를 업로드하는 방법을 보여줍니다.

필수 조건

FUOTA를 수행하기 전에 종단 디바이스가 이미지를 적용할 때 이미지의 신뢰성을 확인할 수 있도록 펌웨어 이미지에 디지털 서명을 해야 합니다. 서드 파티 도구를 사용하여 펌웨어 이미지에 대한 디지털 서명을 생성할 수 있습니다. ARM Mbed GitHub 리포지토리에 포함된 것과 같은 디지털 서명 도구를 사용하는 것이 좋습니다. 여기에는 델타 이미지를 생성하고 디바이스에서 해당 이미지를 사용하기 위한 도구도 포함됩니다.

콘솔을 사용하여 FUOTA 태스크 생성 및 펌웨어 이미지 업로드

콘솔을 사용하여 FUOTA 태스크를 생성하고 펌웨어 이미지를 업로드하려면 콘솔의 FUOTA 태스크(FUOTA tasks) 탭으로 이동한 다음 FUOTA 태스크 생성(Create FUOTA task)을 선택합니다.

1. FUOTA 태스크 생성

   FUOTA 태스크를 생성하려면 태스크 속성과 태그를 지정합니다.

1. FUOTA 태스크 속성 지정

   FUOTA 태스크 속성을 지정하려면 FUOTA 태스크에 대해 다음 정보를 입력합니다.

   - 이름(Name): FUOTA 태스크의 고유한 이름을 입력합니다. 이름에는 문자, 숫자, 하이픈 및 밑줄만 포함되어야 합니다. 공백은 포함할 수 없습니다.
   - 설명(Description): 멀티캐스트 그룹에 대한 설명(선택 사항)을 제공할 수 있습니다. 설명 필드는 최대 2,048자까지 가능합니다.
   - RFRegion: FUOTA 태스크의 주파수 대역을 설정합니다. 주파수 대역은 무선 디바이스 또는 멀티캐스트 그룹을 프로비저닝하는 데 사용한 것과 일치해야 합니다.

2. FUOTA 태스크 태그

   선택적으로 모든 키-값 페어를 FUOTA 태스크에 대한 태그(Tags)로 제공할 수 있습니다. 태스크 생성을 계속하려면 다음(Next)을 선택합니다.

2. 펌웨어 이미지 업로드

FUOTA 태스크에 추가하는 디바이스의 펌웨어를 업데이트하는 데 사용할 펌웨어 이미지 파일을 선택합니다. 펌웨어 이미지 파일은 S3 버킷에 저장됩니다. 사용자를 대신하여 AWS IoT Core for LoRaWAN에 펌웨어 이미지에 액세스할 수 있는 권한을 제공할 수 있습니다. 펌웨어 업데이트를 수행할 때 신뢰성을 확인할 수 있도록 펌웨어 이미지에 디지털 서명을 하는 것이 좋습니다.

1. 펌웨어 이미지 파일 선택

   새 펌웨어 이미지 파일을 S3 버킷에 업로드하거나 S3 버킷에 이미 업로드된 기존 이미지를 선택할 수 있습니다.

   > **Note**
   >
   > 펌웨어 이미지 파일의 크기는 1MB를 넘지 않아야 합니다. 펌웨어 크기가 클수록 업데이트 프로세스를 완료하는 데 더 오래 걸릴 수 있습니다.

   - 기존 이미지를 사용하려면 기존 펌웨어 이미지 선택(Select an existing firmware image)을 선택하고 S3 찾아보기(Browse S3)를 선택한 다음 사용할 펌웨어 이미지 파일을 선택합니다.

     AWS IoT Core for LoRaWAN은 S3 버킷에 있는 펌웨어 이미지 파일의 경로인 S3 URL을 채웁니다. 경로의 형식은 `s3://bucket_name/file_name`입니다. Amazon Simple Storage Service 콘솔에서 파일을 보려면 보기(View)를 선택합니다.

- 새 펌웨어 이미지를 업로드하려면

  a. 새 펌웨어 이미지 업로드를 선택하고 펌웨어 이미지를 업로드합니다. 이미지 파일은 1MB를 넘지 않아야 합니다.

  b. S3 버킷을 생성하고 펌웨어 이미지 파일을 저장할 버킷 이름(Bucket name)을 입력하려면 S3 버킷 생성(Create S3 bucket)을 선택합니다.

2. 버킷에 액세스할 수 있는 권한

   새 서비스 역할을 생성하거나 기존 역할을 선택하여 AWS IoT Core for LoRaWAN이 사용자를 대신하여 S3 버킷의 펌웨어 이미지 파일에 액세스하도록 허용할 수 있습니다. 다음을 선택합니다.

   새 역할을 생성하려면 역할 이름을 입력하거나 임의의 이름이 자동으로 생성되도록 비워 둘 수 있습니다. S3 버킷에 대한 액세스 권한을 부여하는 정책 권한을 보려면 정책 권한 보기(View policy permissions)를 선택합니다.

   S3 버킷을 사용하여 이미지를 저장하고 AWS IoT Core for LoRaWAN에 액세스 권한을 부여하는 방법에 대한 자세한 내용은 S3 버킷에 펌웨어 파일 업로드 및 IAM 역할 추가 섹션을 참조하세요.

3. 검토 및 생성

   FUOTA 태스크를 생성하려면 지정한 FUOTA 태스크 및 구성 세부 정보를 검토하고 태스크 생성(Create task)을 선택합니다.

API를 사용하여 FUOTA 태스크 생성 및 펌웨어 이미지 업로드

API를 사용하여 FUOTA 태스크를 생성하고 펌웨어 이미지 파일을 지정하려면 `CreateFuotaTask` API 작업 또는 `create-fuota-task` CLI 명령을 사용합니다. `input.json` 파일을 `create-fuota-task` 명령에 대한 입력으로 제공할 수 있습니다. API 또는 CLI를 사용할 때 입력으로 제공하는 펌웨어 이미지 파일은 이미 S3 버킷에 업로드되어 있어야 합니다. 또한 S3 버킷의 펌웨어 이미지에 대한 액세스 권한을 AWS IoT Core for LoRaWAN에 부여하는 IAM 역할을 지정합니다.

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

여기서 각 항목은 다음과 같습니다.

input.json 내용

```
{
   "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 태스크를 생성한 후 다음 API 작업 또는 CLI 명령을 사용하여 FUOTA 태스크에 대한 정보를 업데이트, 삭제 또는 가져올 수 있습니다.

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

다음 단계

FUOTA 태스크를 생성하고 펌웨어 이미지를 제공했으므로 이제 펌웨어 업데이트를 위해 작업에 디바이스를 추가할 수 있습니다. 개별 디바이스나 멀티캐스트 그룹을 작업에 추가할 수 있습니다. 자세한 내용은 FUOTA 태스크에 디바이스 및 멀티캐스트 그룹 추가 및 FUOTA 세션 예약 단원을 참조하십시오.

FUOTA 태스크에 디바이스 및 멀티캐스트 그룹 추가 및 FUOTA 세션 예약

FUOTA 태스크를 생성한 후 펌웨어를 업데이트하려는 작업에 디바이스를 추가할 수 있습니다. 디바이스가 FUOTA 태스크에 성공적으로 추가되면 FUOTA 세션을 예약하여 디바이스 펌웨어를 업데이트할 수 있습니다.

- 디바이스 수가 적은 경우 해당 디바이스를 FUOTA 태스크에 직접 추가할 수 있습니다.
- 펌웨어를 업데이트하려는 디바이스가 많은 경우 이러한 디바이스를 멀티캐스트 그룹에 추가한 다음 멀티캐스트 그룹을 FUOTA 태스크에 추가할 수 있습니다. 멀티캐스트 그룹 생성 및 사용에 대한 자세한 내용은 멀티캐스트 그룹을 생성하여 여러 디바이스로 다운링크 페이로드를 전송합니다. 섹션을 참조하세요.

> **Note**
>
> 개별 디바이스나 멀티캐스트 그룹을 FUOTA 태스크에 추가할 수 있습니다. 작업에 디바이스와 멀티캐스트 그룹을 모두 추가할 수는 없습니다.

디바이스 또는 멀티캐스트 그룹을 추가한 후 펌웨어 업데이트 세션을 시작할 수 있습니다. AWS IoT Core for LoRaWAN은 펌웨어 이미지를 수집하고 이미지를 조각화한 다음 암호화된 형식으로 조각을 저장합니다. 종단 디바이스는 조각을 수집하고 새 펌웨어 이미지를 적용합니다. 펌웨어 업데이트에 걸리는 시간은 이미지 크기와 이미지 조각화 방식에 따라 다릅니다. 펌웨어 업데이트가 완료되면 AWS IoT Core for LoRaWAN에서 저장하는 펌웨어 이미지의 암호화된 조각이 삭제됩니다. S3 버킷에서 펌웨어 이미지를 계속 찾을 수 있습니다.

필수 조건

FUOTA 태스크에 디바이스 또는 멀티캐스트 그룹을 추가하기 전에 다음을 수행합니다.

- FUOTA 태스크를 이미 생성하고 펌웨어 이미지를 제공했어야 합니다. 자세한 내용은 FUOTA 태스크 생성 및 펌웨어 이미지 제공 단원을 참조하십시오.
- 디바이스 펌웨어를 업데이트할 무선 디바이스를 프로비저닝합니다. 디바이스 온보딩에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 디바이스 온보딩 섹션을 참조하세요.
- 여러 디바이스의 펌웨어를 업데이트하려면 멀티캐스트 그룹에 해당 디바이스를 추가합니다. 자세한 내용은 멀티캐스트 그룹을 생성하여 여러 디바이스로 다운링크 페이로드를 전송합니다. 단원을 참조하십시오.
- 디바이스를 AWS IoT Core for LoRaWAN에 온보딩할 때 FUOTA 구성 파라미터 `FPorts`를 지정합니다. LoRaWAN v1.0.x 디바이스를 사용하는 경우 `GenAppKey`도 지정해야 합니다. FUOTA 구성 파라미터에 대한 자세한 내용은 멀티캐스트 및 FUOTA 구성을 위한 디바이스 준비 섹션을 참조하세요.

콘솔을 사용하여 FUOTA 태스크에 디바이스 추가 및 FUOTA 세션 예약

콘솔을 사용하여 디바이스 또는 멀티캐스트 그룹을 추가하고 FUOTA 세션을 예약하려면 콘솔의 FUOTA 태스크(FUOTA tasks) 탭으로 이동합니다. 그런 다음 디바이스를 추가할 FUOTA 태스크를 선택하고 펌웨어 업데이트를 수행합니다.

디바이스 및 멀티캐스트 그룹 추가

1. 개별 디바이스나 멀티캐스트 그룹을 FUOTA 태스크에 추가할 수 있습니다. 그러나 개별 디바이스와 멀티캐스트 그룹을 모두 동일한 FUOTA 태스크에 추가할 수는 없습니다. 콘솔을 사용하여 디바이스를 추가하려면 다음을 수행합니다.

   1. FUOTA 태스크 세부 정보(FUOTA task details)에서 디바이스 추가(Add device)를 선택합니다.
   2. 작업에 추가하는 디바이스의 주파수 대역 또는 RFRegion을 선택합니다. 이 값은 FUOTA 태스크에 대해 선택한 RFRegion과 일치해야 합니다.
   3. 태스크에 개별 디바이스를 추가할지 아니면 멀티캐스트 그룹을 추가할지 선택합니다.
      - 개별 디바이스를 추가하려면 개별 디바이스 추가(Add individual devices)를 선택하고 FUOTA 태스크에 추가하려는 각 디바이스의 디바이스 ID를 입력합니다.
      - 멀티캐스트 그룹을 추가하려면 멀티캐스트 그룹 추가(Add multicast groups)를 선택하고 태스크에 멀티캐스트 그룹을 추가합니다. 디바이스 프로파일 또는 태그를 사용하여 작업에 추가하려는 멀티캐스트 그룹을 필터링할 수 있습니다. 디바이스 프로파일로 필터링할 때 클래스 B 지원(Supports Class B) 또는 클래스 C 지원(Supports Class C)이 사용되는 프로파일이 있는 디바이스가 있는 멀티캐스트 그룹을 선택할 수 있습니다.
2. FUOTA 세션 예약

   디바이스 또는 멀티캐스트 그룹이 성공적으로 추가된 후 FUOTA 세션을 예약할 수 있습니다. 세션을 예약하려면 다음을 수행합니다.

   1. 디바이스 펌웨어를 업데이트하려는 FUOTA 태스크를 선택한 다음 FUOTA 세션 예약(Schedule FUOTA session)을 선택합니다.
   2. FUOTA 세션의 시작 날짜(Start date)와 시작 시간(Start time)을 지정합니다. 시작 시간이 현재 시간에서 30분 이후인지 확인합니다.

API를 사용하여 FUOTA 태스크에 디바이스 추가 및 FUOTA 세션 예약

AWS IoT 무선 API 또는 CLI를 사용하여 무선 디바이스 또는 멀티캐스트 그룹을 FUOTA 태스크에 추가할 수 있습니다. 그런 다음 FUOTA 세션을 예약할 수 있습니다.

1. 디바이스 및 멀티캐스트 그룹 추가

   무선 디바이스 또는 멀티캐스트 그룹을 FUOTA 태스크와 연결할 수 있습니다.

   - 개별 디바이스를 FUOTA 태스크에 연결하려면 `AssociateWirelessDeviceWithFuotaTask` API 작업 또는 `associate-wireless-`

device-with-fuota-task CLI 명령을 사용하고 WirelessDeviceID를 입력으로 제공합니다.

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

- 멀티캐스트 그룹을 FUOTA 태스크에 연결하려면 AssociateMulticastGroupWithFuotaTask API 작업 또는 associate-multicast-group-with-fuota-task CLI 명령을 사용하고 MulticastGroupID를 입력으로 제공합니다.

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

무선 디바이스 또는 멀티캐스트 그룹을 FUOTA 태스크에 연결한 후 다음 API 작업 또는 CLI 명령을 사용하여 디바이스 또는 멀티캐스트 그룹을 나열하거나 작업에서 연결을 해제합니다.

- DisassociateWirelessDeviceFromFuotaTask 또는 disassociate-wireless-device-from-fuota-task
- DisassociateMulticastGroupFromFuotaTask 또는 disassociate-multicast-group-from-fuota-task
- ListWirelessDevices 또는 list-wireless-devices
- ListMulticastGroups 또는 list-multicast-groups-by-fuota-task

> **Note**
>
> API:
>
> - ListWirelessDevices는 MulticastGroupID가 필터로 사용될 때 일반 무선 디바이스와 멀티캐스트 그룹과 관련된 디바이스를 나열할 수 있습니다. API는 FuotaTaskID가 필터로 사용될 때 FUOTA 태스크와 관련된 무선 디바이스를 나열합니다.
> - ListMulticastGroups는 일반적으로 멀티캐스트 그룹을 나열하고 FuotaTaskID가 필터로 사용될 때 FUOTA 태스크와 관련된 멀티캐스트 그룹을 나열할 수 있습니다.

2. FUOTA 세션 예약

   디바이스 또는 멀티캐스트 그룹이 FUOTA 태스크에 성공적으로 추가되면 FUOTA 세션을 시작하여 디바이스 펌웨어를 업데이트할 수 있습니다. 시작 시간은 현재 시간에서 30분 이후여야 합니다. API 또는 CLI를 사용하여 FUOTA 세션을 예약하려면 `StartFuotaTask` API 작업 또는 `start-fuota-task` CLI 명령을 사용합니다.

   FUOTA 세션을 시작한 후에는 더 이상 디바이스 또는 멀티캐스트 그룹을 작업에 추가할 수 없습니다. `GetFuotaTask` API 작업 또는 `get-fuota-task` CLI 명령을 사용하여 FUOTA 세션의 상태에 대한 정보를 볼 수 있습니다.

FUOTA 태스크 및 태스크에 추가된 디바이스의 상태 모니터링 및 문제 해결

무선 디바이스를 프로비저닝하고 사용하려는 멀티캐스트 그룹을 생성한 후 다음 단계를 수행하여 FUOTA 세션을 시작할 수 있습니다.

FUOTA 태스크 상태

AWS Management Console에 FUOTA 태스크에 대한 다음 상태 메시지 중 하나가 표시될 수 있습니다.

- 보류중

  이 상태는 FUOTA 태스크를 생성했지만 아직 펌웨어 업데이트 세션이 없음을 나타냅니다. 태스크가 생성되면 이 상태 메시지가 표시됩니다. 이 시간 동안 FUOTA 태스크를 업데이트하고 디바이스 또는 멀티캐스트 그룹을 태스크와 연결하거나 연결 해제할 수 있습니다. 상태가 보류 중(Pending)에서 변경된 후에는 추가 디바이스를 태스크에 추가할 수 없습니다.

- FUOTA 세션 대기 중(FUOTA session waiting)

  디바이스가 FUOTA 태스크에 성공적으로 추가된 후 작업에 예약된 펌웨어 업데이트 세션이 있을 때 이 상태 메시지가 표시됩니다. 이 시간 동안에는 디바이스를 업데이트하거나 FUOTA 세션에 추가할 수 없습니다. FUOTA 세션을 취소하면 그룹 상태가 보류 중(Pending)으로 변경됩니다.

- FUOTA 세션 중(In FUOTA session)

  FUOTA 세션이 시작되면 이 상태 메시지가 표시됩니다. 조각화 세션이 시작되고 종단 디바이스가 조각을 수집하고 펌웨어 이미지를 재구성하고 새 펌웨어 버전을 원래 버전과 비교하고 새 이미지를 적용합니다.

- FUOTA 완료(FUOTA done)

종단 디바이스에서 새 펌웨어 이미지가 적용되었다고 AWS IoT Core for LoRaWAN에 보고한 후 또는 세션 시간이 초과되면 FUOTA 세션이 완료로 표시되고 이 상태가 표시됩니다.

다음과 같은 경우에도 이 상태가 표시되므로 펌웨어 업데이트가 디바이스에 올바르게 적용되었는지 확인해야 합니다.

- FUOTA 태스크 상태가 FUOTA 세션 대기 중(FUOTA session waiting)이면 S3 버킷의 이미지 파일에 대한 링크가 올바르지 않거나 AWS IoT Core for LoRaWAN에 버킷의 파일에 액세스할 수 있는 충분한 권한이 없는 것과 같은 S3 버킷 오류가 있는 것입니다.
- FUOTA 태스크 상태가 FUOTA 세션 대기 중(FUOTA session waiting)이면 FUOTA 세션을 시작하라는 요청이 있지만 FUOTA 작업의 디바이스 또는 멀티캐스트 그룹으로부터 응답이 수신되지 않은 것입니다.
- FUOTA 태스크 상태가 FUOTA 세션 중(In FUOTA session)이면 디바이스 또는 멀티캐스트 그룹이 일정 기간 동안 조각을 전송하지 않아 세션이 시간 초과된 것입니다.

- 삭제 대기 중(Delete waiting)

  다른 상태의 FUOTA 태스크를 삭제하면 이 상태가 표시됩니다. 삭제 작업은 영구적이며 취소할 수 없습니다. 이 작업은 시간이 걸릴 수 있으며 FUOTA 태스크가 삭제될 때까지 태스크 상태는 Delete waiting(삭제 대기 중)이 됩니다. FUOTA 태스크가 이 상태가 된 후에는 다른 상태 중 하나로 전환할 수 없습니다.

FUOTA 태스크의 디바이스 상태

AWS Management Console에 FUOTA 태스크의 디바이스에 대한 다음 상태 메시지 중 하나가 표시될 수 있습니다. 각 상태 메시지 위로 마우스를 가져가서 표시 내용에 대한 자세한 정보를 볼 수 있습니다.

- Initial

  FUOTA 세션이 시작되면 AWS IoT Core for LoRaWAN이 디바이스에 펌웨어 업데이트를 지원하는 패키지가 있는지 확인합니다. 디바이스에 지원되는 패키지가 있는 경우 디바이스에 대한 FUOTA 세션이 시작됩니다. 펌웨어 이미지가 조각화되고 조각이 디바이스로 전송됩니다. 이 상태가 표시되면 디바이스에 대한 FUOTA 세션이 아직 시작되지 않은 것입니다.

- 패키지 지원되지 않음(Package unsupported)

  디바이스에 지원되는 FUOTA 패키지가 없으면 이 상태가 표시됩니다. 펌웨어 업데이트 패키지가 지원되지 않으면 디바이스의 FUOTA 세션을 시작할 수 없습니다. 이 오류를 해결하려면 디바이스의 펌웨어가 FUOTA를 사용하여 펌웨어 업데이트를 수신할 수 있는지 확인합니다.

- 조각화 알고리즘 지원되지 않음(Fragmentation algorithm unsupported)

  FUOTA 세션이 시작될 때 AWS IoT Core for LoRaWAN은 디바이스에 대한 조각화 세션을 설정합니다. 이 상태가 표시되면 사용된 조각화 알고리즘 유형이 디바이스의 펌웨어 업데이트에 적용될 수 없는 것입니다. 디바이스에 지원되는 FUOTA 패키지가 없기 때문에 오류가 발생합니다. 이 오류를 해결하려면 디바이스의 펌웨어가 FUOTA를 사용하여 펌웨어 업데이트를 수신할 수 있는지 확인합니다.

- 메모리가 충분하지 않음

  AWS IoT Core for LoRaWAN이 이미지 조각을 전송한 후 종단 디바이스는 이미지 조각을 수집하고 이 조각으로 이진 이미지를 재구성합니다. 이 상태는 디바이스에 들어오는 펌웨어 이미지 조각을 조합할 메모리가 충분하지 않아 펌웨어 업데이트 세션이 조기에 종료될 수 있는 경우 때 표시됩니다. 이 오류를 해결하려면 디바이스의 하드웨어가 이 업데이트를 수신할 수 있는지 확인합니다. 디바이스에서 이 업데이트를 수신할 수 없는 경우 델타 이미지를 사용하여 펌웨어를 업데이트합니다.

- 조각화 인덱스 지원되지 않음(Fragmentation index unsupported)

  조각화 인덱스는 동시에 가능한 4개의 조각화 세션 중 하나를 식별합니다. 디바이스가 표시된 조각화 인덱스 값을 지원하지 않는 경우 이 상태가 표시됩니다. 이 오류를 해결하려면 다음 중 하나 이상을 수행하세요.

  - 디바이스에 대한 새 FUOTA 태스크를 시작합니다.
  - 오류가 지속될 경우 유니캐스트에서 멀티캐스트 모드로 전환합니다.
  - 그래도 오류가 해결되지 않으면 디바이스 펌웨어를 확인합니다.

- 메모리 오류

  이 상태는 AWS IoT Core for LoRaWAN으로부터 들어오는 조각을 수신할 때 디바이스에 메모리 오류가 발생했음을 나타냅니다. 이 오류가 발생하면 디바이스에서 이 업데이트를 수신하지 못하는 것일 수 있습니다. 이 오류를 해결하려면 디바이스의 하드웨어가 이 업데이트를 수신할 수 있는지 확인합니다. 필요한 경우 델타 이미지를 사용하여 디바이스 펌웨어를 업데이트합니다.

- 잘못된 설명자(Wrong descriptor)

  디바이스가 표시된 설명자를 지원하지 않습니다. 설명자는 조각화 세션 동안 전송될 파일을 설명하는 필드입니다. 이 오류가 표시되면 AWS Support 센터에 문의하세요.

- 세션 수 다시 재생(Session count replay)

  이 상태는 디바이스가 이전에 이 세션 수를 사용했음을 나타냅니다. 오류를 해결하려면 디바이스에 대한 새 FUOTA 태스크를 시작합니다.

- 누락된 조각(Missing fragments)

디바이스가 AWS IoT Core for LoRaWAN에서 이미지 조각을 수집할 때 독립적이고 코딩된 조각에서 새 펌웨어 이미지를 재구성합니다. 디바이스가 모든 조각을 수신하지 못한 경우 새 이미지를 재구성할 수 없으며 이 상태가 표시됩니다. 오류를 해결하려면 디바이스에 대한 새 FUOTA 태스크를 시작합니다.

- 마이크 오류(MIC error)

  디바이스가 수집된 조각에서 새 펌웨어 이미지를 재구성할 때 MIC(Message Integrity Check)를 수행하여 이미지의 신뢰성과 이미지가 올바른 소스에서 왔는지 확인합니다. 디바이스가 조각을 재조립한 후 MIC에서 불일치를 감지하면 이 상태가 표시됩니다. 오류를 해결하려면 디바이스에 대한 새 FUOTA 태스크를 시작합니다.

- 성공(Successful)

  디바이스에 대한 FUOTA 세션이 성공했습니다.

> **Note**
>
> 이 상태 메시지는 디바이스가 조각에서 이미지를 재구성하고 확인했음을 나타내지만 디바이스가 상태를 AWS IoT Core for LoRaWAN에 보고할 때 디바이스 펌웨어가 업데이트되지 않았을 수 있습니다. 디바이스 펌웨어가 업데이트되었는지 확인합니다.

다음 단계

FUOTA 태스크 및 해당 디바이스의 다양한 상태와 문제를 해결할 수 있는 방법에 대해 알아보았습니다. 이러한 각 상태에 대한 자세한 내용은 LoRaWAN 조각화된 데이터 블록 전송 사양, TS004-1.0.0을 참조하세요.

네트워크 분석기를 사용하여 무선 리소스 플릿 실시간 모니터링

네트워크 분석기는 기본 WebSocket 연결을 사용하여 무선 연결 리소스에 대한 실시간 추적 메시지 로그를 수신합니다. 네트워크 분석기를 사용하여 모니터링하려는 리소스를 추가하고, 추적 메시징 세션을 활성화하고, 실시간으로 추적 메시지 수신을 시작할 수 있습니다.

리소스를 모니터링하기 위해 Amazon CloudWatch를 사용할 수도 있습니다. CloudWatch를 사용하려면 IAM 역할을 설정하여 로깅을 구성한 다음 로그 항목이 콘솔에 표시될 때까지 기다립니다. 네트워크 분석기는 연결을 설정하고 추적 메시지 수신을 시작하는 데 걸리는 시간을 크게 줄여 리소스 플릿에 대한 적시 로그 정보를 제공합니다. CloudWatch를 사용한 모니터링에 대한 자세한 내용은 Amazon CloudWatch Logs를 사용하여 AWS IoT 무선 리소스 모니터링 섹션을 참조하세요.

설정 시간을 줄이고 추적 메시지의 정보를 사용하여 리소스를 보다 효과적으로 모니터링하고 의미 있는 인사이트를 얻고 오류를 해결할 수 있습니다. LoRaWAN 디바이스와 LoRaWAN 게이트웨이를 모두 모니터링할 수 있습니다. 예를 들어 LoRaWAN 디바이스 중 하나를 온보딩할 때 조인 오류를 빠르게 식별할 수 있습니다. 오류를 디버깅하려면 제공된 추적 메시지 로그의 정보를 사용합니다.

네트워크 분석기 사용 방법

리소스 플릿을 모니터링하고 추적 메시지 수신을 시작하려면 다음 단계를 수행합니다.

1. 네트워크 분석기 구성 생성 및 리소스 추가

   추적 메시징을 활성화하려면 먼저 네트워크 분석기 구성을 생성하고 구성에 리소스를 추가합니다. 먼저 로그 수준 및 무선 디바이스 프레임 정보를 포함하는 구성 설정을 지정합니다. 그런 다음 무선 게이트웨이 및 무선 디바이스 식별자를 사용하여 모니터링할 무선 리소스를 추가합니다.

2. WebSockets를 사용하여 추적 메시지 스트리밍

   WebSocket 프로토콜로 IAM 역할의 자격 증명을 사용하여 미리 서명된 요청 URL을 생성하여 네트워크 분석기 추적 메시지를 스트리밍할 수 있습니다.

3. 추적 메시징 세션 활성화 및 추적 메시지 모니터링

   추적 메시지 수신을 시작하려면 추적 메시징 세션을 활성화합니다. 추가 비용이 발생하지 않도록 네트워크 분석기 추적 메시징 세션을 비활성화하거나 닫을 수 있습니다.

다음 동영상에서는 AWS IoT Core for LoRaWAN 네트워크 분석기의 작동 방법에 대해 설명하며 네트워크 분석기를 사용하여 리소스를 추가하고 조인 활동을 추적하는 프로세스를 안내합니다.

다음 주제는 구성을 생성하고 리소스를 추가하고 추적 메시징 세션을 활성화하는 방법을 보여 줍니다.

주제

네트워크 분석기에 필요한 IAM 역할 추가

네트워크 분석기를 사용하는 경우 네트워크 분석기 리소스에 액세스하기 위해 사용자에게 API 작업 UpdateNetworkAnalyzerConfiguration 및 GetNetworkAnalyzerConfiguration을 사용할 수 있는 권한을 부여해야 합니다. 다음은 권한을 부여하는 데 사용하는 IAM 정책을 보여줍니다.

네트워크 분석기용 IAM 정책

다음 중 하나를 사용하세요.

- 전체 액세스 무선 정책

  역할에 AWSIoTWirelessFullAccess 정책을 연결하여 AWS IoT Core for LoRaWAN에 전체 액세스 정책을 부여합니다. 자세한 내용은 AWSIoTWirelessFullAccess 정책 요약을 참조하세요.

- 가져오기 및 업데이트 API에 대한 범위가 지정된 IAM 정책

  IAM 콘솔의 정책 생성(Create policy)으로 이동하여 시각적 편집기(Visual editor) 탭에서 다음 IAM 정책을 생성합니다.

  1. 서비스(Service)에 IoTWireless를 선택합니다.
  2. 액세스 수준(Access level)에서 읽기(Read)를 확장하고 GetNetworkAnalyzerConfiguration을 선택한 후 쓰기(Write)를 확장하고 UpdateNetworkAnalyzerConfiguration을 선택합니다.
  3. 다음:태그(Next:Tags)를 선택하고 정책의 이름(Name)을 입력합니다(예: IoTWirelessNetworkAnalyzerPolicy. 정책 생성을 선택합니다.

  다음은 생성한 IoTWirelessNetworkAnalyzerPolicy 정책을 보여줍니다. IAM 정책 생성에 대한 자세한 내용은 IAM 정책 생성을 참조하세요.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

특정 리소스에 액세스할 수 있도록 범위가 지정된 정책

보다 세분화된 액세스 제어를 구성하려면 무선 게이트웨이와 디바이스를 리소스(Resource) 필드에 추가해야 합니다. 다음 정책은 와일드카드 ARN을 사용하여 모든 게이트웨이 및 디바이스에 대한 액세스 권한을 부여합니다. `WirelessGatewayId` 및 `WirelessDeviceId`를 사용하여 특정 게이트웨이 및 디바이스에 대한 액세스를 제어할 수 있습니다.

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

사용자에게 네트워크 분석기를 사용할 수 있지만 무선 게이트웨이나 디바이스는 사용할 수 없는 권한을 부여하려면 다음 정책을 사용합니다. 달리 지정하지 않는 한 리소스 사용 권한은 암시적으로 거부됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
```

```
            ],
            "Resource": [
                "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
            ]
        }
    ]
}
```

다음 단계

정책을 만들었으므로 네트워크 분석기 구성에 리소스를 추가하고 해당 리소스에 대한 추적 메시징 정보를 받을 수 있습니다. 자세한 내용은 네트워크 분석기 구성 생성 및 리소스 추가 단원을 참조하십시오.

네트워크 분석기 구성 생성 및 리소스 추가

추적 메시징을 스트리밍하려면 먼저 네트워크 분석기 구성을 생성하고 모니터링하려는 리소스를 이 구성에 추가합니다. 구성을 생성할 때 다음을 수행할 수 있습니다.

- 구성 이름을 지정하고 선택적으로 설명을 입력합니다.
- 로그 메시지의 프레임 정보 및 세부 수준 등의 구성 설정을 사용자 지정합니다.
- 모니터링하기를 원하는 리소스를 추가합니다. 리소스는 무선 디바이스 또는 무선 게이트웨이 또는 둘 다일 수 있습니다.

지정하는 구성 설정에 따라 구성에 추가하는 리소스에 대해 수신할 추적 메시징 정보가 결정됩니다. 모니터링 사용 사례에 따라 여러 구성을 만들 수도 있습니다.

다음은 구성을 생성하고 리소스를 추가하는 방법을 보여줍니다.

주제

- 네트워크 분석기 구성 생성
- 리소스 추가 및 네트워크 분석기 구성 업데이트

네트워크 분석기 구성 생성

무선 게이트웨이 또는 무선 디바이스를 모니터링하려면 먼저 네트워크 분석기 구성을 만들어야 합니다. 구성을 생성할 때 구성 이름만 지정하면 됩니다. 구성 설정을 생성한 후에도 구성 설정을 사용자 지정하고 모니터링할 리소스를 구성에 추가할 수 있습니다. 구성 설정에 따라 해당 리소스에 대해 수신할 추적 메시징 정보가 결정됩니다.

모니터링할 리소스와 해당 리소스에 대해 수신하려는 정보 수준에 따라 여러 구성을 만들 수 있습니다. 예를 들어, AWS 계정에 있는 특정 게이트웨이 집합에 대한 오류 정보만 표시하는 구성을 만들 수 있습니다. 모니터링할 무선 디바이스에 대한 모든 정보를 표시하는 구성을 만들 수도 있습니다.

다음 섹션에서는 다양한 구성 설정과 구성을 생성하는 방법을 보여줍니다.

구성 설정

네트워크 분석기 구성을 생성하거나 업데이트할 때 다음 파라미터를 사용자 지정하여 로그 스트림 정보를 필터링할 수도 있습니다.

- 프레임 정보(Frame info)

  이 설정은 추적 메시지의 무선 디바이스 리소스에 대한 프레임 정보입니다. 프레임 정보는 네트워크 서버와 최종 디바이스 간의 통신을 디버깅하는 데 사용할 수 있습니다. 기본적으로 활성화됩니다.

- 로그 수준

  정보 또는 오류 로그를 보거나 로깅을 해제할 수 있습니다.

  - 정보

    로그 수준이 정보(Info)인 로그는 더 상세하며 오류 로그 스트림과 정보 로그 스트림을 모두 포함합니다. 정보 로그를 사용하여 디바이스 또는 게이트웨이 상태의 변경 사항을 볼 수 있습니다.

    > **Note**
    >
    > 자세한 로그 스트림을 수집하면 추가 비용이 발생할 수 있습니다. 요금에 대한 자세한 내용은 AWS IoT Core 요금을 참조하십시오.

  - 오류

    로그 수준이 오류(Error)인 로그는 덜 상세하고 오류 정보만 표시합니다. 애플리케이션에 디바이스 연결 오류와 같은 오류가 있는 경우 이러한 로그를 사용할 수 있습니다. 로그 스트림의 정보를 사용하여 플릿의 리소스에 대한 오류를 식별하고 문제를 해결할 수 있습니다.

콘솔을 사용하여 구성 생성

AWS IoT 콘솔 또는 AWS IoT 무선 API를 사용하여 네트워크 분석기 구성을 생성하고 선택적으로 파라미터를 사용자 지정할 수 있습니다. 또한 여러 구성을 생성하고 나중에 사용하지 않는 구성을 삭제할 수 있습니다.

네트워크 분석기 구성 생성

1. AWS IoT 콘솔의 네트워크 분석기 허브를 열고 Create configuration(구성 생성)을 선택합니다.
2. 구성 설정을 지정합니다.
   - 이름, 설명 및 태그

     문자, 숫자, 하이픈 또는 밑줄만 포함하는 고유한 Configuration name(구성 이름)을 지정합니다. 선택 사항인 설명(Description) 필드를 사용하여 구성에 대한 정보를 제공하고 태그(Tags) 필드를 사용하여 구성에 대한 메타데이터의 키 값 쌍을 추가합니다. 리소스 이름 지정 및 설명에 대한 자세한 내용은 AWS IoT 무선 리소스 설명 단원을 참조하세요.
   - 구성 설정

     프레임 정보를 사용 중지할지 여부를 선택하고 로그 수준 선택(Select log levels)을 사용하여 추적 메시지 로그에 사용할 로그 수준을 선택합니다. 다음을 선택합니다.
3. 구성에 리소스 추가 지금 리소스를 추가하거나 생성(Create)을 선택한 후 나중에 리소스를 추가합니다. 나중에 리소스를 추가하려면 생성(Create)을 선택합니다.

   네트워크 분석기 허브(Network Analyzer hub) 페이지에서 생성한 구성과 설정을 볼 수 있습니다. 새 구성의 세부 정보를 보려면 구성 이름을 선택합니다.

네트워크 분석기 구성 삭제

모니터링할 리소스와 해당 리소스에 대해 수신하려는 추적 메시징 정보 수준에 따라 여러 개의 네트워크 분석기 구성을 만들 수 있습니다.

콘솔에서 구성을 제거하는 방법

1. AWS IoT 콘솔의 네트워크 분석기 허브로 이동하여 제거할 구성을 선택합니다.
2. 작업을 선택한 후 삭제를 선택합니다.

API를 사용하여 구성 생성

API를 사용하여 네트워크 분석기 구성을 만들려면 CreateNetworkAnalyzerConfiguration API 작업 또는 create-network-analyzer-configuration CLI 명령을 사용합니다.

구성을 생성할 때 구성 이름만 지정하면 됩니다. 또한 이 API 작업을 사용하여 구성 설정을 지정하고 구성을 생성할 때 리소스를 추가할 수 있습니다. 또는 나중에 UpdateNetworkAnalyzerConfiguration API 작업 또는 pdate-network-analyzer-configuration CLI 명령을 사용하여 지정할 수 있습니다.

- 구성 생성

  구성을 생성할 때 구성 이름을 지정해야 합니다. 예를 들어, 다음 명령은 이름과 선택적으로 설명만 제공하여 구성을 생성합니다. 기본적으로 구성에는 프레임 정보가 활성화되어 있으며 INFO 로그 수준을 사용합니다.

  ```
  aws iotwireless create-network-analyzer-configuration \
      --configuration-name My_Network_Analyzer_Config \
      --description "My first network analyzer configuration"
  ```

  이 명령을 실행하면 네트워크 분석기 구성의 ARN 및 ID가 표시됩니다.

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

- 리소스로 구성 생성

  구성 설정을 사용자 지정하려면 trace-content 파라미터를 사용하세요. 리소스를 추가하려면 WirelessDevices 및 WirelessGateways 파라미터를 사용하여 구성에 추가할 게이트웨이, 디바이스 또는 둘 다를 지정하세요. 예를 들어, 다음 명령은 구성 설정을 사용자 지정하고 구성에 WirelessGatewayID 및 WirelessDeviceID로 지정된 무선 리소스를 추가합니다.

  ```
  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"
  ```

  다음 예에서는 명령 실행의 출력을 보여줍니다.

  ```
  {
  ```

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

네트워크 분석기 구성 나열

모니터링할 리소스와 해당 리소스에 대해 수신하려는 추적 메시징 정보의 세부 수준에 따라 여러 개의 네트워크 분석기 구성을 만들 수 있습니다. 이러한 구성을 만든 후 ListNetworkAnalyzerConfigurations API 작업 또는 list-network-analyzer-configuration CLI 명령을 사용하여 해당 구성 목록을 가져옵니다.

```
aws iotwireless list-network-analyzer-configurations
```

이 명령을 실행하면 AWS 계정에 있는 모든 네트워크 분석기 구성이 표시됩니다. `max-results` 파라미터를 사용하여 표시할 구성의 개수를 지정할 수도 있습니다. 다음은 이 명령 실행의 출력을 보여줍니다.

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

네트워크 분석기 구성 삭제

DeleteNetworkAnalyzerConfiguration API 작업 또는 delete-network-analyzer-configuration CLI 명령을 사용하여 더 이상 사용하지 않는 구성을 삭제할 수 있습니다.

```
aws iotwireless delete-network-analyzer-configuration \
    --configuration-name My_NetworkAnalyzer_Config
```

이 명령을 실행하면 출력을 생성하지 않습니다. 사용 가능한 구성을 보려면 `ListNetworkAnalyzerConfigurations` API 작업을 사용하면 됩니다.

다음 단계

네트워크 분석기 구성을 만들었으므로 구성에 리소스를 추가하거나 구성 설정을 업데이트할 수 있습니다. 자세한 내용은 리소스 추가 및 네트워크 분석기 구성 업데이트 단원을 참조하십시오.

리소스 추가 및 네트워크 분석기 구성 업데이트

추적 메시징을 활성화하려면 먼저 구성에 리소스를 추가해야 합니다. 하나의 기본 네트워크 분석기 구성만 사용할 수 있습니다. AWS IoT Core for LoRaWAN은 NetworkAnalyzerConfig_Default라는 이름을 이 구성에 할당하며 이 필드는 편집할 수 없습니다. 이 구성은 콘솔에서 네트워크 분석기를 사용할 때 자동으로 AWS 계정에 추가됩니다.

이 기본 구성에 모니터링할 리소스를 추가할 수 있습니다. 리소스는 LoRaWAN 디바이스와 LoRaWAN 게이트웨이 중 하나 또는 둘 모두일 수 있습니다. 구성에 각 개별 리소스를 추가하려면 무선 게이트웨이 및 무선 디바이스 식별자를 사용합니다.

구성 설정

설정을 구성하려면 먼저 기본 구성에 리소스를 추가하고 추적 메시징을 활성화합니다. 추적 메시지 로그를 수신한 후 다음 파라미터를 사용자 지정하여 기본 구성을 업데이트하고 로그 스트림을 필터링할 수도 있습니다.

- 프레임 정보(Frame info)

  이 설정은 추적 메시지의 무선 디바이스 리소스의 프레임 정보입니다. 프레임 정보는 기본적으로 사용되며 네트워크 서버와 종단 디바이스 간의 통신을 디버깅하는 데 사용할 수 있습니다.

- 로그 수준

  정보 또는 오류 로그를 보거나 로깅을 해제할 수 있습니다.

  - 정보

    로그 수준이 정보(Info)인 로그는 더 상세하며 자세한 정보를 제공하고 오류가 들어 있는 로그 스트림을 포함합니다. 정보 로그를 사용하여 디바이스 또는 게이트웨이 상태의 변경 사항을 볼 수 있습니다.

> **Note**
>
> 자세한 로그 스트림을 수집하면 추가 비용이 발생할 수 있습니다. 요금에 대한 자세한 내용은 AWS IoT Core 요금을 참조하십시오.

- 오류

  로그 수준이 오류(Error)인 로그는 덜 상세하고 오류 정보만 표시합니다. 애플리케이션에 디바이스 연결 오류와 같은 오류가 있는 경우 이러한 로그를 사용할 수 있습니다. 로그 스트림의 정보를 사용하여 플릿의 리소스에 대한 오류를 식별하고 문제를 해결할 수 있습니다.

필수 조건

리소스를 추가하려면 먼저 모니터링할 게이트웨이 및 디바이스를 AWS IoT Core for LoRaWAN에 온보딩해야 합니다. 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 및 디바이스 연결 단원을 참조하십시오.

콘솔을 사용하여 리소스 추가 및 네트워크 분석기 구성 업데이트

AWS IoT 콘솔 또는 AWS IoT 무선 API를 사용하여 리소스를 추가하고 선택적 파라미터를 사용자 지정할 수 있습니다. 리소스 외에도 구성 설정을 편집하고 업데이트된 구성을 저장할 수도 있습니다.

구성에 리소스를 추가하려면(콘솔)

1. AWS IoT 콘솔의 네트워크 분석기 허브를 열고 네트워크 분석기 구성인 NetworkAnalyzerConfig_Default를 선택합니다.
2. 리소스 추가(Add resources)를 선택합니다.
3. 무선 게이트웨이 및 무선 디바이스 식별자를 사용하여 모니터링할 리소스를 추가합니다. 최대 250개의 무선 게이트웨이 또는 무선 디바이스를 추가할 수 있습니다. 리소스를 추가하려면
   a. 게이트웨이 보기(View gateways) 또는 디바이스 보기(View devices) 탭을 사용하여 AWS 계정에 추가한 게이트웨이 및 디바이스 목록을 봅니다.
   b. 모니터링하려는 디바이스나 게이트웨이의 `WirelessDeviceID` 또는 `WirelessGatewayID`를 복사하여 해당 리소스의 식별자 값을 입력합니다.
   c. 리소스를 계속 추가하려면 게이트웨이 추가(Add gateway) 또는 디바이스 추가(Add device)를 선택하고 무선 게이트웨이 또는 디바이스를 추가합니다. 더 이상 모니터링하지 않을 리소스를 추가한 경우 리소스 제거(Remove resource)를 선택합니다.

4. 리소스를 모두 추가한 다음 추가(Add)를 선택합니다.

   네트워크 분석기 허브 페이지(Network Analyzer hub page)에서 추가한 게이트웨이 및 디바이스 수를 확인할 수 있습니다. 추적 메시징 세션을 활성화할 때까지 게이트웨이와 디바이스를 계속 추가할 수 있습니다. 세션이 활성화된 후 리소스를 추가하려면 세션을 비활성화해야 합니다.

네트워크 분석기 구성을 편집하려면(콘솔)

네트워크 분석기 구성을 편집하고 추적 메시지 로그에 대한 프레임 정보 및 로그 수준을 사용 중지할지 여부를 선택할 수도 있습니다.

1. AWS IoT 콘솔의 네트워크 분석기 허브를 열고 네트워크 분석기 구성인 NetworkAnalyzerConfig_Default를 선택합니다.
2. 편집을 선택합니다.
3. 프레임 정보를 사용 중지할지 여부를 선택하고 로그 수준 선택(Select log levels)을 사용하여 추적 메시지 로그에 사용할 로그 수준을 선택합니다. Save(저장)를 선택합니다.

   네트워크 분석기 구성의 세부 정보 페이지에 지정한 구성 설정이 표시됩니다.

API를 사용하여 리소스 추가 및 네트워크 분석기 구성 업데이트

AWS IoT 무선 API 작업 또는 AWS IoT 무선 CLI 명령을 사용하여 리소스를 추가하고 네트워크 분석기 구성에 대한 구성 설정을 업데이트할 수 있습니다.

- 리소스를 추가하고 네트워크 분석기 구성을 업데이트하려면 UpdateNetworkAnalyzerConfiguration API 또는 update-network-analyzer-configuration CLI를 사용합니다.
  - 리소스 추가

    추가하려는 무선 디바이스의 경우 `WirelessDevicesToAdd`를 사용하여 디바이스의 `WirelessDeviceID`를 문자열 배열로 입력합니다. 추가하려는 무선 게이트웨이의 경우 `WirelessGatewaysToAdd`를 사용하여 게이트웨이의 `WirelessGatewayID`를 문자열 배열로 입력합니다.
  - 구성 편집

    네트워크 분석기 구성을 편집하려면 `TraceContent` 파라미터를 사용하여 `WirelessDeviceFrameInfo`가 `ENABLED` 또는 `DISABLED`여야 하는지 여부와 `LogLevel` 파라미터가 `INFO`, `ERROR` 또는 `DISABLED`여야 하는지 여부를 지정합니다.

```
{
   "TraceContent": {
      "LogLevel": "string",
      "WirelessDeviceFrameInfo": "string"
   },
   "WirelessDevicesToAdd": [ "string" ],
   "WirelessDevicesToRemove": [ "string" ],
   "WirelessGatewaysToAdd": [ "string" ],
   "WirelessGatewaysToRemove": [ "string" ]
}
```

- 추가한 구성 및 리소스에 대한 정보를 보려면 GetNetworkAnalyzerConfiguration API 작업 또는 get-network-analyzer-configuration 명령을 사용합니다. 네트워크 분석기 구성의 이름 `NetworkAnalyzerConfig_Default`를 입력으로 제공합니다.

다음 단계

리소스를 추가하고 구성에 대한 선택적 구성 설정을 지정했으므로 이제 WebSocket 프로토콜을 사용하여 네트워크 분석기를 사용하는 AWS IoT Core for LoRaWAN과의 연결을 설정할 수 있습니다. 그런 다음 추적 메시징을 활성화하고 리소스에 대한 추적 메시지 수신을 시작할 수 있습니다. 자세한 내용은 WebSocket을 사용하여 네트워크 분석기 추적 메시지 스트리밍 단원을 참조하십시오.

WebSocket을 사용하여 네트워크 분석기 추적 메시지 스트리밍

WebSocket 프로토콜을 사용하면 실시간으로 네트워크 분석기 추적 메시지를 스트리밍할 수 있습니다. 요청을 전송하면 서비스가 JSON 구조로 응답합니다. 추적 메시징을 활성화한 후 메시지 로그를 사용하여 리소스에 대한 정보를 보고 오류를 해결할 수 있습니다. 자세한 내용은 WebSocket 프로토콜을 참조하세요.

다음은 WebSocket을 사용하여 네트워크 분석기 추적 메시지를 스트리밍하는 방법을 보여줍니다.

주제

- WebSocket 라이브러리를 사용하여 미리 서명된 요청 생성
- WebSocket 메시지 및 상태 코드

WebSocket 라이브러리를 사용하여 미리 서명된 요청 생성

다음은 WebSocket 라이브러리를 사용하여 서비스에 요청을 전송할 수 있도록 미리 서명된 요청을 생성하는 방법을 보여줍니다.

IAM 역할에 WebSocket 요청에 대한 정책 추가

WebSocket 프로토콜을 사용하여 네트워크 분석기를 호출하려면 이 요청을 하는 AWS Identity and Access Management(IAM) 역할에 다음 정책을 연결합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotwireless:StartNetworkAnalyzerStream",
            "Resource": "*"
        }
    ]
}
```

미리 서명된 URL 생성

애플리케이션과 네트워크 분석기 간에 통신을 설정하는 데 필요한 정보가 포함되어 있는 WebSocket 요청에 대한 URL을 생성합니다. 요청의 자격 증명을 확인하려면 WebSocket 스트리밍에서는 Amazon 서명 버전 4 프로세스를 사용하여 요청에 서명합니다. 서명 버전 4에 대한 자세한 내용은 Amazon Web Services 일반 참조의 AWS API 요청에 서명을 참조하세요.

네트워크 분석기를 호출하려면 `StartNetworkAnalyzerStream` 요청 URL을 사용합니다. 요청은 앞에서 언급한 IAM 역할의 자격 증명을 사용하여 서명됩니다. URL은 가독성을 위해 줄 바꿈이 추가된 다음 형식입니다.

```
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
```

서명 버전 4 파라미터에 대해 다음 값을 사용합니다.

- X-Amz-Algorithm – 서명 프로세스에서 사용하는 알고리즘입니다. 유일한 유효 값은 `AWS4-HMAC-SHA256`입니다.

- X-Amz-Credential – 사용자의 액세스 키 ID와 자격 증명 범위 구성 요소를 연결해 형성한, 슬래시('/')로 구분한 문자열입니다. 자격 증명 범위에는 YYYYMMDD 형식의 날짜, AWS 리전, 서비스 이름, 종료 문자열(aws4_request)이 포함됩니다.
- X-Amz-Date – 서명이 생성된 날짜 및 시간입니다. Amazon Web Services 일반 참조의 서명 버전 4에서 날짜 처리의 지침에 따라 날짜와 시간을 생성합니다.
- X-Amz-Expires – 자격 증명이 만료될 때까지의 기간(초)입니다. 최대값은 300초(5분)입니다.
- X-Amz-Security-Token – (선택 사항) 임시 자격 증명을 위한 서명 버전 4 토큰입니다. 이 파라미터를 지정하는 경우 표준 요청에 포함합니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 임시 보안 자격 증명 요청 섹션을 참조하세요.
- X-Amz-Signature – 요청에 대해 생성한 서명 버전 4 서명입니다.
- X-Amz-SignedHeaders – 요청에 대한 서명을 생성할 때 서명한 헤더입니다. 유일한 유효 값은 `host`입니다.

요청 URL 구성 및 서명 버전 4 서명 생성

요청에 대한 URL과 서명 버전 4 서명을 생성하려면 다음 단계를 따르십시오. 유사 코드의 예제입니다.

작업 1: 표준 요청 생성

요청의 정보가 포함된 문자열을 표준화된 형식으로 생성합니다. 그러면 AWS에서 요청을 수신할 때 사용자가 작업 3: 서명 계산에서 계산한 것과 동일한 서명을 계산할 수 있습니다. 자세한 내용은 Amazon Web Services 일반 참조의 서명 버전 4에 대한 표준 요청 생성을 참조하세요.

1. 애플리케이션의 요청에 대한 변수를 정의합니다.

```
# 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. 정식 URI(Uniform Resource Identifier)를 생성합니다. 표준 URI는 도메인과 쿼리 문자열 간의 URI 부분입니다.

```
canonical_uri = "/start-network-analyzer-stream"
```

3. 표준 헤더 및 서명된 헤더를 생성합니다. 표준 헤더의 후행 \n에 유의하세요.

   - 소문자 헤더 이름과 콜론을 차례대로 추가합니다.
   - 헤더에 대한 쉼표로 구분된 값 목록을 추가합니다. 여러 값을 가진 헤더에서 값을 정렬하지 마세요.
   - 새 줄(\n)을 추가합니다.

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. 이 알고리즘을 해싱 알고리즘과 일치시킵니다. 이때 SHA-256을 사용해야 합니다.

```
algorithm = "AWS4-HMAC-SHA256"
```

5. 생성된 키의 범위를 요청이 수행된 날짜, 리전 및 서비스로 지정하는 자격 증명 범위를 생성합니다.

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. 표준 쿼리 문자열을 생성합니다. 쿼리 문자열 값은 URL로 인코딩되어야 하며 이름을 기준으로 정렬되어야 합니다.

   - 문자 코드 포인트를 기준으로 파라미터 이름을 오름차순으로 정렬합니다. 중복된 이름을 가진 파라미터는 값별로 정렬해야 합니다. 예를 들어 대문자 F로 시작하는 파라미터 이름 앞에 소문자 b로 시작하는 파라미터 이름이 옵니다.
   - 예약되지 않은 문자는 URI로 인코딩하지 않습니다. RFC 3986에 정의된 예약되지 않은 문자는 A-Z, a-z, 0-9, 하이픈(-), 밑줄(_), 마침표(.) 및 물결표(~)입니다.
   - %XY와 같이 모든 기타 문자를 퍼센트 인코딩합니다. 여기서 X 및 Y는 16진 문자(0 ~ 9 및 대문자 A ~ F)입니다. 예를 들어 공백 문자는 %20(일부 인코딩 구조인 +가 아님)로 인코딩되고, 확장 UTF-8 문자는 %XY%ZA%BC 형식이어야 합니다.
   - 매개변수 값에서 등호(=) 문자를 두 번 인코딩합니다.

```
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. 페이로드의 해시를 생성합니다. GET 요청의 경우 페이로드는 빈 문자열입니다.

```
payload_hash = HashSHA256(("").Encode("utf-8")).HexDigest()
```

8. 모든 요소를 결합하여 표준 요청을 생성합니다.

```
canonical_request = method + '\n'
   + canonical_uri + '\n'
   + canonical_querystring + '\n'
   + canonical_headers + '\n'
   + signed_headers + '\n'
   + payload_hash
```

작업 2: 서명할 문자열 생성

서명할 문자열에는 요청에 대한 메타 정보가 포함되어 있습니다. 요청 서명을 계산할 때 다음 단계의 서명할 문자열을 사용합니다. 자세한 내용은 Amazon Web Services 일반 참조의 서명 버전 4에 대한 서명할 문자열 생성을 참조하세요.

```
string_to_sign=algorithm + "\n"
   + amz_date + "\n"
   + credential_scope + "\n"
   + HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

작업 3: 서명 계산

AWS 보안 액세스 키에서 서명 키를 생성합니다. 생성된 키는 더 높은 수준의 보호를 위해 날짜, 서비스 및 AWS 리전에 고유합니다. 생성된 키를 사용하여 요청에 서명합니다. 자세한 내용은 Amazon Web Services 일반 참조의 AWS 서명 버전 4에 대한 서명 계산을 참조하세요.

이 코드는 `GetSignatureKey` 함수를 구현하여 서명 키를 생성했다고 가정합니다. 자세한 내용과 예제 함수는 Amazon Web Services 일반 참조의 서명 버전 4에 대한 서명 키 생성 방법을 보여주는 예를 참조하세요.

`HMAC(key, data)` 함수는 결과를 이진 형식으로 반환하는 HMAC-SHA256 함수를 나타냅니다.

```
#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
```

작업 4: 요청에 서명 정보 추가 및 요청 URL 생성

서명을 계산한 후 쿼리 문자열에 추가합니다. 자세한 내용은 Amazon Web Services 일반 참조의 요청에 서명 추가를 참조하세요.

```
#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
```

다음 단계

이제 WebSocket 라이브러리와 함께 요청 URL을 사용하여 서비스에 요청하고 메시지를 관찰할 수 있습니다. 자세한 내용은 WebSocket 메시지 및 상태 코드 단원을 참조하십시오.

WebSocket 메시지 및 상태 코드

미리 서명된 요청을 생성한 후 WebSocket 라이브러리 또는 프로그래밍 언어에 적합한 라이브러리와 함께 요청 URL을 사용하여 서비스에 요청할 수 있습니다. 이 미리 서명된 요청을 생성하는 방법에 대한 자세한 내용은 WebSocket 라이브러리를 사용하여 미리 서명된 요청 생성 섹션을 참조하세요.

WebSocket 메시지

WebSocket 프로토콜을 사용하여 양방향 연결을 설정할 수 있습니다. 메시지는 클라이언트에서 서버로, 서버에서 클라이언트로 전송될 수 있습니다. 그러나 네트워크 분석기는 서버에서 클라이언트로 전송하는 메시지만 지원합니다. 클라이언트로부터 받은 모든 메시지는 예상치 못한 것이며 클라이언트로부터 메시지를 받으면 서버는 자동으로 WebSocket 연결을 닫습니다.

요청이 수신되고 추적 메시징 세션이 시작되면 서버는 페이로드인 JSON 구조로 응답합니다. 페이로드 및 AWS Management Console에서 추적 메시징을 활성화하는 방법에 대한 자세한 내용은 네트워크 분석기 추적 메시지 로그 실시간 보기 및 모니터링 섹션을 참조하세요.

WebSocket 상태 코드

다음은 서버에서 클라이언트로의 통신을 위한 WebSocket 상태 코드를 보여줍니다. WebSocket 상태 코드는 연결의 정상 폐쇄에 대한 RFC 표준을 따릅니다.

다음은 지원되는 상태 코드를 보여줍니다.

- 1000

  이 상태 코드는 정상적인 폐쇄를 나타내며, 이는 WebSocket 연결이 설정되고 요청이 이행되었음을 의미합니다. 세션이 유휴 상태여서 연결 시간 초과가 발생할 때 이 상태가 관찰될 수 있습니다.
- 1002

  이 상태 코드는 프로토콜 오류로 인해 엔드포인트가 연결을 종료하고 있음을 나타냅니다.
- 1003

  이 상태 코드는 엔드포인트가 수락할 수 없는 형식의 데이터를 수신하여 연결을 종료한 오류 상태를 나타냅니다. 엔드포인트는 텍스트 데이터만 지원하며 지원되지 않는 형식을 사용하는 클라이언트로부터 이진 메시지 또는 메시지를 수신하는 경우 이 상태 코드를 표시할 수 있습니다.
- 1008

  이 상태 코드는 엔드포인트가 정책을 위반하는 메시지를 수신하여 연결을 종료한 오류 상태를 나타냅니다. 이 상태는 일반적이며 1003 또는 1009와 같은 다른 상태 코드가 적용되지 않을 때 표시됩니다. 정책을 숨겨야 하거나 만료된 서명과 같은 권한 부여 실패가 있는 경우에도 이 상태가 표시됩니다.
- 1011

  이 상태 코드는 서버가 요청을 이행하지 못하게 하는 예기치 않은 조건이나 내부 오류가 발생하여 연결을 종료하는 오류 상태를 나타냅니다.

다음 단계

미리 서명된 요청을 생성하는 방법과 WebSocket 연결을 사용하여 서버에서 메시지를 관찰하는 방법을 배웠으므로 이제 추적 메시징을 활성화하고 무선 게이트웨이 및 무선 디바이스 리소스에 대한 메시지 로그 수신을 시작할 수 있습니다. 자세한 내용은 네트워크 분석기 추적 메시지 로그 실시간 보기 및 모니터링 단원을 참조하십시오.

네트워크 분석기 추적 메시지 로그 실시간 보기 및 모니터링

네트워크 분석기 구성에 리소스를 추가한 경우 추적 메시징을 활성화하여 리소스에 대한 추적 메시지 수신을 시작할 수 있습니다. AWS Management Console, AWS IoT 무선 API 또는 AWS CLI를 사용할 수 있습니다.

필수 조건

네트워크 분석기를 사용하여 추적 메시징을 활성화하려면 먼저 다음을 수행해야 합니다.

- 이 기본 네트워크 분석기 구성에 모니터링할 리소스를 추가합니다. 자세한 내용은 리소스 추가 및 네트워크 분석기 구성 업데이트 단원을 참조하십시오.
- `StartNetworkAnalyzerStream` 요청 URL을 사용하여 미리 서명된 요청을 생성합니다. 이 요청을 하는 AWS Identity and Access Management 역할의 자격 증명을 사용하여 요청이 서명됩니다. 자세한 내용은 미리 서명된 URL 생성 단원을 참조하십시오.

콘솔을 사용하여 추적 메시지 활성화

추적 메시징을 활성화하려면

1. AWS IoT 콘솔의 네트워크 분석기 허브를 열고 네트워크 분석기 구성인 NetworkAnalyzerConfig_Default를 선택합니다.
2. 네트워크 분석기 구성의 세부 정보 페이지에서 추적 메시징 활성화(Activate trace messaging)를 선택한 다음 활성화(Activate)를 선택합니다.

   최신 추적 메시지가 콘솔에서 가장 먼저 나타나는 추적 메시지 수신을 시작합니다.

> **Note**
>
> 메시징 세션이 시작된 후 추적 메시지를 수신하면 세션을 비활성화하거나 추적 세션에서 나갈 때까지 추가 비용이 발생할 수 있습니다. 요금에 대한 자세한 내용은 AWS IoT Core 요금을 참조하십시오.

추적 메시지 보기 및 모니터링

추적 메시징을 활성화하면 WebSocket 연결이 설정되고 추적 메시지가 실시간으로 최신 항목부터 나타나기 시작합니다. 각 페이지에 표시할 추적 메시지 수를 지정하고 각 메시지에 대한 관련 필드만 표시하도록 기본 설정을 사용자 지정할 수 있습니다. 예를 들어, 로그 수준(Log level)이 ERROR로 설정된 무선 게이트웨이 리소스에 대한 로그만 표시하도록 추적 메시지 로그를 사용자 지정할 수 있으므로 게이트웨이의 오류를 빠르게 식별하고 디버그할 수 있습니다. 추적 정보 메시지에는 다음 정보가 포함됩니다.

- 메시지 번호(Message Number): 가장 먼저 수신된 마지막 메시지를 표시하는 고유 번호입니다.
- 리소스 ID(Resource ID): 리소스의 무선 게이트웨이 또는 무선 디바이스 ID입니다.
- 타임스탬프(Timestamp): 메시지가 수신된 시간입니다.
- 메시지 ID: AWS IoT Core for LoRaWAN이 수신된 각 메시지에 할당하는 식별자입니다.
- FPort: WebSocket 연결을 사용하여 디바이스와 통신하기 위한 주파수 포트입니다.
- DevEui: 무선 디바이스의 확장 고유 식별자(EUI)입니다.
- 리소스(Resource): 모니터링되는 리소스가 무선 디바이스인지 또는 무선 게이트웨이인지 여부입니다.
- 이벤트: 무선 디바이스에 대한 로그 메시지의 이벤트로 조인(Join), 리조인(Rejoin), Uplink_Data, Downlink_Data 또는 등록(Registration)일 수 있습니다.
- 로그 수준(Log level): 디바이스의 INFO 또는 ERROR 로그 스트림에 대한 정보입니다.

네트워크 분석기 JSON 로그 메시지

한 번에 하나의 추적 메시지를 선택하여 해당 메시지에 대한 JSON 페이로드를 볼 수도 있습니다. 추적 메시지 로그에서 선택한 메시지에 따라 CustomerLog 및 LoRaFrame의 두 부분이 포함되어 있음을 나타내는 정보가 JSON 페이로드에 표시됩니다.

CustomerLog

JSON의 CustomerLog 부분은 메시지를 수신한 리소스의 유형 및 식별자, 로그 수준 및 메시지 내용을 표시합니다. 다음 예제에서는 CustomerLog 로그 메시지를 보여줍니다. JSON의 message 필드를 사용하여 오류 및 해결 방법에 대한 자세한 정보를 얻을 수 있습니다.

LoRaFrame

JSON의 LoRaFrame 부분에는 메시지 ID(Message ID)가 있으며 디바이스의 물리적 페이로드 및 무선 메타데이터에 대한 정보가 포함되어 있습니다.

다음은 추적 정보 메시지의 구조를 보여줍니다.

```
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;
  },
};
```

검토 및 다음 단계

이 섹션에서는 추적 메시지를 보고 정보를 사용하여 오류를 디버깅하는 방법을 배웠습니다. 모든 메시지를 본 후 다음 작업을 수행할 수 있습니다.

- 추적 메시징 비활성화

  추가 비용이 발생하지 않도록 네트워크 분석기 추적 메시징 세션을 비활성화할 수 있습니다. 세션을 비활성화하면 WebSocket 연결이 끊어지므로 추가 추적 메시지가 수신되지 않습니다. 콘솔에서 기존 메시지를 계속 볼 수 있습니다.
- 구성에 대한 프레임 정보 편집

네트워크 분석기 구성을 편집하고 프레임 정보를 비활성화할지 여부를 선택하고 메시지의 로그 수준을 선택할 수 있습니다. 구성을 업데이트하기 전에 추적 메시징 세션을 비활성화하는 것이 좋습니다. 이렇게 편집하려면 AWS IoT 콘솔의 네트워크 분석기 세부 정보 페이지를 열고 편집(Edit)을 선택합니다. 그런 다음 새 구성 설정으로 구성을 업데이트하고 추적 메시징을 활성화하여 업데이트된 메시지를 볼 수 있습니다.

- 구성에 리소스 추가

  또한 더 많은 리소스를 추가하고 네트워크 분석기 구성에 실시간으로 모니터링할 수 있습니다. 총 250개의 무선 게이트웨이 및 무선 디바이스 리소스를 결합하여 추가할 수 있습니다. 리소스를 추가하려면 AWS IoT 콘솔의 네트워크 분석기 세부 정보 페이지에서 리소스(Resources) 탭을 선택하고 리소스 추가(Add resources)를 선택합니다. 그런 다음 새 리소스로 구성을 업데이트하고 추적 메시징을 활성화하여 추가 리소스에 대한 업데이트된 메시지를 볼 수 있습니다.

구성 설정을 편집하고 리소스를 추가하여 네트워크 분석기 구성을 업데이트하는 방법에 대한 자세한 내용은 리소스 추가 및 네트워크 분석기 구성 업데이트 섹션을 참조하세요.

네트워크 분석기를 사용하여 멀티캐스트 그룹 및 FUOTA 작업 디버깅 및 문제 해결

모니터링할 수 있는 무선 리소스에는 LoRaWAN 디바이스, LoRaWAN 게이트웨이 및 멀티캐스트 그룹이 포함됩니다. 또한 네트워크 분석기를 사용하여 FUOTA 작업과 관련된 문제를 디버깅하고 해결할 수 있습니다. 또한 FUOTA 작업이 진행 중일 때 설정, 데이터 전송 및 상태 쿼리와 관련된 메시지를 모니터링하고 추적할 수 있습니다.

FUOTA 작업을 모니터링하려면 작업에 멀티캐스트 그룹이 포함된 경우 멀티캐스트 그룹과 그룹 내 디바이스를 모두 네트워크 분석기 구성에 추가해야 합니다. 또한 FUOTA 태스크가 진행되는 동안 멀티캐스트 그룹 및 디바이스와 교환되는 유니캐스트 및 멀티캐스트 업링크 및 다운링크 메시지를 추적하려면 프레임 정보와 멀티캐스트 프레임 정보를 활성화해야 합니다.

멀티캐스트 그룹을 모니터링하려면 네트워크 분석기 구성에 그룹을 추가하고 멀티캐스트 프레임 정보를 사용하여 이러한 그룹으로 전송되는 멀티캐스트 다운링크 메시지 문제를 해결할 수 있습니다. 유니캐스트 통신이 사용되는 그룹에 조인하려는 디바이스의 문제를 해결하려면 이러한 디바이스도 네트워크 분석기 구성에 포함해야 합니다. 그룹 내 디바이스와의 유니캐스트 통신만 모니터링하려면 무선 디바이스의 프레임 정보를 활성화하세요. 이 접근 방식을 사용하면 멀티캐스트 그룹과 그룹에 조인하는 디바이스 모두에 대한 포괄적인 모니터링 및 진단이 가능합니다.

다음 섹션에서는 네트워크 분석기를 사용하여 멀티캐스트 그룹 및 FUOTA 작업을 디버깅하고 문제를 해결하는 방법을 설명합니다.

주제

디바이스만 포함된 FUOTA 태스크 디버깅

네트워크 분석기를 사용하여 LoRaWAN 디바이스만 작업에 추가된 FUOTA 작업을 디버깅할 수 있습니다. FUOTA 작업에 디바이스를 추가하는 방법에 대한 자세한 내용은 FUOTA 태스크에 디바이스 및 멀티캐스트 그룹 추가 및 FUOTA 세션 예약 섹션을 참조하세요. FUOTA 작업을 디버깅하려면 다음 단계를 수행하세요.

1. 태스크가 진행되는 동안 디바이스와 교환되는 FUOTA 업링크 및 다운링크 메시지를 모니터링할 수 있도록 무선 디바이스의 프레임 정보를 활성화하여 네트워크 분석기 구성을 생성합니다.
2. 무선 디바이스 식별자를 사용하여 FUOTA 작업의 디바이스를 네트워크 분석기 구성에 추가합니다.
3. 추적 메시징을 활성화하여 네트워크 분석기 구성에 있는 디바이스에 대한 추적 메시지 수신을 시작합니다.

추적 메시지 정보 `applicationCommandType` 열에서 데이터 전송 및 조각화 설정과 관련된 유니캐스트 다운링크 메시지 수신을 시작합니다.

> **Note**
>
> 추적 메시지 표에 `applicationCommandType` 열이 표시되지 않는 경우 테이블 설정을 조정하여 이 열을 표시할 수 있습니다.

WirelessMetadata > ApplicationInfo의 JSON 로그 메시지에서 `applicationCommandType` 및 기타 세부 메시지를 볼 수도 있습니다.

멀티캐스트 그룹을 사용하여 FUOTA 작업 디버깅

네트워크 분석기를 사용하여 멀티캐스트 그룹 및 LoRaWAN 디바이스가 그룹에 추가된 FUOTA 작업을 디버깅할 수 있습니다. FUOTA 작업에 디바이스를 추가하는 방법에 대한 자세한 내용은 FUOTA 태스크에 디바이스 및 멀티캐스트 그룹 추가 및 FUOTA 세션 예약 섹션을 참조하세요. FUOTA 작업을 디버깅하려면 다음 단계를 수행하세요.

1. 무선 디바이스 및 멀티캐스트 그룹의 프레임 정보 및 멀티캐스트 프레임 정보 설정을 활성화하여 네트워크 분석기 구성을 생성합니다.
2. 멀티캐스트 그룹 식별자를 사용하여 FUOTA 작업의 멀티캐스트 그룹을 네트워크 분석기 구성에 추가합니다. 멀티캐스트 프레임 정보를 활성화하면 FUOTA 작업이 진행되는 동안 그룹에 전송되는 펌웨어 데이터 메시지와 FUOTA 상태 쿼리 메시지를 디버깅할 수 있습니다.
3. 무선 디바이스 식별자를 사용하여 멀티캐스트 그룹의 디바이스를 네트워크 분석기 구성에 추가합니다. 프레임 정보를 활성화하면 FUOTA 작업이 진행되는 동안 디바이스와 직접 교환되는 업링크 및 다운링크 메시지를 모니터링할 수 있습니다.
4. 추적 메시징을 활성화하여 네트워크 분석기 구성에 있는 디바이스 및 멀티캐스트 그룹에 대한 추적 메시지 수신을 시작합니다.

그런 다음 추적 메시지 표의 `applicationCommandType` 열과 디바이스만 포함된 FUOTA 태스크 디버깅에 설명된 JSON 로그 메시지의 세부 정보를 사용하여 추적 메시지를 보고 디버깅할 수 있습니다.

멀티캐스트 그룹에 조인하려는 디바이스 디버깅

네트워크 분석기를 사용하여 멀티캐스트 그룹에 조인하려는 디바이스를 디버깅할 수 있습니다. 멀티캐스트 그룹에 디바이스를 추가하는 방법에 대한 자세한 내용은 멀티캐스트 그룹 생성 및 그룹에 디바이스 추가 섹션을 참조하세요. 멀티캐스트 그룹을 디버깅하려면 다음 단계를 수행하세요.

1. 무선 디바이스의 프레임 정보를 활성화하여 네트워크 분석기 구성을 생성합니다.
2. 무선 디바이스 식별자를 사용하여 모니터링할 디바이스를 네트워크 분석기 구성에 추가합니다.
3. 추적 메시징을 활성화하여 네트워크 분석기 구성에 있는 디바이스에 대한 추적 메시지 수신을 시작합니다.
4. 그룹 내 디바이스에 대해 추적 메시징이 활성화된 후 멀티캐스트 그룹에 디바이스를 연결하기 시작합니다.

멀티캐스트 그룹 세션 디버깅

네트워크 분석기를 사용하여 멀티캐스트 그룹 세션을 디버깅할 수 있습니다. 자세한 내용은 멀티캐스트 그룹의 디바이스로 전송하도록 다운링크 메시지 예약 단원을 참조하십시오. 멀티캐스트 그룹 세션을 디버깅하려면 다음 단계를 수행하세요.

1. 멀티캐스트 그룹의 멀티캐스트 프레임 정보를 활성화하여 네트워크 분석기 구성을 생성합니다.
2. 멀티캐스트 그룹 식별자를 사용하여 모니터링할 멀티캐스트 그룹을 네트워크 분석기 구성에 추가합니다.
3. 멀티캐스트 세션이 시작되기 전에 추적 메시징을 활성화하여 멀티캐스트 그룹 세션에 대한 추적 메시지 수신을 시작합니다.
4. 멀티캐스트 그룹 세션을 시작하고 추적 메시지 테이블에 표시된 메시지와 JSON 로그 메시지를 확인하여 상태를 모니터링합니다.

추적 메시지 테이블의 `DevAddr` 열에 `MulticastAddr`이 표시됩니다. JSON 로그 메시지의 WirelessMetadata > ApplicationInfo에서 `MulticastGroupId`와 같은 세부 메시지를 볼 수도 있습니다.

AWS IoT Core for LoRaWAN 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

공용 인터넷을 통해 연결하는 대신 Virtual Private Cloud(VPC)의 Interface VPC 엔드포인트(AWS PrivateLink)를 통해 AWS IoT Core for LoRaWAN에 직접 연결할 수 있습니다. VPC 인터페이스 엔드포인트를 사용하는 경우 VPC와 AWS IoT Core for LoRaWAN 간의 통신은 AWS 네트워크에서 완전하고 안전하게 수행됩니다.

AWS IoT Core for LoRaWAN은 AWS PrivateLink에서 구동되는 Amazon Virtual Private Cloud 인터페이스 엔드포인트를 지원합니다. 각 VPC 엔드포인트는 하나 이상의 탄력적 네트워크 인터페이스 및 VPC 서브넷의 프라이빗 IP 주소로 표현됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 VPC 엔드포인트(AWS PrivateLink)를 참조하세요.

VPC 및 엔드포인트에 대한 자세한 내용은 Amazon VPC란 무엇입니까를 참조하세요.

AWS PrivateLink에 대한 자세한 내용은 AWS PrivateLink 및 VPC 엔드포인트를 참조하세요.

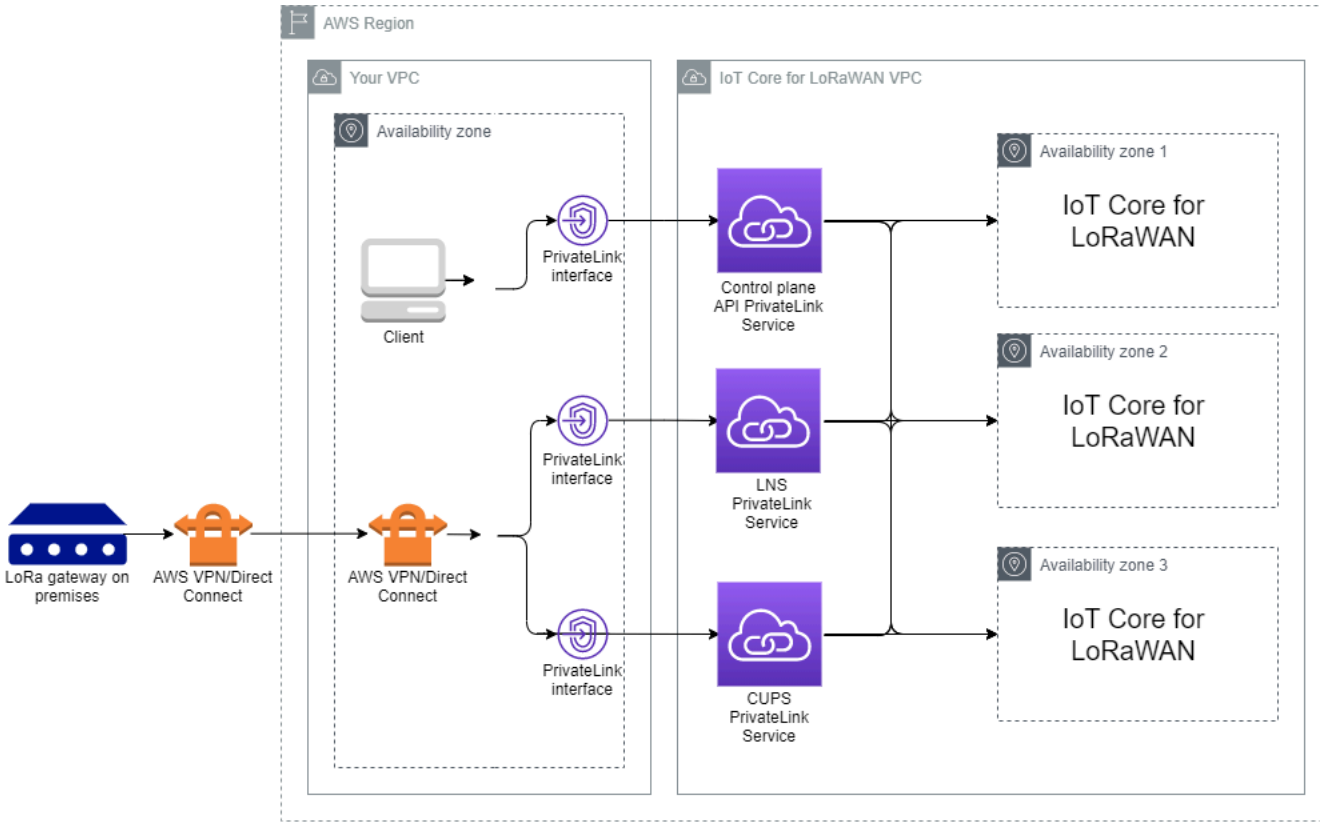
AWS IoT Wireless VPC 엔드포인트에 대한 고려 사항

AWS IoT Wireless용 인터페이스 VPC 엔드포인트를 설정하기 전에 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 속성 및 제한 사항을 검토해야 합니다.

AWS IoT Wireless는 VPC에서 모든 API 작업에 대한 호출 수행을 지원합니다. AWS IoT Wireless에서는 VPC 엔드포인트 정책이 지원되지 않습니다. 기본적으로 엔드포인트를 통해 AWS IoT Wireless에 대한 전체 액세스가 허용됩니다. 자세한 정보는 Amazon VPC 사용 설명서의 VPC 엔드포인트를 통해 서비스에 대한 액세스 제어 섹션을 참조하세요.

AWS IoT Core for LoRaWAN privatelink 아키텍처

다음 다이어그램은 AWS IoT Core for LoRaWAN의 프라이빗 링크 아키텍처를 보여줍니다. 이 아키텍처는 Transit Gateway 및 Route 53 Resolver를 사용하여 VPC, AWS IoT Core for LoRaWAN VPC 및 온프레미스 환경 간에 AWS PrivateLink 인터페이스 엔드포인트를 공유할 수 있습니다. VPC 인터페이스 엔드포인트에 대한 연결을 설정할 때 더 자세한 아키텍처 다이어그램을 확인할 수 있습니다.



AWS IoT Core for LoRaWAN 엔드포인트

AWS IoT Core for LoRaWAN에는 세 개의 퍼블릭 엔드포인트가 있습니다. 각 퍼블릭 엔드포인트에는 해당하는 VPC 인터페이스 엔드포인트가 있습니다. 퍼블릭 엔드포인트는 제어 영역 및 데이터 영

역 엔드포인트로 분류될 수 있습니다. 이러한 엔드포인트에 대한 자세한 내용은 AWS IoT Core for LoRaWAN API 엔드포인트를 참조하세요.

- 제어 영역 API 엔드포인트

  제어 영역 API 엔드포인트를 사용하여 AWS IoT 무선 API와 상호 작용할 수 있습니다. 이러한 엔드포인트는 AWS PrivateLink를 사용하여 Amazon VPC에서 호스팅되는 클라이언트에서 액세스할 수 있습니다.
- 데이터 영역 API 엔드포인트

  데이터 영역 API 엔드포인트는 LoRaWAN Network Server(LNS) 및 Configuration and Update Server(CUPS) 엔드포인트이며, 이 엔드포인트를 사용하여 AWS IoT Core for LoRaWAN LNS 및 CUPS 엔드포인트와 상호 작용할 수 있습니다. 이러한 엔드포인트는 AWS VPN 또는 AWS Direct Connect를 사용하여 온프레미스의 LoRa 게이트웨이에서 액세스할 수 있습니다. 게이트웨이를 AWS IoT Core for LoRaWAN에 온보딩할 때 이러한 엔드포인트를 얻습니다. 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 추가 단원을 참조하십시오.

주제

- AWS IoT Core for LoRaWAN 제어 영역 API 엔드포인트 온보딩
- AWS IoT Core for LoRaWAN 데이터 영역 API 엔드포인트 온보딩

AWS IoT Core for LoRaWAN 제어 영역 API 엔드포인트 온보딩

AWS IoT Core for LoRaWAN 제어 영역 API 엔드포인트를 사용하여 AWS IoT 무선 API와 상호 작용할 수 있습니다. 예를 들어 이 엔드포인트를 사용하여 SendDataToWirelessDevice API를 실행해 AWS IoT에서 LoRaWAN 디바이스로 데이터를 전송할 수 있습니다. 자세한 내용은 AWS IoT Core for LoRaWAN 제어 영역 API 엔드포인트를 참조하세요.

Amazon VPC에서 호스팅되는 클라이언트를 사용하여 AWS PrivateLink에서 구동되는 제어 영역 엔드포인트에 액세스할 수 있습니다. 이러한 엔드포인트는 공용 인터넷을 통해 연결하는 대신 Virtual Private Cloud(VPC)의 인터페이스 엔드포인트를 통해 AWS IoT 무선 API에 연결하는 데 사용할 수 있습니다.

제어 영역 엔드포인트를 온보딩하려면:

- Amazon VPC 및 서브넷 생성
- 서브넷에서 Amazon EC2 인스턴스 시작
- Amazon VPC 인터페이스 엔드포인트 생성

- 엔드포인트 인터페이스와의 연결 테스트

Amazon VPC 및 서브넷 생성

인터페이스 엔드포인트에 연결하려면 먼저 VPC 및 서브넷을 생성해야 합니다. 그런 다음 서브넷에서 EC2 인스턴스를 시작하여 인터페이스 엔드포인트에 연결하는 데 사용할 수 있습니다.

VPC를 생성하려면:

1. Amazon VPC 콘솔의 VPC 페이지로 이동하여 VPC 생성을 선택합니다.
2. VPC 생성 페이지에서:
   - VPC 이름 태그 - 선택 사항에 이름(예: **VPC-A**)을 입력합니다.
   - IPv4 CIDR 블록에서 VPC의 IPv4 주소 범위(예: **10.100.0.0/16**)를 입력합니다.
3. 다른 필드에 대한 기본값을 유지하고 VPC 생성을 선택합니다.

서브넷을 생성하려면:

1. Amazon VPC 콘솔의 서브넷 페이지로 이동하여 서브넷 생성을 선택합니다.
2. 서브넷 생성 페이지에서:
   - VPC ID에서, 앞서 생성한 VPC(예: VPC-A)를 선택합니다.
   - 서브넷 이름에 이름(예: **Private subnet**)을 입력합니다.
   - 서브넷의 가용 영역을 선택합니다.
   - IPv4 CIDR 블록에 서브넷의 IP 주소 블록(예: **10.100.0.0/24**)을 CIDR 형식으로 입력합니다.
3. 서브넷을 생성하여 VPC에 추가하려면 서브넷 생성을 선택합니다.

자세한 내용은 VPC 및 서브넷 관련 작업을 참조하세요.

서브넷에서 Amazon EC2 인스턴스 시작

EC2 인스턴스를 시작하려면:

1. Amazon EC2 콘솔로 이동하여 인스턴스 시작을 선택합니다.
2. AMI의 경우 Amazon Linux 2 AMI(HVM), SSD 볼륨 유형을 선택한 다음 t2 micro 인스턴스 유형을 선택합니다. 인스턴스 세부 정보를 구성하려면 다음을 선택합니다.
3. 인스턴스 세부 정보 구성 페이지에서:

- 네트워크에서, 앞서 생성한 VPC(예: VPC-A)를 선택합니다.
- 서브넷에서, 앞서 생성한 서브넷(예: **Private subnet**)을 선택합니다.
- IAM 역할(IAM role)에서 AWSIoTWirelessFullAccess 역할을 선택하여 AWS IoT Core for LoRaWAN 전체 액세스 정책을 부여합니다. 자세한 내용은 AWSIoTWirelessFullAccess 정책 요약을 참조하세요.
- 프라이빗 IP 가정(Assume Private IP)에서 IP 주소(예: 10.100.0.42)를 사용합니다.

4. 다음: 스토리지 추가를 선택한 후 다음: 태그 추가를 선택합니다. 필요에 따라 EC2 인스턴스에 연결하기 위해 태그를 추가할 수 있습니다. 다음: 보안 그룹 구성(Next: Configure Security Group)을 선택합니다.
5. 보안 그룹 구성 페이지에서 다음을 허용하도록 보안 그룹을 구성합니다.
   - 소스의 모든 TCP를 10.200.0.0/16(으)로 엽니다.
   - 소스의 모든 ICMP - IPV4를 10.200.0.0/16(으)로 엽니다.
6. 인스턴스 세부 정보를 검토하고 EC2 인스턴스를 시작하려면 검토 및 시작을 선택합니다.

자세한 내용은 Amazon EC2 Linux 인스턴스 시작하기를 참조하세요.

Amazon VPC 인터페이스 엔드포인트 생성

VPC에 대한 VPC 엔드포인트를 생성한 다음 EC2 API에서 액세스할 수 있습니다. 엔드포인트를 생성하려면:

1. VPC 엔드포인트 콘솔로 이동하고 엔드포인트 생성을 선택합니다.
2. 엔드포인트 생성 페이지에서 다음 정보를 지정합니다.
   - 서비스 범주에서 AWS 서비스를 선택합니다.
   - 서비스 이름에 키워드 **iotwireless**를 입력하여 검색합니다. 표시된 iotwireless 서비스 목록에서 해당 리전의 제어 영역 API 엔드포인트를 선택합니다. 엔드포인트 형식은 com.amazonaws.*region*.iotwireless.api이(가) 될 것입니다.
   - VPC 및 서브넷에서 엔드포인트를 생성하려는 VPC 및 엔드포인트 네트워크를 생성하려는 가용 영역(AZ)을 선택합니다.

> **Note**
>
> iotwireless 서비스가 모든 가용 영역을 지원할 수 있는 것은 아닙니다.

- DNS 이름 활성화(Enable DNS name)에서 이 엔드포인트에 대해 활성화(Enable for this endpoint)를 선택합니다.

  이 옵션을 선택하면 DNS가 자동으로 확인되고 Amazon Route 53 Public Data Plane의 경로가 생성되므로 나중에 연결을 테스트하는 데 사용하는 API가 privatelink 엔드포인트를 통과합니다.
- 보안 그룹에서 엔드포인트 네트워크 인터페이스와 연결하려는 보안 그룹을 선택합니다.
- 선택적으로 태그를 추가하거나 제거할 수 있습니다. 태그는 엔드포인트와 연결하는 데 사용하는 이름-값 페어입니다.

3. VPC 엔드포인트를 생성하려면 엔드포인트 생성을 선택합니다.

엔드포인트 인터페이스와의 연결 테스트

SSH를 사용하여 Amazon EC2 인스턴스에 액세스한 다음 AWS CLI를 사용하여 프라이빗 링크 인터페이스 엔드포인트에 연결할 수 있습니다.

인터페이스 엔드포인트에 연결하기 전에 Linux에서 AWS CLI 버전 2 설치, 업그레이드 및 설치 제거에 설명된 지침에 따라 최신 AWS CLI 버전을 다운로드합니다.

다음 예제에서는 CLI를 사용하여 인터페이스 엔드포인트에 대한 연결을 테스트하는 방법을 보여줍니다.

```
aws iotwireless create-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com  \
    --name='test-privatelink'
```

다음 예제에서는 명령 실행 예제를 보여줍니다.

```
Response:
{
 "Arn": "arn:aws:iotwireless:region:acct_number:ServiceProfile/1a2345ba-4c5d-67b0-ab67-e0c8342f2857",
 "Id": "1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
}
```

마찬가지로 다음 명령을 실행하여 서비스 프로파일 정보를 가져오거나 모든 서비스 프로파일을 나열할 수 있습니다.

```
aws iotwireless get-service-profile \
```

```
    --endpoint-url https://api.iotwireless.region.amazonaws.com
    --id="1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
```

다음은 list-device-profiles 명령의 예입니다.

```
aws iotwireless list-device-profiles \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
```

AWS IoT Core for LoRaWAN 데이터 영역 API 엔드포인트 온보딩

AWS IoT Core for LoRaWAN 데이터 영역 엔드포인트는 다음 엔드포인트로 구성됩니다. 게이트웨이를 AWS IoT Core for LoRaWAN에 추가할 때 이러한 엔드포인트를 얻습니다. 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 추가 단원을 참조하십시오.

- LoRaWAN 네트워크 서버(LNS) 엔드포인트

  LNS 엔드포인트의 형식은 *account-specific-prefix*.lns.lorawan.*region*.amazonaws.com입니다. 이 엔드포인트를 사용하여 LoRA 업링크 및 다운링크 메시지를 교환하기 위한 연결을 설정할 수 있습니다.

- Configuration and Update Server(CUPS) 엔드포인트

  CUPS 엔드포인트의 형식은 *account-specific-prefix*.cups.lorawan.*region*.amazonaws.com입니다. 게이트웨이의 자격 증명 관리, 원격 구성 및 펌웨어 업데이트에 이 엔드포인트를 사용할 수 있습니다.

자세한 내용은 CUPS 및 LNS 프로토콜 사용 단원을 참조하십시오.

AWS 계정 및 리전에 대한 데이터 영역 API 엔드포인트를 찾으려면 여기에 표시된 get-service-endpoint CLI 명령 또는 GetServiceEndpoint REST API를 사용합니다. 자세한 내용은 AWS IoT Core for LoRaWAN 데이터 영역 API 엔드포인트를 참조하세요.

온프레미스에서 LoRaWAN 게이트웨이를 연결하여 AWS IoT Core for LoRaWAN 엔드포인트와 통신할 수 있습니다. 이 연결을 설정하려면 먼저 VPN 연결을 사용하여 VPC의 AWS 계정에 온프레미스 게이트웨이를 연결합니다. 그런 다음 프라이빗 링크로 구동되는 AWS IoT Core for LoRaWAN VPC의 데이터 영역 인터페이스 엔드포인트와 통신할 수 있습니다.

다음은 이러한 엔드포인트를 온보딩하는 방법을 보여 줍니다.

- VPC 인터페이스 엔드포인트 및 프라이빗 호스팅 영역 생성

- VPN을 사용하여 LoRa 게이트웨이를 AWS 계정에 연결

VPC 인터페이스 엔드포인트 및 프라이빗 호스팅 영역 생성

AWS IoT Core for LoRaWAN에는 두 개의 데이터 영역 엔드포인트인 구성 및 업데이트 서버(CUPS) 엔드포인트와 LoRaWAN 네트워크 서버(LNS) 엔드포인트가 있습니다. 두 엔드포인트에 대한 프라이빗 링크 연결을 설정하는 설치 프로세스는 동일하므로 설명을 위해 LNS 엔드포인트를 사용할 수 있습니다.

데이터 영역 엔드포인트의 경우 LoRA 게이트웨이는 먼저 Amazon VPC의 AWS 계정에 연결합니다. 그런 다음 AWS IoT Core for LoRaWAN VPC의 VPC 엔드포인트에 연결합니다.

엔드포인트에 연결할 때 DNS 이름은 하나의 VPC 내에서 확인할 수 있지만 여러 VPC에서 확인할 수는 없습니다. 엔드포인트를 생성할 때 프라이빗 DNS를 비활성화하려면 DNS 이름 활성화 설정을 비활성화합니다. 프라이빗 호스팅 영역을 사용하여 Route 53가 VPC의 DNS 쿼리에 응답하는 방법에 대한 정보를 제공할 수 있습니다. VPC를 온프레미스 환경과 공유하려면 Route 53 Resolver를 사용하여 하이브리드 DNS를 용이하게 할 수 있습니다.

이 자습서의 절차를 완료하려면 다음 단계를 수행하세요.

- Amazon VPC 및 서브넷 생성
- Amazon VPC 엔드포인트 생성
- 프라이빗 호스팅 영역 구성
- Route 53 인바운드 해석기 구성
- 다음 단계

Amazon VPC 및 서브넷 생성

제어 영역 엔드포인트를 온보딩할 때 생성한 Amazon VPC와 서브넷을 재사용할 수 있습니다. 자세한 내용은 Amazon VPC 및 서브넷 생성 단원을 참조하십시오.

Amazon VPC 엔드포인트 생성

VPC에 대한 VPC 엔드포인트를 생성할 수 있습니다. 이는 제어 영역 엔드포인트에 대한 VPC 엔드포인트를 생성하는 방법과 유사합니다.

1. VPC 엔드포인트 콘솔로 이동하고 엔드포인트 생성을 선택합니다.
2. 엔드포인트 생성 페이지에서 다음 정보를 지정합니다.

- 서비스 범주에서 AWS 서비스를 선택합니다.
- 서비스 이름에 키워드 **lns**를 입력하여 검색합니다. 표시된 lns 서비스 목록에서 해당 리전의 LNS 데이터 영역 API 엔드포인트를 선택합니다. 엔드포인트 형식은 `com.amazonaws.region.lorawan.lns`이(가) 될 것입니다.

> **Note**
>
> CUPS 엔드포인트에 대해 이 절차를 따르는 경우 cups을(를) 검색합니다. 엔드포인트 형식은 `com.amazonaws.region.lorawan.cups`이(가) 될 것입니다.

- VPC 및 서브넷에서 엔드포인트를 생성하려는 VPC 및 엔드포인트 네트워크를 생성하려는 가용 영역(AZ)을 선택합니다.

> **Note**
>
> iotwireless 서비스가 모든 가용 영역을 지원할 수 있는 것은 아닙니다.

- DNS 이름 활성화에서 이 엔드포인트에 대해 활성화를 선택하지 않도록 합니다.

  이 옵션을 선택하지 않으면 VPC 엔드포인트에 대해 프라이빗 DNS를 비활성화하고 대신 프라이빗 호스팅 영역을 사용할 수 있습니다.
- 보안 그룹에서 엔드포인트 네트워크 인터페이스와 연결하려는 보안 그룹을 선택합니다.
- 선택적으로 태그를 추가하거나 제거할 수 있습니다. 태그는 엔드포인트와 연결하는 데 사용하는 이름-값 페어입니다.

3. VPC 엔드포인트를 생성하려면 엔드포인트 생성을 선택합니다.

프라이빗 호스팅 영역 구성

프라이빗 링크 엔드포인트를 만든 후 엔드포인트의 세부 정보 탭에서 DNS 이름 목록이 표시됩니다. 이러한 DNS 이름 중 하나를 사용하여 프라이빗 호스팅 영역을 구성할 수 있습니다. DNS 이름의 형식은 `vpce-xxxx.lns.lorawan.region.vpce.amazonaws.com`이 됩니다.

프라이빗 호스팅 영역 생성

프라이빗 호스팅 영역을 생성하려면

1. Route 53 호스팅 영역 콘솔로 이동하여 호스팅 영역 생성을 선택합니다.
2. 호스팅 영역 생성 페이지에서 다음 정보를 지정합니다.

- 도메인 이름에 LNS 엔드포인트의 전체 서비스 이름(**lns.lorawan.region.amazonaws.com**)을 입력합니다.

  > **Note**
  >
  > CUPS 엔드포인트에 대해 이 절차를 따르는 경우 **cups.lorawan.region.amazonaws.com**을(를) 입력합니다.

- 유형 목록에서 프라이빗 호스팅 영역을 선택합니다.
- 필요에 따라 호스팅 영역과 연결할 태그를 추가하거나 제거할 수 있습니다.

3. 프라이빗 호스팅 영역을 생성하려면 호스팅 영역 생성을 선택합니다.

자세한 내용은 프라이빗 호스팅 영역 생성을 참조하세요.

프라이빗 호스팅 영역을 생성한 후 DNS에 트래픽을 해당 도메인으로 라우팅할 방법을 알려주는 레코드를 생성할 수 있습니다.

레코드 생성

프라이빗 호스팅 영역을 생성한 후 DNS에 트래픽을 해당 도메인으로 라우팅할 방법을 알려주는 레코드를 생성할 수 있습니다. 레코드를 생성하려면:

1. 표시된 호스팅 영역 목록에서 이전에 생성한 프라이빗 호스팅 영역을 선택하고 레코드 생성을 선택합니다.
2. 마법사 메서드를 사용하여 레코드를 만듭니다. 콘솔이 빠른 생성 메서드를 제공하는 경우 마법사로 전환을 선택합니다.
3. 라우팅 정책에서 단순 라우팅을 선택한 후 다음을 선택합니다.
4. 레코드 구성 페이지에서 단순 레코드 정의를 선택합니다.
5. 단순 레코드 정의 페이지에서:
   - 레코드 이름에 AWS 계정 번호의 별칭을 입력합니다. 게이트웨이를 온보딩할 때 또는 `GetServiceEndpoint` REST API를 사용하여 이 값을 얻습니다.
   - 레코드 유형에서 값을 `A - Routes traffic to an IPv4 address and some AWS resources`로 유지합니다.
   - 값/트래픽 라우팅 대상(Value/Route traffic to)에서 VPC 엔드포인트에 대한 별칭(Alias to VPC endpoint)을 선택합니다. 그리고 나서 리전을 선택한 다음 Amazon VPC 엔드포인트 생성에 설명된 대로 표시된 엔드포인트 목록에서 이전에 생성한 엔드포인트를 선택합니다.

6. 단순 레코드 정의를 선택하여 레코드를 만듭니다.

Route 53 인바운드 해석기 구성

VPC 엔드포인트를 온프레미스 환경과 공유하려면 Route 53 Resolver를 사용하여 하이브리드 DNS를 용이하게 할 수 있습니다. 인바운드 해석기를 사용하면 공용 인터넷을 거치지 않고도 온프레미스 네트워크에서 데이터 영역 엔드포인트로 트래픽을 라우팅할 수 있습니다. 서비스의 프라이빗 IP 주소 값을 반환하려면 VPC 엔드포인트와 동일한 VPC에서 Route 53 Resolver를 생성합니다.

인바운드 해석기를 생성할 때 VPC 및 이전에 가용 영역(AZ)에서 생성한 서브넷만 지정하면 됩니다. Route 53 Resolver는 이 정보를 사용하여 트래픽을 각 서브넷으로 라우팅하는 IP 주소를 자동으로 할당합니다.

인바운드 해석기를 생성하려면 다음과 같이 하세요.

1. Route 53 인바운드 엔드포인트 콘솔로 이동하고 인바운드 엔드포인트 생성을 선택합니다.

> **Note**
>
> 엔드포인트 및 프라이빗 호스팅 영역을 생성할 때 사용한 것과 동일한 AWS 리전을 사용 중이어야 합니다.

2. 인바운드 엔드포인트 생성 페이지에서 다음 정보를 지정합니다.
   - 엔드포인트 이름에 이름(예: **`VPC_A_Test`**)을 입력합니다.
   - 리전의 VPC에서, VPC 엔드포인트를 생성할 때 사용한 것과 동일한 VPC를 선택합니다.
   - 이 엔드포인트에 대한 보안 그룹(Security group for this endpoint)을 구성하여 온프레미스 네트워크에서 들어오는 트래픽을 허용할 수 있습니다.
   - IP 주소에서, 자동으로 선택된 IP 주소 사용(Use an IP address that is selected automatically)을 선택합니다.
3. 제출을 클릭하여 인바운드 해석기를 만듭니다.

이 샘플의 경우 IP 주소 `10.100.0.145` 및 `10.100.192.10`이 트래픽 라우팅을 위한 인바운드 Route 53 Resolver에 할당되었습니다.

다음 단계

프라이빗 호스팅 영역과 인바운드 해석기를 생성하여 DNS 항목의 트래픽을 라우팅했습니다. 이제 Site-to-Site VPN 또는 Client VPN 엔드포인트를 사용할 수 있습니다. 자세한 내용은 VPN을 사용하여 LoRa 게이트웨이를 AWS 계정에 연결 단원을 참조하십시오.

VPN을 사용하여 LoRa 게이트웨이를 AWS 계정에 연결

온프레미스의 게이트웨이를 AWS 계정에 연결하려면 Site-to-Site VPN 연결이나 Client VPN 엔드포인트를 사용할 수 있습니다.

온프레미스 게이트웨이를 연결하려면 먼저 VPC 엔드포인트를 생성하고, 게이트웨이의 트래픽이 공용 인터넷을 거치지 않도록 프라이빗 호스팅 영역 및 인바운드 해석기를 구성해야 합니다. 자세한 내용은 VPC 인터페이스 엔드포인트 및 프라이빗 호스팅 영역 생성 단원을 참조하십시오.

Site-to-Site VPN 엔드포인트

게이트웨이 하드웨어가 없거나 다른 AWS 계정을 사용하여 VPN 연결을 테스트하려면 Site-to-Site VPN 연결을 사용할 수 있습니다. Site-to-Site VPN을 사용하여 동일한 AWS 계정, 또는 다른 AWS 리전에서 사용할 수 있는 다른 AWS 계정의 VPC 엔드포인트에 연결할 수 있습니다.

> **Note**
>
> 게이트웨이 하드웨어를 사용하고 있고 VPN 연결을 설정하려는 경우 Client VPN을 대신 사용하는 것이 좋습니다. 지침은 Client VPN 엔드포인트을(를) 참조하십시오.

Site-to-Site VPN을 설정하려면:

1. 연결을 설정하려는 사이트에서 다른 VPC를 생성합니다. VPC-A에 대해 이전에 생성한 VPC를 재사용할 수 있습니다. 다른 VPC(예: VPC-B)를 생성하려면 이전에 생성한 VPC의 CIDR 블록과 겹치지 않는 CIDR 블록을 사용합니다.

   VPC 설정에 대한 자세한 내용은 AWS Site-to-Site VPN 연결 설정에 설명된 지침을 따르세요.

   > **Note**
   >
   > 이 문서에 설명된 Site-to-Site VPN 메서드는 VPN 연결에 OpenSWAN을 사용합니다. 이는 하나의 VPN 터널만 지원합니다. VPN에 다른 상용 소프트웨어를 사용하는 경우 사이트 사이에 두 개의 터널을 설정할 수 있습니다.

2. VPN 연결을 설정한 후 AWS 계정에서 인바운드 해석기의 IP 주소를 추가하여 `/etc/resolv.conf` 파일을 업데이트합니다. 이 IP 주소는 nameserver에 사용합니다. 이 IP 주소를 얻는 방법에 대한 자세한 내용은 Route 53 인바운드 해석기 구성 단원을 참조하세요. 이 예에서는 Route 53 Resolver를 만들 때 할당한 IP 주소 `10.100.0.145`를 사용할 수 있습니다.

```
options timeout:2 attempts:5
; generated by /usr/sbin/dhclient-script
search region.compute.internal
nameserver 10.100.0.145
```

3. 이제 `nslookup` 명령을 사용하여 VPN 연결이 공용 인터넷을 통하지 않고 AWS PrivateLink 엔드포인트를 사용하는지 테스트할 수 있습니다. 다음 예제에서는 명령 실행 예제를 보여줍니다.

```
nslookup account-specific-prefix.lns.lorawan.region.amazonaws.com
```

   다음 예제에서는 AWS PrivateLink LNS 엔드포인트에 대한 연결이 설정된 프라이빗 IP 주소를 표시하는 명령 실행의 출력을 보여 줍니다.

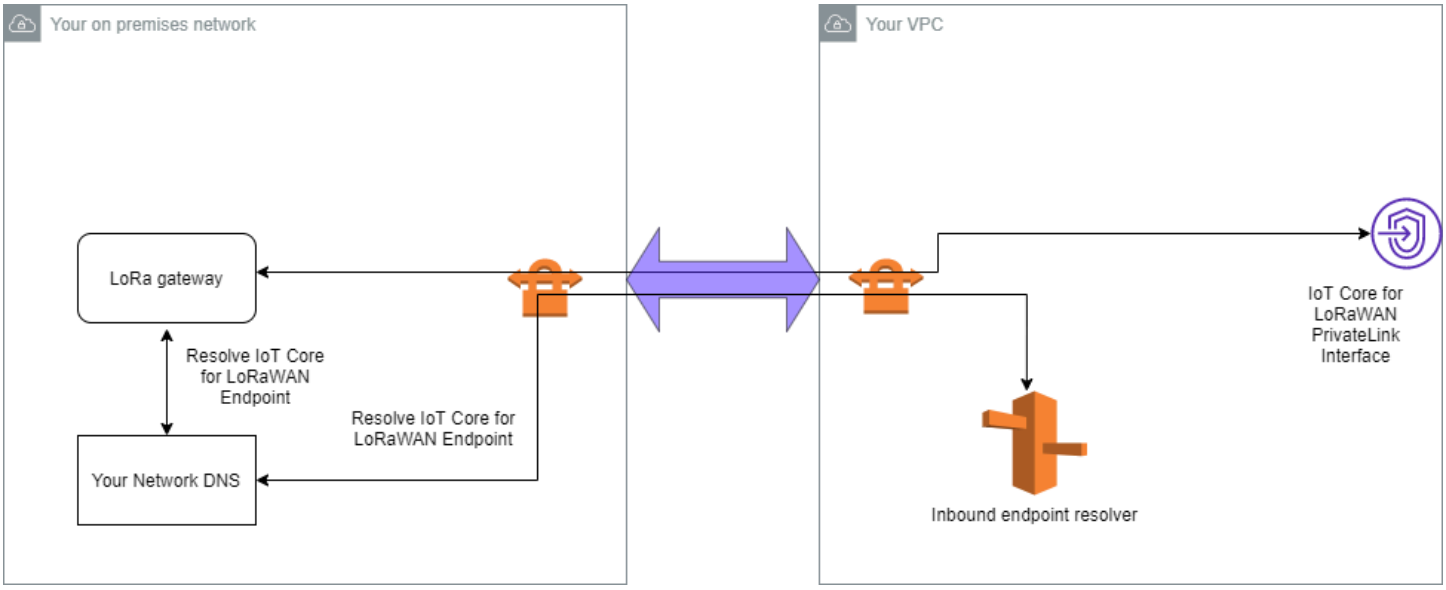
```
Server: 10.100.0.145
Address: 10.100.0.145

Non-authoritative answer:
Name: https://xxxxx.lns.lorawan.region.amazonaws.com
Address: 10.100.0.204
```

Site-to-Site VPN 연결 사용에 대한 자세한 내용은 Site-to-Site VPN 작동 방식을 참조하세요.

Client VPN 엔드포인트

AWS Client VPN은 AWS 리소스와 온프레미스 네트워크 리소스를 안전하게 액세스할 수 있게 해주는 관리형 클라이언트 기반 VPN 서비스입니다. 다음은 클라이언트 VPN 서비스의 아키텍처를 보여 줍니다.



Client VPN 엔드포인트에 대한 VPN 연결을 설정하려면:

1. AWS Client VPN 시작하기에 설명된 지침에 따라 Client VPN 엔드포인트를 생성합니다.
2. 해당 라우터의 액세스 URL(예: `192.168.1.1`)을 사용하여 온프레미스 네트워크(예: Wi-Fi 라우터)에 로그인하고 루트 이름과 암호를 찾습니다.
3. 게이트웨이 설명서의 지침을 따라 LoRaWAN 게이트웨이를 설정한 다음 게이트웨이를 AWS IoT Core for LoRaWAN에 추가합니다. 게이트웨이를 추가하는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 온보딩 단원을 참조하세요.
4. 게이트웨이의 펌웨어가 최신 상태인지 확인합니다. 펌웨어가 오래된 경우 온프레미스 네트워크에 제공된 지침에 따라 게이트웨이의 펌웨어를 업데이트할 수 있습니다. 자세한 내용은 AWS IoT Core for LoRaWAN에서 CUPS 서비스를 사용하여 게이트웨이 펌웨어 업데이트 단원을 참조하십시오.
5. OpenVPN이 활성화되었는지 확인합니다. 활성화된 경우 다음 단계로 건너뛰고 온프레미스 네트워크 내에서 OpenVPN 클라이언트를 구성합니다. 활성화되지 않은 경우 OpenVPN for OpenWrt 설치 가이드의 지침을 따릅니다.

   > **Note**
   >
   > 이 예제에서는 OpenVPN을 사용합니다. 다른 VPN 클라이언트(예: AWS VPN 또는 AWS Direct Connect)를 사용하여 Client VPN 연결을 설정할 수 있습니다.

6. 클라이언트 구성의 정보와, LuCi를 사용하는 OpenVPN 클라이언트의 사용 방법에 따라 OpenVPN 클라이언트를 구성합니다.

7. 온프레미스 네트워크에 SSH 연결하고, AWS 계정에서 인바운드 해석기의 IP 주소(`10.100.0.145`)를 추가하여 `/etc/resolv.conf` 파일을 업데이트합니다.
8. AWS PrivateLink를 사용하여 엔드포인트에 연결하는 게이트웨이 트래픽의 경우, 게이트웨이의 첫 번째 DNS 항목을 인바운드 해석기의 IP 주소로 바꿉니다.

Site-to-Site VPN 연결 사용에 대한 자세한 내용은 Client VPN 시작하기를 참조하세요.

LNS 및 CUPS VPC 엔드포인트에 연결

다음은 LNS 및 CUPS VPC 엔드포인트에 대한 연결을 테스트하는 방법을 보여줍니다.

CUPS 엔드포인트 테스트

LoRa 게이트웨이에서 CUPS 엔드포인트로의 AWS PrivateLink 연결을 테스트하려면 다음 명령을 실행합니다.

```
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 엔드포인트 테스트

LNS 엔드포인트를 테스트하려면 먼저 무선 게이트웨이와 함께 작동하는 LoRaWAN 디바이스를 프로비저닝합니다. 그런 다음 디바이스를 추가하고 조인 프로시저를 수행한 후에 업링크 메시지 전송을 시작할 수 있습니다.

Amazon Sidewalk용 AWS IoT Core

Amazon Sidewalk용 AWS IoT Core는 Sidewalk 엔드 디바이스를 AWS 클라우드에 연결하고 다른 AWS 서비스를 사용하는 데 사용할 수 있는 클라우드 서비스를 제공합니다.

Amazon Sidewalk는 커뮤니티의 디바이스를 연결하고 연결 상태를 유지할 수 있게 해주는 안전한 공유 네트워크입니다. Amazon Sidewalk는 Sidewalk 엔드 디바이스와 Sidewalk 게이트웨이 간에, 그리고 Sidewalk 게이트웨이와 Sidewalk 클라우드 간에 데이터를 전송합니다.

Amazon Sidewalk용 AWS IoT Core에 액세스

콘솔 또는 AWS IoT 무선 API 작업을 사용하여 Sidewalk 엔드 디바이스를 AWS IoT에 온보딩할 수 있습니다. 디바이스가 온보딩되면 메시지가 AWS IoT Core로 전송됩니다. 그러면 Amazon Sidewalk 엔드 디바이스의 데이터를 사용하는 AWS 클라우드에서 비즈니스 애플리케이션 개발을 시작할 수 있습니다.

콘솔 사용

Sidewalk 엔드 디바이스를 온보딩하려면 AWS Management Console에 로그인하고 AWS IoT 콘솔의 디바이스 페이지로 이동하세요. 디바이스가 온보딩되면 IoT 콘솔의 이 페이지에서 디바이스를 보고 관리할 수 있습니다.

API 또는 CLI 사용

AWS IoT 무선 API 작업을 사용하여 Sidewalk 및 LoRaWAN 디바이스를 모두 온보딩할 수 있습니다. AWS IoT Core가 내장되어 있는 AWS IoT 무선 API는 AWS SDK에서 지원됩니다. 자세한 내용은 AWS SDK 및 도구 단원을 참조하세요.

AWS CLI를 사용하여 Sidewalk 엔드 디바이스를 온보딩하고 관리하기 위한 명령을 실행할 수 있습니다. 자세한 내용은 AWS IoT 무선 CLI 참조를 참조하세요.

Amazon Sidewalk용 AWS IoT Core 리전 및 엔드포인트

Amazon Sidewalk는 `us-east-1` AWS 리전에서만 사용할 수 있습니다. Amazon Sidewalk용 AWS IoT Core는 이 리전의 컨트롤 플레인 및 데이터 영역 API 엔드포인트에 대한 지원을 제공합니다. 데이터 영역 API 엔드포인트는 AWS 계정에 한정됩니다. 자세한 정보는 AWS 일반 참조의 AWS IoT 무선 서비스 엔드포인트를 참조하세요.

Amazon Sidewalk용 AWS IoT Core에는 디바이스와 AWS 클라우드 간에 전송되는 디바이스 데이터에 적용되는 할당량과 AWS IoT 무선 API 작업을 위한 최대 TPS가 있습니다. 자세한 정보는 AWS 일반 참조의 AWS IoT 무선 할당량을 참조하세요.

Amazon Sidewalk용 AWS IoT Core 요금

AWS 가입 시 AWS 프리 티어를 사용하여 무료로 Amazon Sidewalk용 AWS IoT Core를 시작할 수 있습니다.

일반 제품 개요 및 요금에 대한 자세한 내용은 AWS IoT Core 요금을 참조하세요.

Amazon Sidewalk용 AWS IoT Core란?

Amazon Sidewalk용 AWS IoT Core를 사용하면 Amazon Sidewalk 엔드 디바이스를 AWS IoT에 온보딩하여 관리 및 모니터링할 수 있습니다. 또한 디바이스 데이터를 다른 AWS 서비스로 전송하는 대상도 관리합니다.

Amazon Sidewalk용 AWS IoT Core의 기능

Amazon Sidewalk용 AWS IoT Core를 사용하면 다음과 같은 작업을 수행할 수 있습니다.

- AWS IoT 콘솔, Amazon Sidewalk용 AWS IoT Core API 작업 또는 AWS CLI 명령을 사용하여 Sidewalk 엔드 디바이스를 AWS IoT에 온보딩합니다.
- AWS 클라우드에서 제공하는 기능을 활용합니다.
- AWS IoT 규칙을 사용하여 수신되는 페이로드 메시지를 처리하고 다른 AWS 서비스와 상호 작용하는 대상을 만듭니다.
- 이벤트 알림을 활성화하여 Sidewalk 엔드 디바이스가 프로비저닝 또는 등록될 때, 다운링크 메시지가 디바이스에 성공적으로 전달되었는지 여부와 같은 이벤트에 대한 메시지를 수신합니다.
- Sidewalk 엔드 디바이스를 실시간으로 로깅 및 모니터링하여 유용한 인사이트를 얻고 오류를 식별하여 문제를 해결합니다.
- Sidewalk 엔드 디바이스를 AWS IoT 사물과 연결하면 디바이스를 클라우드에 묘사하여 저장하는 데 도움이 됩니다. AWS IoT의 사물을 사용하면 기능을 더 쉽게 검색 및 관리하고 다른 AWS IoT Core 기능에 액세스할 수 있습니다.

다음 주제는 Amazon Sidewalk와 Amazon Sidewalk용 AWS IoT Core에 대해 배우는 데 도움이 될 것입니다.

주제

Amazon Sidewalk란?

Amazon Sidewalk는 호환되는 Amazon Echo 및 Ring 디바이스와 같은 Amazon Sidewalk Bridge를 사용하여 IoT 디바이스에 클라우드 연결을 제공하는 안전한 커뮤니티 네트워크입니다. Amazon Sidewalk는 근거리 통신에는 Bluetooth LE를 사용하고 장거리 통신에는 LoRa 및 900MHz 주파수의 FSK 무선 프로토콜을 사용하여 가정은 물론 밖에서도 저대역폭 및 장거리 연결을 지원합니다.

Amazon Sidewalk를 활성화하면 이 네트워크는 커뮤니티의 다른 Sidewalk 엔드 디바이스를 지원할 수 있으며, 환경 감지와 같은 애플리케이션에 사용할 수 있습니다. Amazon Sidewalk는 디바이스가 연결되고 연결 상태를 유지하도록 도와줍니다.

Amazon Sidewalk의 기능

Amazon Sidewalk의 기능은 다음과 같습니다.

- Amazon Sidewalk는 Ring 및 일부 Echo 디바이스가 포함된 Sidewalk 게이트웨이를 사용하여 저대역폭 네트워크를 생성합니다. 게이트웨이를 사용하면 인터넷 대역폭의 일부를 공유할 수 있으며, 이 대역폭은 엔드 디바이스를 네트워크에 연결하는 데 사용됩니다.
- Amazon Sidewalk는 여러 계층의 암호화 및 보안을 갖춘 안전한 네트워킹 메커니즘을 제공합니다.
- Amazon Sidewalk는 Sidewalk 참여를 활성화하거나 비활성화할 수 있는 간단한 메커니즘을 제공합니다.

Amazon Sidewalk 개념

다음은 Amazon Sidewalk의 몇 가지 주요 개념입니다.

Sidewalk 게이트웨이

> Sidewalk 게이트웨이 또는 Amazon Sidewalk Bridge는 Sidewalk 엔드 디바이스와 클라우드 간에 데이터를 라우팅합니다. 게이트웨이는 SubG-CSS(비동기식, LDR), subG-FSK(동기식, HDR) 또는 Sidewalk 통신을 위한 Bluetooth LE를 지원하는 Echo 디바이스 또는 Ring Floodlight Cam과 같은 Amazon 디바이스입니다. Sidewalk 게이트웨이는 Sidewalk 커뮤니티와 인터넷 대역폭의 일부를 공유하여 Sidewalk 지원 디바이스 그룹에 연결을 제공합니다.

Sidewalk 엔드 디바이스

Sidewalk 엔드 디바이스는 Sidewalk 게이트웨이에 연결하여 Amazon Sidewalk에서 로밍합니다. 엔드 디바이스는 Sidewalk 지원 조명 또는 도어록과 같은 저대역폭, 저전력 스마트 제품입니다.

> **Note**
>
> 특정 Sidewalk 게이트웨이는 엔드 디바이스 역할을 할 수도 있습니다.

Sidewalk 네트워크 서버

Amazon에서 운영하는 Sidewalk 네트워크 서버는 수신 패킷을 확인하고 업링크 및 다운링크 메시지를 원하는 대상으로 라우팅하는 동시에 Sidewalk 네트워크의 시간을 동기화된 상태로 유지합니다.

Amazon Sidewalk 자세히 알아보기

Amazon Sidewalk에 대한 자세한 내용은 다음 웹 페이지를 참조하세요.

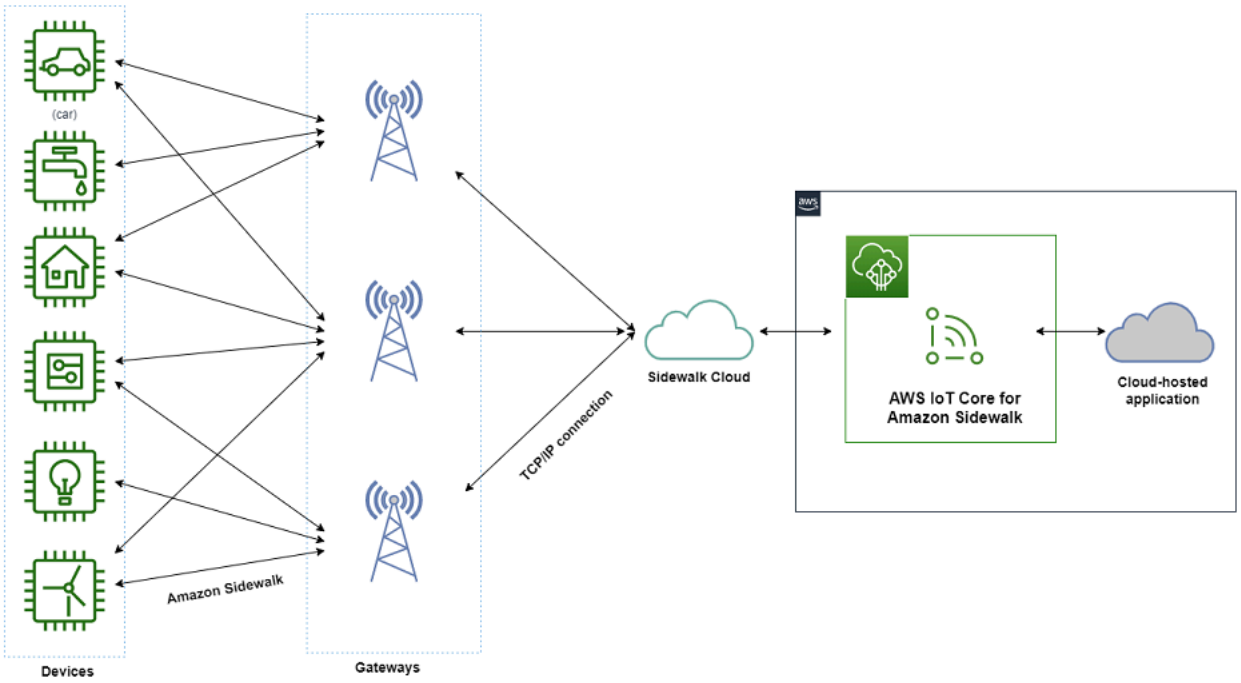
- Amazon Sidewalk
- Amazon Sidewalk 설명서
- Amazon Sidewalk용 AWS IoT Core

Amazon Sidewalk용 AWS IoT Core의 작동 방식

Amazon Sidewalk용 AWS IoT Core를 사용하면 Amazon Sidewalk 엔드 디바이스를 AWS IoT에 온보딩하여 관리 및 모니터링할 수 있습니다. 또한 디바이스 데이터를 다른 AWS 서비스로 전송하는 대상도 관리합니다.

Amazon Sidewalk용 AWS IoT Core는 Sidewalk 엔드 디바이스를 AWS 클라우드에 연결하고 다른 AWS 서비스를 사용하는 데 사용할 수 있는 클라우드 서비스를 제공합니다. 또한 Amazon Sidewalk용 AWS IoT Core를 사용하여 Sidewalk 디바이스를 관리하고 해당 디바이스에서 애플리케이션을 모니터링 및 구축할 수 있습니다.

Sidewalk 엔드 디바이스는 Sidewalk 게이트웨이를 통해 AWS IoT Core와 통신합니다. Amazon Sidewalk용 AWS IoT Core는 Sidewalk 엔드 디바이스 및 게이트웨이를 관리하고 이와 통신하는 데 AWS IoT Core에 필요한 서비스 및 디바이스 정책을 관리합니다. 또한 디바이스 데이터를 다른 AWS 서비스로 전송하는 대상도 관리합니다.



Amazon Sidewalk용 AWS IoT Core 사용 시작하기

AWS IoT 콘솔, Amazon Sidewalk용 AWS IoT Core API 또는 AWS CLI를 사용하여 Sidewalk 엔드 디바이스를 생성 및 온보딩하고 이를 Sidewalk 네트워크에 연결할 수 있습니다. Amazon Sidewalk를 시작하고 엔드 디바이스를 AWS IoT에 온보딩하는 방법에 대한 자세한 내용은 다음 주제를 참조하세요.

- Amazon Sidewalk용 AWS IoT Core 시작하기

  이 주제에서는 Sidewalk 엔드 디바이스를 온보딩하기 위한 사전 요구 사항을 살펴보고, 센서 모니터링 애플리케이션을 사용하는 워크플로를 설명하고, AWS CLI 명령을 사용하여 디바이스를 온보딩하는 방법에 대한 개요를 제공합니다.

- Amazon Sidewalk용 AWS IoT Core에 연결

  이 섹션에서는 온보딩 워크플로 소개의 여러 단계를 설명하고 콘솔 및 API 작업을 사용한 엔드 디바이스 온보딩을 안내합니다. 또한 디바이스를 연결하고 디바이스와 Amazon Sidewalk용 AWS IoT Core 간에 교환되는 메시지를 확인합니다.

- Amazon Sidewalk용 AWS IoT Core로 디바이스 대량 프로비저닝

  이 단원에서는 Amazon Sidewalk용 AWS IoT Core를 사용하여 Sidewalk 엔드 디바이스를 대량 프로비저닝하는 방법에 대한 자세한 단계별 자습서를 제공합니다. 대량 프로비저닝 워크플로와 다수의 Sidewalk 디바이스를 온보딩하는 방법을 배우게 됩니다.

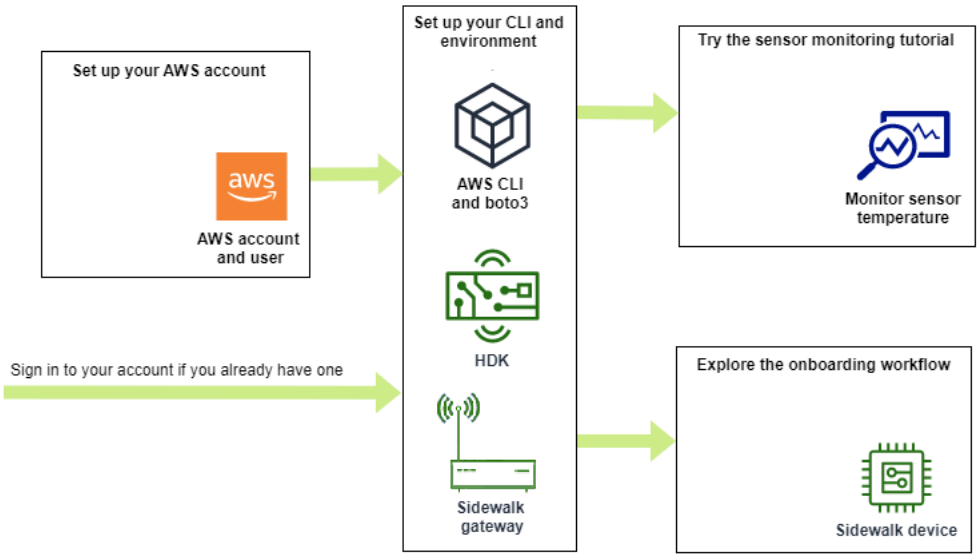
Amazon Sidewalk용 AWS IoT Core 자세히 알아보기

Amazon Sidewalk용 AWS IoT Core에 대한 자세한 내용은 다음 웹 페이지를 참조하세요.

- Amazon Sidewalk
- Amazon Sidewalk 설명서
- Amazon Sidewalk용 AWS IoT Core

Amazon Sidewalk용 AWS IoT Core 시작하기

이 단원에서는 Sidewalk 엔드 디바이스를 Amazon Sidewalk용 AWS IoT Core에 연결하기 시작하는 방법을 설명합니다. Amazon Sidewalk에 엔드 디바이스를 연결하고 서로 간에 메시지를 전달하는 방법을 설명합니다. 또한 Sidewalk 샘플 애플리케이션에 대해 알아보고 Amazon Sidewalk용 AWS IoT Core를 사용하여 센서 모니터링을 수행하는 방법에 대한 개요도 알아봅니다. 샘플 애플리케이션은 센서 온도의 변화를 보고 모니터링할 수 있는 대시보드를 제공합니다.



다음 주제는 Amazon Sidewalk용 AWS IoT Core를 시작하는 데 도움이 될 수 있습니다.

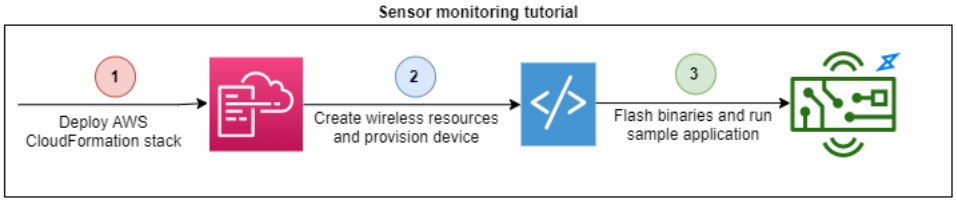
주제

- 센서 모니터링 자습서 사용해 보기
- Sidewalk 디바이스 온보딩 소개

센서 모니터링 자습서 사용해 보기

이 섹션에서는 센서 온도를 모니터링하는 방법을 보여주는 GitHub의 Amazon Sidewalk 샘플 애플리케이션에 대한 개요를 제공합니다. 이 자습서에서는 필요한 무선 리소스를 프로그래밍 방식으로 생성하고, 엔드 디바이스를 프로비저닝하고, 바이너리를 플래시한 다음, 엔드 디바이스를 애플리케이션에 연결하는 스크립트를 사용합니다. AWS CLI 및 Python 명령을 사용하는 스크립트는 AWS CloudFormation 스택과 무선 리소스를 만든 다음 바이너리를 플래시하고 애플리케이션을 Hardware Development Kit(HDK)에 배포합니다.

다음 다이어그램은 샘플 애플리케이션을 실행하고 Sidewalk 엔드 디바이스를 애플리케이션에 연결할 때 수반되는 단계를 보여줍니다. 이 자습서의 사전 요구 사항 및 구성을 포함한 자세한 지침은 GitHub의 README 문서를 참조하세요.



Sidewalk 디바이스 온보딩 소개

이 단원에서는 Sidewalk 엔드 디바이스를 Amazon Sidewalk용 AWS IoT Core에 온보딩하는 방법을 설명합니다. 디바이스를 온보딩하려면 먼저 Sidewalk 디바이스를 추가한 다음 디바이스를 프로비저닝 및 등록한 다음 하드웨어를 클라우드 애플리케이션에 연결합니다. 이 자습서를 실행하기 전에 먼저 Python 및 AWS CLI 설치 섹션을 검토하고 완료하세요.

다음 단계는 Sidewalk 엔드 디바이스를 Amazon Sidewalk용 AWS IoT Core에 온보딩하고 연결하는 방법을 보여 줍니다. AWS CLI를 사용하여 디바이스를 온보딩하려면 이 섹션에 제공된 샘플 명령을 참조하세요. AWS IoT 콘솔을 사용하여 디바이스를 온보딩하는 방법에 대한 자세한 내용은 Amazon Sidewalk용 AWS IoT Core에 연결 섹션을 참조하세요.

> **Important**
>
> 전체 온보딩 워크플로를 수행하려면 엔드 디바이스를 프로비저닝 및 등록하고 Hardware Development Kit(HDK)를 연결합니다. 자세한 내용은 Amazon Sidewalk 설명서의 엔드 디바이스 프로비저닝 및 등록을 참조하세요.

주제

- 1단계: Amazon Sidewalk용 AWS IoT Core에 Sidewalk 디바이스 추가

- 2단계: Sidewalk 엔드 디바이스의 대상 생성
- 3단계: 엔드 디바이스 프로비저닝 및 등록
- 4단계: Sidewalk 엔드 디바이스에 연결 및 메시지 교환

1단계: Amazon Sidewalk용 AWS IoT Core에 Sidewalk 디바이스 추가

다음은 Sidewalk 엔드 디바이스를 Amazon Sidewalk용 AWS IoT Core에 추가하기 위해 수행할 단계를 간략히 설명한 것입니다. 디바이스 프로필 및 생성한 무선 디바이스에 대해 수집한 정보를 저장합니다. 이 정보를 사용하여 엔드 디바이스를 프로비전하고 등록하게 됩니다. 이 단계에 대한 자세한 내용은 Amazon Sidewalk용 AWS IoT Core에 디바이스 추가 섹션을 참조하세요.

1. 디바이스 프로필 생성

   Sidewalk 디바이스의 공유 구성을 포함하는 디바이스 프로필을 생성합니다. 프로필을 생성할 때 프로필의 *name*을 영숫자 문자열로 지정하세요. 프로필을 생성하려면 AWS IoT 콘솔에서 프로필 허브의 Sidewalk 탭으로 이동하여 프로필 생성을 선택하거나, 이 예시와 같이 `CreateDeviceProfile` API 작업 또는 `create-device-profile` CLI 명령을 사용하세요.

   ```
   // Add your device profile using a name and the sidewalk object.
   aws iotwireless create-device-profile --name sidewalk_profile --sidewalk {}
   ```

2. Sidewalk 엔드 디바이스 생성

   Amazon Sidewalk용 AWS IoT Core로 엔드 디바이스를 생성합니다. 대상 이름과 이전 단계에서 얻은 디바이스 프로필의 ID를 지정합니다. 디바이스를 추가하려면 AWS IoT 콘솔에서 디바이스 허브의 Sidewalk 탭으로 이동하여 디바이스 프로비저닝을 선택하거나, 이 예시와 같이 `CreateWirelessDevice` API 작업 또는 `create-wireless-device` CLI 명령을 사용하세요.

   > **Note**
   >
   > AWS 계정 및 AWS 리전에서 고유한 대상 이름을 지정하세요. Amazon Sidewalk용 AWS IoT Core에 대상을 추가할 때 동일한 대상 이름을 사용하게 됩니다.

   ```
   // 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. 디바이스 프로필 및 무선 디바이스 정보 가져오기

   디바이스 프로필 및 무선 디바이스 정보를 JSON으로 가져옵니다. JSON에는 디바이스 세부 정보, 디바이스 인증서, 프라이빗 키, `DeviceTypeId` 및 Sidewalk 제조 일련번호(SMSN)에 대한 정보가 포함됩니다.

   - AWS IoT 콘솔을 사용하는 경우 디바이스 허브의 Sidewalk 탭을 사용하여 Sidewalk 엔드 디바이스용 통합 JSON 파일을 다운로드할 수 있습니다.
   - API 작업을 사용하는 경우 API 작업 `GetDeviceProfile` 및 `GetWirelessDevice`에서 얻은 응답을 별도의 JSON 파일(예: *`device_profile.json`* 및 *`wireless_device.json`*)로 저장하세요.

   ```
   // 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
   ```

2단계: Sidewalk 엔드 디바이스의 대상 생성

다음은 대상을 Amazon Sidewalk용 AWS IoT Core에 추가하기 위해 수행할 단계를 간략히 설명한 것입니다. AWS Management Console 또는 AWS IoT 무선 API 작업 또는 AWS CLI로 다음 단계를 실행하여 AWS IoT 규칙 및 대상을 생성합니다. 그런 다음 하드웨어 플랫폼에 연결하여 메시지를 보고 교환할 수 있습니다. 이 섹션의 AWS CLI 예시에 사용되는 샘플 IAM 역할 및 AWS IoT 규칙은 대상에 대한 IAM 역할 및 IoT 규칙 생성 섹션을 참조하세요.

1. IAM 역할 생성

   AWS IoT 규칙에 데이터를 전송할 권한을 Amazon Sidewalk용 AWS IoT Core에 부여하는 IAM 역할을 생성합니다. 역할을 생성하려면 `CreateRole` API 작업 또는 `create-role` CLI 명령을 사용합니다. 역할 이름을 *`SidewalkRole`*로 지정할 수 있습니다.

   ```
   aws iam create-role --role-name lambda-ex \
       --assume-role-policy-document file://lambda-trust-policy.json
   ```

2. 대상에 대한 규칙 생성

   디바이스 데이터를 처리하는 AWS IoT 규칙을 만들고 메시지를 게시할 주제를 지정합니다. 하드웨어 플랫폼에 연결한 후 이 주제에 대한 메시지를 관찰합니다.. AWS IoT Core API 작업 CreateTopicRule 또는 AWS CLI 명령 create-topic-rule을 사용하여 대상에 대한 규칙을 생성합니다.

   ```
   aws iot create-topic-rule --rule-name Sidewalkrule \
       --topic-rule-payload file://myrule.json
   ```

3. 대상 생성

   Sidewalk 디바이스를 다른 AWS 서비스와 함께 사용할 수 있도록 처리하는 IoT 규칙과 연결하는 대상을 생성합니다. AWS IoT 콘솔의 대상 허브, CreateDestination API 작업 또는 create-destination CLI 명령을 사용하여 대상을 추가할 수 있습니다.

   ```
   aws iotwireless create-destination --name SidewalkDestination \
       --expression-type RuleName --expression SidewalkRule \
       --role-arn arn:aws:iam::123456789012:role/SidewalkRole
   ```

3단계: 엔드 디바이스 프로비저닝 및 등록

Python 명령을 사용하여 엔드 디바이스를 프로비저닝하고 등록할 수 있습니다. 프로비저닝 스크립트는 획득한 디바이스 JSON 데이터를 사용하여 제조 바이너리 이미지를 생성합니다. 이 이미지는 하드웨어 보드에 플래시됩니다. 그런 다음 하드웨어 플랫폼에 연결할 엔드 디바이스를 등록합니다. 자세한 내용은 Amazon Sidewalk 설명서의 엔드 디바이스 프로비저닝 및 등록을 참조하세요.

> **Note**
>
> Sidewalk 엔드 디바이스를 등록할 때는 게이트웨이가 Amazon Sidewalk에 옵트인해야 하며 게이트웨이와 디바이스가 서로 범위 내에 있어야 합니다.

4단계: Sidewalk 엔드 디바이스에 연결 및 메시지 교환

엔드 디바이스를 등록한 후 엔드 디바이스를 연결하고 메시지 및 디바이스 데이터 교환을 시작할 수 있습니다.

1. Sidewalk 엔드 디바이스 연결

   HDK를 컴퓨터에 연결하고 공급업체 설명서에서 제공하는 지침에 따라 HDK에 연결합니다. 자세한 내용은 Amazon Sidewalk 설명서의 엔드 디바이스 프로비저닝 및 등록을 참조하세요.

2. 메시지 보기 및 교환

   MQTT 클라이언트를 사용하여 규칙에 지정된 주제를 구독하고 수신된 메시지를 봅니다. 또한 `SendDataToWirelessDevice` API 작업 또는 `send-data-to-wireless-device` CLI 명령을 사용하여 디바이스에 다운링크 메시지를 보내고 연결 상태를 확인할 수 있습니다.

   (선택 사항) 메시지 전송 상태 이벤트를 활성화하여 다운링크 메시지가 성공적으로 수신되었는지 확인할 수 있습니다.

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

Amazon Sidewalk용 AWS IoT Core에 연결

이 섹션에서는 Sidewalk 엔드 디바이스를 온보딩한 다음 디바이스를 Sidewalk 네트워크에 연결하는 방법을 설명합니다. Sidewalk 디바이스 온보딩 소개에 설명된 대로 온보딩 자습서에서 수행하는 단계를 설명합니다. AWS IoT 콘솔과 Amazon Sidewalk용 AWS IoT Core API 작업을 사용하여 디바이스를 온보딩하는 방법을 배우게 됩니다. 또한 이러한 작업을 수행하는 AWS CLI 명령에 대해서도 알아봅니다.

필수 조건

엔드 디바이스와 대상을 Amazon Sidewalk용 AWS IoT Core에 추가하려면 AWS 계정을 설정해야 합니다. AWS IoT 무선 API 또는 AWS CLI 명령을 사용하여 이러한 작업을 수행하려면 AWS CLI도 설정해야 합니다. 사전 조건 및 설정에 대한 자세한 내용은 Python 및 AWS CLI 설치 섹션을 참조하세요.

> **Note**
>
> 엔드 디바이스를 프로비저닝 및 등록하고 Hardware Development Kit(HDK)에 연결하는 전체 온보딩 워크플로를 수행하려면 Sidewalk 게이트웨이 및 HDK도 설정해야 합니다. 자세한 내용

은 Amazon Sidewalk 설명서의 Hardware Development Kit(HDK) 설정 및 Sidewalk 게이트웨이 설정을 참조하세요.

Sidewalk 리소스 설명

시작하고 리소스를 생성하기 전에 Sidewalk 엔드 디바이스, 디바이스 프로필 및 대상의 명명 규칙을 고려하세요. Amazon Sidewalk용 AWS IoT Core는 생성한 리소스에 고유 식별자를 할당합니다. 그러나 좀 더 설명이 가미된 이름을 지정하거나, 설명을 추가하거나, 식별 및 관리에 도움이 되는 선택적 태그를 추가할 수 있습니다.

> **Note**
>
> 대상 이름은 생성되고 나면 변경할 수 없습니다. AWS 계정 및 AWS 리전에 고유한 이름을 사용하세요.

자세한 내용은 AWS IoT 무선 리소스 설명 단원을 참조하십시오.

주제

- Amazon Sidewalk용 AWS IoT Core에 디바이스 추가
- Sidewalk 엔드 디바이스의 대상 추가
- Sidewalk 디바이스 연결 및 업링크 메타데이터 형식 보기

Amazon Sidewalk용 AWS IoT Core에 디바이스 추가

무선 디바이스를 만들기 전에 먼저 디바이스 프로필을 만드세요. 디바이스 프로필은 Sidewalk 디바이스의 디바이스 기능 및 기타 파라미터를 정의합니다. 단일 디바이스 프로필을 여러 디바이스에 연결할 수 있습니다.

디바이스 프로필을 만든 후 프로필에 대한 정보를 검색하면 `DeviceTypeId`가 반환됩니다. 엔드 디바이스를 프로비저닝할 때는 이 ID, 디바이스 인증서, 애플리케이션 서버 퍼블릭 키 및 SMSN을 사용하게 됩니다.

디바이스 생성 및 추가 방법

1. Sidewalk 엔드 디바이스의 디바이스 프로필을 생성합니다. Sidewalk 디바이스에 사용할 프로필 이름을 영숫자 문자열로 지정합니다. 프로필은 해당 프로필과 연결할 디바이스를 식별하는 데 도움이 됩니다.
   - (콘솔) Sidewalk 디바이스를 추가할 때 새 프로필을 만들 수도 있습니다. 이렇게 하면 디바이스를 Amazon Sidewalk용 AWS IoT Core에 빠르게 추가하고 프로필에 연결할 수 있습니다.
   - (API) 프로필 이름과 Sidewalk 객체 `sidewalk {}`를 지정하여 `CreateDeviceProfile` API 작업을 사용합니다. API 응답에는 프로필 ID와 Amazon 리소스 이름(ARN)이 포함됩니다.
2. Amazon Sidewalk용 AWS IoT Core에 무선 디바이스 추가 대상 이름을 지정하고 이전 단계에서 생성한 디바이스 프로필을 선택합니다.
   - (콘솔) Sidewalk 디바이스를 추가할 때 대상 이름을 입력하고 생성한 프로필을 선택합니다.
   - (API) `CreateWirelessDevice` API 작업을 사용합니다. 대상 이름과 이전에 얻은 디바이스 프로필의 ID를 지정합니다.

   무선 디바이스 파라미터

파라미터	설명	참고
대상 이름	다른 AWS 서비스가 사용할 디바이스 데이터를 처리하는 AWS IoT 규칙을 설명하는 대상의 이름.	대상을 생성하지 않은 경우 임의의 문자열 값을 제공할 수 있습니다. Amazon Sidewalk용 AWS IoT Core는 디바이스를 생성할 때 빈 대상을 생성하며, 대상을 추가할 때 이를 업데이트할 수 있습니다.
디바이스 프로필	이전에 생성한 디바이스 프로필.	–

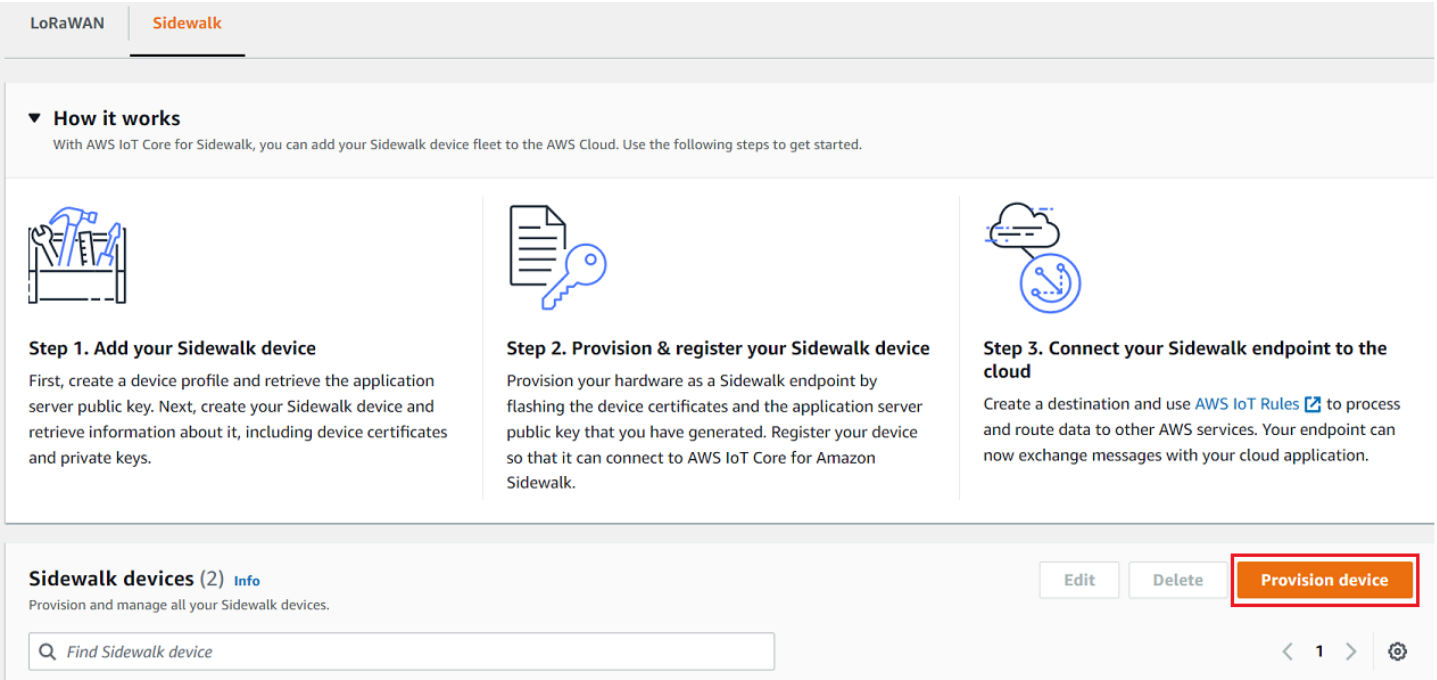
3. 엔드 디바이스를 프로비저닝하는 데 필요한 정보가 들어 있는 JSON 파일을 확보합니다.
   - (콘솔) 생성한 Sidewalk 디바이스의 세부 정보 페이지에서 이 파일을 다운로드합니다.
   - (API) `GetDeviceProfile` 및 `GetWirelessDevice` API 작업을 사용하여 디바이스 프로필 및 무선 디바이스에 대한 정보를 검색합니다. API 응답 정보를 JSON 파일(예: *`device_profile.json`* 및 *`wireless_device.json`*)로 저장합니다.

디바이스 프로필 및 Sidewalk 엔드 디바이스 추가

이 섹션에서는 디바이스 프로필을 생성하는 방법을 소개합니다. 또한 AWS IoT 콘솔 및 AWS CLI를 사용하여 Sidewalk 엔드 디바이스를 Amazon Sidewalk용 AWS IoT Core에 추가하는 방법도 보여 줍니다.

Sidewalk 디바이스 추가 (콘솔)

AWS IoT 콘솔을 사용하여 Sidewalk 디바이스를 추가하려면 디바이스 허브의 Sidewalk 탭으로 이동하여 디바이스 프로비저닝을 선택한 후 다음 단계를 수행하세요.



1. 디바이스 세부 정보 지정

   Sidewalk 디바이스의 구성 정보를 지정합니다. 새 디바이스 프로필을 만들거나 Sidewalk 디바이스의 기존 프로필을 선택할 수도 있습니다.

   a. 디바이스 이름 및 선택적 설명을 지정합니다. 설명은 최대 2,048자입니다. 이러한 필드는 디바이스를 만든 후에 편집할 수 있습니다.

   b. Sidewalk 디바이스와 연결할 디바이스 프로필을 선택합니다. 기존 디바이스 프로필이 있는 경우 프로필을 선택할 수 있습니다. 새 프로젝트를 생성하려면 새 프로필 생성을 선택하고 프로필의 이름을 입력합니다.

> **Note**
>
> 디바이스 프로필에 태그를 부착하려면 프로필을 만든 후 프로필 허브로 이동한 다음 프로필을 편집하여 이 정보를 추가합니다.

c. 디바이스의 메시지를 다른 AWS 서비스로 라우팅할 대상 이름을 지정합니다. 아직 대상을 만들지 않았다면 대상 허브로 이동하여 대상을 만듭니다. 그런 다음 Sidewalk 디바이스의 대상을 선택할 수 있습니다. 자세한 내용은 Sidewalk 엔드 디바이스의 대상 추가 단원을 참조하십시오.

d. Sidewalk 디바이스를 계속 추가하려면 다음을 선택합니다.

2. Sidewalk 디바이스를 AWS IoT 사물과 연결 (선택 사항)

   원하는 경우 Sidewalk 디바이스를 AWS IoT 사물에 연결할 수도 있습니다. IoT는 AWS IoT 디바이스 레지스트리의 항목입니다. 사물을 사용하면 디바이스를 더 쉽게 검색하고 관리할 수 있습니다. 사물을 디바이스에 연결하면 디바이스에서 다른 AWS IoT Core 기능에 액세스할 수 있습니다.

   디바이스를 사물과 연결하려면 자동 사물 등록을 선택합니다.

   a. Sidewalk 디바이스를 연결하려는 IoT 사물의 고유한 이름을 입력합니다. 사물 이름은 대소문자를 구분하며 AWS 계정 및 AWS 리전에서 고유해야 합니다.

   b. 사물 유형이나 사물 목록에서 필터링하는 데 사용할 수 있는 검색 가능한 속성을 사용하는 등 IoT 사물에 대한 추가 구성을 제공합니다.

   c. 다음을 선택하고 Sidewalk 디바이스에 대한 정보를 확인한 다음 생성을 선택합니다.

Sidewalk 디바이스 추가 (CLI)

Sidewalk 디바이스를 추가하고 Sidewalk 디바이스를 프로비저닝하는 데 사용할 JSON 파일을 다운로드하려면 다음 API 작업을 수행하세요.

주제

- 1단계: 디바이스 프로필 생성
- 2단계: Sidewalk 디바이스 추가

1단계: 디바이스 프로필 생성

AWS 계정에서 디바이스 프로필을 생성하려면 CreateDeviceProfile API 작업 또는 create-device-profile CLI 명령을 사용합니다. 디바이스 프로필을 생성할 때 이름을 지정하고 선택적으로 태그를 이름-값 페어로 제공하세요.

예를 들어 다음 명령을 실행하면 Sidewalk 디바이스의 디바이스 프로필이 생성됩니다.

```
aws iotwireless create-device-profile \
    --name sidewalk_profile --sidewalk {}
```

이 명령을 실행하면 디바이스 프로필의 Amazon 리소스 이름(ARN)과 ID가 출력으로 반환됩니다.

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

2단계: Sidewalk 디바이스 추가

Amazon Sidewalk용 AWS IoT Core 계정에 Sidewalk 디바이스를 추가하려면 CreateWirelessDevice API 작업 또는 create-wireless-device CLI 명령을 사용합니다. 디바이스를 생성할 때 다음 파라미터와 함께 선택적으로 Sidewalk 디바이스의 이름과 설명을 지정하세요.

> **Note**
>
> Sidewalk 디바이스를 AWS IoT 사물과 연결하려면 AssociateWirelessDeviceWithThing API 작업 또는 associate-wireless-device-with-thing CLI 명령을 사용합니다.

다음 명령은 Sidewalk 디바이스 생성의 예시를 보여줍니다.

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

다음은 device.json 파일의 콘텐츠를 보여줍니다.

device.json의 내용

```
{
  "Type": "Sidewalk",
  "Name": "SidewalkDevice",
  "DestinationName": "SidewalkDestination",
  "Sidewalk": {
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
    }
}
```

이 명령을 실행하면 디바이스 ID와 Amazon 리소스 이름(ARN)이 출력으로 반환됩니다.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/23456789-
abcd-0123-bcde-fabc012345678",
    "Id": "23456789-abcd-0123-bcde-fabc012345678"
}
```

프로비저닝에 사용할 디바이스 JSON 파일 확보

Sidewalk 디바이스를 Amazon Sidewalk용 AWS IoT Core에 추가한 후 엔드 디바이스를 프로비전하는 데 필요한 정보가 들어 있는 JSON 파일을 다운로드하세요. AWS IoT 콘솔이나 AWS CLI를 사용하여 이 정보를 검색할 수 있습니다. 디바이스를 프로비저닝하는 방법에 대한 자세한 내용은 Amazon Sidewalk 설명서의 엔드 디바이스 프로비저닝 및 등록을 참조하세요.

JSON 파일 확보 (콘솔)

Sidewalk 디바이스를 프로비저닝하기 위한 JSON 파일을 확보하는 방법:

1. Sidewalk 디바이스 허브로 이동합니다.
2. AWS IoT Core에 추가한 디바이스를 선택하여 세부 정보를 확인합니다.
3. 추가한 디바이스의 세부 정보 페이지에서 디바이스 JSON 파일 다운로드를 선택하여 JSON 파일을 확보합니다.

   엔드 디바이스를 프로비저닝하는 데 필요한 정보가 포함된 `certificate.json` 파일이 다운로드됩니다. 다음은 샘플 JSON 파일을 보여줍니다. 여기에는 디바이스 인증서, 프라이빗 키, Sidewalk 제조 일련번호(SMSN), `DeviceTypeID`가 포함됩니다.

   ```
   {
   ```

```
  "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 디바이스의 세부 정보 페이지에는 다음에 대한 정보도 표시됩니다.

- 디바이스 ID, Amazon 리소스 이름(ARN) 및 디바이스와 연결된 모든 AWS IoT 사물의 세부 정보.
- 디바이스 프로필 및 대상 세부 정보.
- 디바이스로부터 마지막 업링크 메시지를 수신한 시간.
- 디바이스가 프로비저닝되었는지 또는 등록되었는지를 나타내는 상태.

JSON 파일 확보 (CLI)

Amazon Sidewalk용 AWS IoT Core API 또는 AWS CLI를 사용하여 Sidewalk 엔드 디바이스를 프로비저닝하는 데 필요한 JSON 파일을 확보하려면 디바이스 프로필 및 무선 디바이스에 대한 정보를 검색한 결과 얻은 API 응답을 JSON 파일(예: *wireless_device.json* 및 *device_profile.json*)로 임시 저장합니다. 이를 사용하여 Sidewalk 디바이스를 프로비저닝합니다.

다음은 JSON 파일을 검색하는 방법을 보여줍니다.

주제

- 1단계: 디바이스 프로필 정보를 JSON 파일로 가져오기
- 2단계: Sidewalk 디바이스 정보를 JSON 파일로 가져오기

1단계: 디바이스 프로필 정보를 JSON 파일로 가져오기

GetDeviceProfile API 작업 또는 get-device-profile CLI 명령을 사용하여 Amazon Sidewalk 용 AWS IoT Core 계정에 추가한 디바이스 프로필에 대한 정보를 가져옵니다. 디바이스 프로필에 대한 정보를 검색하려면 프로필 ID를 지정합니다.

그러면 API가 지정된 식별자 및 디바이스 ID와 일치하는 디바이스 프로필에 대한 정보를 반환합니다. 이 응답 정보를 파일로 저장하고 *device_profile.json*과 같은 이름을 지정합니다.

다음은 CLI 명령의 예시입니다.

```
aws iotwireless get-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" > device_profile.json
```

이 명령을 실행하면 디바이스 프로필, 애플리케이션 서버 퍼블릭 키 및 DeviceTypeID의 파라미터가 반환됩니다. 다음은 API의 샘플 응답 정보가 포함된 JSON 파일을 보여줍니다. API 응답의 파라미터에 대한 자세한 내용은 GetDeviceProfile 섹션을 참조하세요.

GetDeviceProfile API 응답(***device_profile.json***의 내용)

```
{
    "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
    }
}
```

2단계: Sidewalk 디바이스 정보를 JSON 파일로 가져오기

GetWirelessDevice API 작업 또는 get-wireless-device CLI 명령을 사용하여 Amazon Sidewalk용 AWS IoT Core 계정에 추가한 Sidewalk 디바이스에 대한 정보를 가져옵니다. 엔드 디바이스에 대한 정보를 가져오려면 디바이스를 추가할 때 얻은 무선 디바이스의 식별자를 제공합니다.

그러면 API가 지정된 식별자 및 디바이스 ID와 일치하는 디바이스에 대한 정보를 반환합니다. 이 응답 정보를 JSON 파일로 저장합니다. 파일에 의미 있는 이름을 지정합니다(예 *wireless_device.json*).

다음은 CLI르 사용한 명령 실행의 예시를 보여줍니다.

```
aws iotwireless get-wireless-device --identifier-type WirelessDeviceId \
    --identifier "23456789-abcd-0123-bcde-fabc012345678" > wireless_device.json
```

이 명령을 실행하면 디바이스 세부 정보, 디바이스 인증서, 프라이빗 키 및 Sidewalk 제조 일련번호(SMSN)가 반환됩니다. 다음은 이 명령 실행의 예시 출력을 보여줍니다. API 응답의 파라미터에 대한 자세한 내용은 GetWirelessDevice 섹션을 참조하세요.

GetWirelessDevice API 응답(***wireless_device.json***의 내용)

```
{
    "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"
    },

    ...

}
```

다음 단계

JSON 파일 *wireless_device.json* 및 *device_profile.json*을 임시로 저장합니다. 이 파일은 다음 단계에서 하드웨어 플랫폼에 연결하기 위해 엔드 디바이스를 프로비저닝하고 등록하는 데 사용됩니다. 자세한 내용은 Amazon Sidewalk 설명서의 엔드 디바이스 프로비저닝 및 등록을 참조하세요.

Sidewalk 엔드 디바이스의 대상 추가

AWS IoT 규칙을 사용하여 데이터 및 디바이스 메시지를 처리하고 이를 다른 서비스로 라우팅하세요. 다른 서비스에서 메시지를 쉽게 사용할 수 있도록 디바이스에서 받은 바이너리 메시지를 처리하여 메시지를 다른 형식으로 변환하는 규칙도 정의할 수 있습니다. 대상은 Sidewalk 엔드 디바이스를 디바이스의 데이터를 처리하여 다른 AWS 서비스로 보내는 규칙과 연결합니다.

대상을 만들고 사용하는 방법

1. 대상에 대한 AWS IoT 규칙과 IAM 역할을 생성합니다. AWS IoT 규칙은 디바이스의 데이터를 처리하고 다른 AWS 서비스 및 애플리케이션에서 사용할 수 있도록 라우팅하는 규칙을 지정합니다. IAM 역할은 이 규칙에 액세스하는 권한을 부여합니다.
2. `CreateDestination` API 작업을 사용하여 Sidewalk 디바이스의 대상을 생성합니다. 대상 이름, 규칙 이름, 역할 이름 및 기타 선택적 파라미터를 지정합니다. API는 대상의 고유 식별자를 반환하며, 이 식별자는 엔드 디바이스를 Amazon Sidewalk용 AWS IoT Core에 추가할 때 지정할 수 있습니다.

다음은 대상, 대상에 대한 AWS IoT 규칙 및 IAM 역할을 생성하는 방법을 보여줍니다.

주제

- Sidewalk 디바이스의 대상 생성
- 대상에 대한 IAM 역할 및 IoT 규칙 생성

Sidewalk 디바이스의 대상 생성

대상 허브를 사용하거나 `CreateDestination`을 사용하여 Amazon Sidewalk용 AWS IoT Core 계정에 대상을 추가할 수 있습니다. 대상을 만들 때 다음을 지정하세요.

- Sidewalk 엔드 디바이스에 사용할 대상의 고유한 이름.

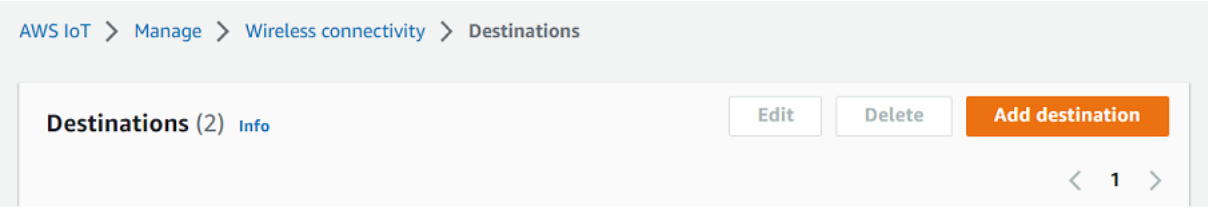
> **Note**
>
> 대상 이름을 사용하여 디바이스를 이미 추가한 경우 대상을 만들 때 해당 이름을 사용해야 합니다. 자세한 내용은 2단계: Sidewalk 디바이스 추가 단원을 참조하십시오.

- 디바이스 데이터를 처리하는 AWS IoT 규칙의 이름 및 메시지가 게시되는 주제.
- 규칙에 액세스하도록 디바이스의 데이터에 권한을 부여하는 IAM 역할.

다음 섹션에서는 대상에 대한 AWS IoT 규칙 및 IAM 역할을 생성하는 방법을 설명즙니다.

대상 생성 (콘솔)

AWS IoT 콘솔을 사용하여 대상을 만들려면 대상 허브로 이동하여 대상 추가를 선택합니다.



디바이스의 데이터를 처리하려면, 대상을 생성할 때 다음 필드를 지정하고 대상 추가를 선택합니다.

- 대상 세부 사항

  대상 이름을 입력하고 대상의 설명(선택 사항)을 입력합니다.

- 규칙 이름

  디바이스에서 보낸 메시지를 평가하고 디바이스의 데이터를 처리하도록 구성된 AWS IoT 규칙입니다. 규칙 이름이 대상에 매핑됩니다. 대상은 수신하는 메시지를 처리하기 위한 규칙이 필요합니다. AWS IoT 규칙을 호출하거나 AWS IoT 메시지 브로커에 게시하여 메시지를 처리하도록 선택할 수 있습니다.

  - 규칙 이름 입력(Enter a rule name)을 선택하는 경우 이름을 입력한 다음 복사(Copy)를 선택하여 규칙 이름(AWS IoT 규칙을 생성할 때 입력)을 복사합니다. 규칙 생성을 선택하여 지금 규칙을 생성하거나, AWS IoT 콘솔의 규칙 허브로 이동하여 해당 이름으로 규칙을 생성할 수 있습니다.

    규칙을 입력하고 고급(Advanced) 설정을 사용하여 주제 이름을 지정할 수도 있습니다. 주제 이름은 규칙 호출 중에 제공되며 규칙 내에서 `topic` 표현식을 사용하여 액세스됩니다. AWS IoT 규칙에 대한 자세한 내용은 AWS IoT 규칙을 참조하세요.

- AWS IoT 메시지 브로커에 게시를 선택하는 경우 주제 이름을 입력합니다. 그런 다음 MQTT 주제 이름을 복사하면 여러 구독자가 이 주제를 구독하여 해당 주제에 게시된 메시지를 받을 수 있습니다. 자세한 내용은 MQTT 주제를 참조하세요.

  대상에 대한 AWS IoT 규칙을 자세히 알아보려면 LoRaWAN 디바이스 메시지를 처리하는 규칙 생성을 참조하세요.
- 역할 이름

  규칙 이름(Rule name)에서 명명된 규칙에 액세스할 수 있는 디바이스의 데이터 권한을 부여하는 IAM 역할입니다. 콘솔에서 새 서비스 역할을 생성하거나 기존 서비스 역할을 선택합니다. 새 서비스 역할을 생성하는 경우 역할 이름(예: **SidewalkDestinationRole**)을 입력하거나 AWS IoT Core for LoRaWAN이 새 역할 이름을 생성할 수 있도록 비워 둘 수 있습니다. 그러면 AWS IoT Core for LoRaWAN이 사용자를 대신하여 적절한 권한이 있는 IAM 역할을 자동으로 생성합니다.

대상 생성 (CLI)

디바이스 프로필을 생성하려면 `CreateDestination` API 작업 또는 `create-destination` CLI 명령을 사용합니다. 예를 들어 다음 명령을 실행하면 Sidewalk 엔드 디바이스에 대한 대상이 생성됩니다.

```
aws iotwireless create-destination --name SidewalkDestination \
    --expression-type RuleName --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

이 명령을 실행하면 Amazon 리소스 이름(ARN)과 대상 이름 등 대상 세부 정보가 반환됩니다.

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/SidewalkDestination",
    "Name": "SidewalkDestination"
}
```

대상 생성에 대한 자세한 내용은 LoRaWAN 디바이스 메시지를 처리하는 규칙 생성을 참조하세요.

대상에 대한 IAM 역할 및 IoT 규칙 생성

AWS IoT 규칙은 디바이스 메시지를 다른 서비스로 전송합니다. 또한 AWS IoT 규칙은 Sidewalk 엔드 디바이스에서 수신한 바이너리 메시지를 처리하여 다른 서비스가 사용할 수 있도록 합니다. Amazon Sidewalk용 AWS IoT Core 대상은 무선 디바이스를 디바이스 메시지 데이터를 처리하여 다른 서비스로 전송하는 규칙과 연결합니다. 규칙은 Amazon Sidewalk용 AWS IoT Core에서 수신하는 즉시 디바

이스의 데이터에 적용됩니다. 동일한 서비스에 데이터를 보내는 모든 디바이스가 공유하는 대상을 만들 수 있습니다. 또한 규칙에 데이터를 전송할 권한을 부여하는 IAM 역할을 생성해야 합니다.

대상에 대한 IAM 역할 생성

AWS IoT 규칙에 데이터를 전송할 권한을 Amazon Sidewalk용 AWS IoT Core에 부여하는 IAM 역할을 생성합니다. 역할을 생성하려면 `CreateRole` API 작업 또는 `create-role` CLI 명령을 사용합니다. 역할 이름을 *`SidewalkRole`*로 지정할 수 있습니다.

```
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"}]}'
```

JSON 파일을 사용하여 역할에 대한 신뢰 정책을 정의할 수도 있습니다.

```
aws iam create-role --role-name SidewalkRole \
    --assume-role-policy-document file://trust-policy.json
```

다음은 JSON 파일의 내용을 보여줍니다.

trust-policy.json의 내용

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

대상에 대한 규칙 생성

AWS IoT Core API 작업 `CreateTopicRule` 또는 AWS CLI 명령 `create-topic-rule`을 사용하여 규칙을 생성합니다. 대상은 주제 규칙을 사용하여 Sidewalk 엔드 디바이스에서 수신한 데이터를 다른 AWS 서비스로 라우팅합니다. 예를 들어, Lambda 함수에 메시지를 보내는 규칙 작업을 생성할 수 있

습니다. 디바이스로부터 애플리케이션 데이터를 수신하고 다른 애플리케이션에서 사용할 수 있도록 base64를 사용하여 페이로드 데이터를 디코딩하도록 Lambda 함수를 정의할 수 있습니다.

다음 단계는 Lambda 함수를 생성한 다음 이 함수에 메시지를 보내는 주제 규칙을 생성하는 방법을 보여줍니다.

1. 실행 역할 및 정책 생성

   함수에 AWS 리소스에 액세스할 수 있는 권한을 제공하는 IAM 역할을 만듭니다. JSON 파일을 사용하여 역할에 대한 신뢰 정책을 정의할 수도 있습니다.

   ```
   aws iam create-role --role-name lambda-ex \
       --assume-role-policy-document file://lambda-trust-policy.json
   ```

   다음은 JSON 파일의 내용을 보여줍니다.

   lambda-trust-policy.json의 내용

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

2. Lambda 함수 생성 및 테스트

   base64가 페이로드 데이터를 디코딩하는 AWS Lambda 함수를 생성하려면 다음 단계를 수행하세요.

   a. 페이로드 데이터를 디코딩하기 위한 코드를 작성합니다. 예를 들어, 다음 샘플 Python 코드를 사용할 수 있습니다. 스크립트 이름을 지정합니다(예: *base64_decode.py*).

      base64_decode.py의 내용

      ```
      // ------------------------------------------------------------
      ```

```
// ----- 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 파일이 포함된 zip 파일로 배포 패키지를 만들고 이름을 *base64_decode.zip*으로 지정합니다. `CreateFunction` API 또는 `create-function` CLI 명령을 사용하여 샘플 코드 *base64_decode.py*용 Lambda 함수를 생성합니다.

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
```

다음과 같이 출력되어야 합니다. 주제 규칙을 생성할 때 출력의 Amazon 리소스 이름(ARN)인 `FunctionArn` 값을 사용합니다.

```
{
    "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. 명령줄에서 호출에 대한 로그를 가져오려면 invoke 명령과 함께 --log-type 옵션을 사용하세요. 호출에서 base64로 인코딩된 로그를 최대 4KB까지 포함하는 LogResult 필드가 응답에 포함됩니다.

```
aws lambda invoke --function-name my-function out --log-type Tail
```

StatusCode가 200인 응답을 받게 됩니다. AWS CLI를 통한 Lambda 함숨 생성 및 사용에 대한 자세한 내용은 AWS CLI에서 Lambda 사용을 참조하세요.

3. 주제 규칙 생성

CreateTopicRule API 또는 create-topic-rule CLI 명령을 사용하여 이 Lambda 함수에 메시지를 보내는 주제 규칙을 생성합니다. AWS IoT 주제에 다시 게시하는 두 번째 규칙 작업을 추가할 수도 있습니다. 이 주제 규칙의 이름을 *Sidewalkrule*로 지정합니다.

```
aws iot create-topic-rule --rule-name Sidewalkrule \
    --topic-rule-payload file://myrule.json
```

myrule.json 파일을 사용하여 규칙에 대한 자세한 내용을 지정할 수 있습니다. 예를 들어, 다음 JSON 파일은 AWS IoT 주제에 다시 게시하고 Lambda 함수에 메시지를 보내는 방법을 보여줍니다.

```
{
    "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 디바이스 연결 및 업링크 메타데이터 형식 보기

이 자습서에서는 MQTT 테스트 클라이언트를 사용하여 연결성을 테스트하고 엔드 디바이스와 AWS 클라우드 간에 교환되는 메시지를 확인합니다. 메시지를 수신하려면 MQTT 테스트 클라이언트에서 대상에 대한 IoT 규칙을 생성할 때 지정한 주제를 구독하세요. `SendDataToWirelessDevice` API 작업을 사용하여 Amazon Sidewalk용 AWS IoT Core에서 디바이스에 다운링크 메시지를 보낼 수도 있습니다. 메시지 전송 상태 이벤트 알림을 활성화하여 메시지가 전송되었는지 확인할 수 있습니다.

> **Note**
>
> 하드웨어 플랫폼 연결 및 설정에 대한 자세한 내용은 Amazon Sidewalk 설명서의 엔드 디바이스 프로비저닝 및 등록 및 Hardware Development Kit(HDK) 설정을 참조하세요.

엔드 디바이스로 다운링크 메시지 전송

`SendDataToWirelessDevice` API 작업 또는 `send-data-to-wireless-device` CLI 명령을 사용하여 Amazon Sidewalk용 AWS IoT Core에서 Sidewalk 엔드 디바이스로 다운링크 메시지를 보냅니다. 다음은 이 명령을 실행하는 방법의 예시를 보여줍니다. 페이로드 데이터는 base64로 인코딩된, 전송할 바이너리입니다.

```
aws iotwireless send-data-to-wireless-device \
    --id "<Wireless_Device_ID>" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

다음은 디바이스로 전송된 다운링크 메시지의 ID인 이 명령 실행의 샘플 출력입니다.

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

> **Note**
>
> SendDataToWirelessDevice API는 메시지 ID를 반환할 수 있지만 메시지가 성공적으로 전송되지 않을 수 있습니다. 디바이스로 전송된 메시지의 상태를 확인하려면 Sidewalk 계정 및 디바이스에 대한 메시지 전송 상태 이벤트를 활성화할 수 있습니다. 이러한 이벤트를 활성화하는 방법에 대한 자세한 내용은 Sidewalk 리소스에 대한 이벤트 알림 섹션을 참조하세요. 이 이벤트 유형에 대한 자세한 내용은 메시지 전송 이벤트를 참조하세요.

디바이스의 업링크 메시지 형식 보기

디바이스를 연결한 후 대상 규칙을 생성할 때 지정한 주제(예: *project/sensor/observed*)를 구독하고 디바이스의 업링크 메시지를 관찰할 수 있습니다.

대상을 만들 때 주제 이름을 지정한 경우 주제를 구독하여 엔드 디바이스의 업링크 메시지를 모니터링할 수 있습니다. AWS IoT 콘솔의 테스트 페이지에 있는 MQTT 테스트 클라이언트로 이동하여 주제 이름(예: *project/sensor/observed*)을 입력한 다음 구독을 선택합니다.

다음 예에서는 Sidewalk 디바이스에서 AWS IoT로 전송한 업링크 메시지의 형식을 보여줍니다. WirelessMetadata에는 메시지 요청에 대한 메타데이터가 포함되어 있습니다.

```
{
   "PayloadData":"ZjRlNjY1ZWNlNw==",
   "WirelessDeviceId":"wireless_device_id",
   "WirelessMetadata":{
      "Sidewalk":{
         "CmdExStatus":"Cmd",
         "SidewalkId":"device_id",
         "Seq":0,
         "MessageType":"messageType"
      }
    }
}
```

다음 표에서는 업링크 메타데이터의 여러 파라미터에 대한 정의를 보여 줍니다. 이 *device-id*는 *ABCDEF1234*와 같은 무선 디바이스의 ID이며, *messageType*은 디바이스에서 수신된 업링크 메시지의 유형입니다.

업링크 메타데이터 파라미터

파라미터	설명	유형	필수
`PayloadData`	무선 디바이스에서 전송되는 메시지 페이로드입니다.	String	예
`WirelessDeviceID`	데이터를 전송하는 무선 디바이스의 식별자입니다.	String	예
`Sidewalk.CmdExStatus`	명령 런타임 상태입니다. 응답 유형 메시지는 상태 코드 `COMMAND_EXEC_STATUS_SUCCESS` 를 포함해야 합니다. 그러나 알림은 상태 코드를 포함하지 않을 수 있습니다.	열거	아니요
`Sidewalk.NackExStatus`	응답 nack 상태(`RADIO_TX_ERROR` 또는 `MEMORY_ERROR`)입니다.	문자열 배열	아니요

Amazon Sidewalk용 AWS IoT Core로 디바이스 대량 프로비저닝

대량 프로비저닝을 사용하여 여러 개의 엔드 디바이스를 위해 대량으로 Amazon Sidewalk용 AWS IoT Core에 온보딩할 수 있습니다. 대량 프로비저닝은 공장에서 대량의 디바이스를 제조하고 이 디바이스를 AWS IoT에 온보딩하려는 경우에 특히 유용합니다. 디바이스 제조에 대한 자세한 내용은 Amazon Sidewalk 설명서에서 Amazon Sidewalk 디바이스 제조를 참조하세요.

다음 주제에서는 대량 프로비저닝의 작동 방식을 보여줍니다.

- Amazon Sidewalk 대량 프로비저닝 워크플로

  이 주제에서는 대량 프로비저닝의 몇 가지 주요 개념과 작동 방식을 보여줍니다. 또한 Sidewalk 디바이스를 Amazon Sidewalk용 AWS IoT Core로 가져오기 위해 수행해야 하는 단계도 보여 줍니다.
- 공장 지원이 적용된 디바이스 프로필 생성

  이 주제에서는 디바이스 프로필을 만들고 이에 대한 공장 지원을 받는 방법을 설명합니다. 또한 디바이스 제조 후 YubiHSM 키를 검색한 후 제조업체에 보내 제어 로그를 얻는 방법도 배우게 됩니다.
- 가져오기 작업을 사용하여 Sidewalk 디바이스 프로비저닝

이 주제에서는 가져오기 작업을 생성하고 사용하여 Sidewalk디바이스치를 대량으로 프로비저닝하는 방법을 보여줍니다. 또한 가져오기 작업을 업데이트하거나 삭제하는 방법과 작업에서 가져오기 작업 및 디바이스의 상태를 확인하는 방법을 알아봅니다.

주제

Amazon Sidewalk 대량 프로비저닝 워크플로

다음 섹션에서는 대량 프로비저닝의 주요 개념과 작동 방식을 보여줍니다. 대량 프로비저닝에 포함되는 단계는 다음과 같습니다.

1. Amazon Sidewalk용 AWS IoT Core를 사용하여 디바이스 프로필을 생성합니다.
2. Amazon Sidewalk 팀에 YubiHSM 키를 요청하고 공장 지원을 받아 디바이스 프로필을 업데이트하도록 요청합니다.
3. 디바이스 제조 후 Amazon Sidewalk용 AWS IoT Core가 제어 로그를 받을 수 있도록 YubiHSM 키를 제조업체로 보냅니다.
4. 가져오기 태스크를 생성하고 Amazon Sidewalk용 AWS IoT Core에 온보딩할 디바이스의 일련번호(SMSN)를 제공합니다.

대량 프로비저닝의 구성 요소

다음 개념은 대량 프로비저닝의 몇 가지 주요 구성 요소와 이를 Sidewalk 디바이스 대량 프로비저닝의 일부로 사용하는 방법을 보여줍니다.

YubiHSM 키

Amazon은 Sidewalk 제품 각각에 대해 하나 이상의 하드웨어 보안 모듈(HSM)을 생성합니다. 각 HSM에는 YubiHSM 키라는 고유한 일련번호가 하드웨어 모듈에 인쇄되어 있습니다. 이 키는 Yubico 웹 페이지에서 구매할 수 있습니다.

키는 HSM마다 고유하며 Amazon Sidewalk용 AWS IoT Core를 사용하여 생성하는 각 디바이스 프로필에 연결됩니다. YubiHSM 키를 받으려면 Amazon Sidewalk 팀에 문의하세요. YubiHSM 키를 제조업

체에 보내면 Sidewalk 디바이스가 공장에서 제조된 후 Amazon Sidewalk용 AWS IoT Core가 디바이스의 일련번호가 포함된 제어 로그 파일을 받게 됩니다. 그런 다음 이 정보를 디바이스를 AWS IoT에 온보딩하는 데 사용할 입력 CSV 파일과 비교합니다.

디바이스 증명 키(**DAK**)

Sidewalk 엔드 디바이스가 Sidewalk 네트워크에 조인되면 Sidewalk 디바이스 인증서로 프로비저닝되어야 합니다. 디바이스 설정에 사용되는 인증서에는 사설 디바이스별 인증서 및 Sidewalk 인증서 체인에 해당하는 퍼블릭 디바이스 인증서가 포함됩니다. Sidewalk 디바이스가 제조되면 YubiHSM이 디바이스 인증서에 서명합니다.

다음은 디바이스 인증서와 프라이빗 키가 포함된 샘플 JSON 파일입니다. 자세한 내용은 프로비저닝에 사용할 디바이스 JSON 파일 확보 단원을 참조하십시오.

```
{
  "p256R1": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbH ... DANKkOKoNT3bUGz+/f/pyTE
+xMRdIUBZ1Bw==",
  "eD25519": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbHD ... UiZmntHiUr1GfkTOFMYqRB+Aw==",
  "metadata": {
    "devicetypeid": "fe98",

    ...

    "devicePrivKeyP256R1":
 "3e704bf8d319b3a475179f1d68c60737b28c708f845d0198f2d00d00c88ee018",
    "devicePrivKeyEd25519":
 "17dacb3a46ad9a42d5c520ca5f47f0167f59ce54d740aa13918465faf533b8d0"
  },
  "applicationServerPublicKey":
 "5ce29b89c2e3ce6183b41e75fe54e45f61b8bb320efbdd2abd7aefa5957a316b"
}
```

디바이스 증명 키(DAK)는 디바이스 프로필을 만들 때 받는 프라이빗 키입니다. 이는 각 Sidewalk 제품에 발급되는 고유한 인증서인 제품 인증서에 해당합니다. Amazon Sidewalk 팀에 문의하면 Sidewalk 인증서 체인, YubiHSM 키 및 제품 디바이스 증명 키(DAK)로 프로비저닝된 HSM을 받게 됩니다.

디바이스 프로필도 새로운 디바이스 증명 키(DAK)로 업데이트되고 공장 지원이 활성화됩니다. 디바이스 프로필의 DAK 메타데이터 정보는 DAK 이름, 인증서 ID, 광고 대상 제품 ID(ApID), 공장 지원 활성화 여부, DAK가 서명할 수 있는 최대 서명 수와 같은 세부 정보를 제공합니다.

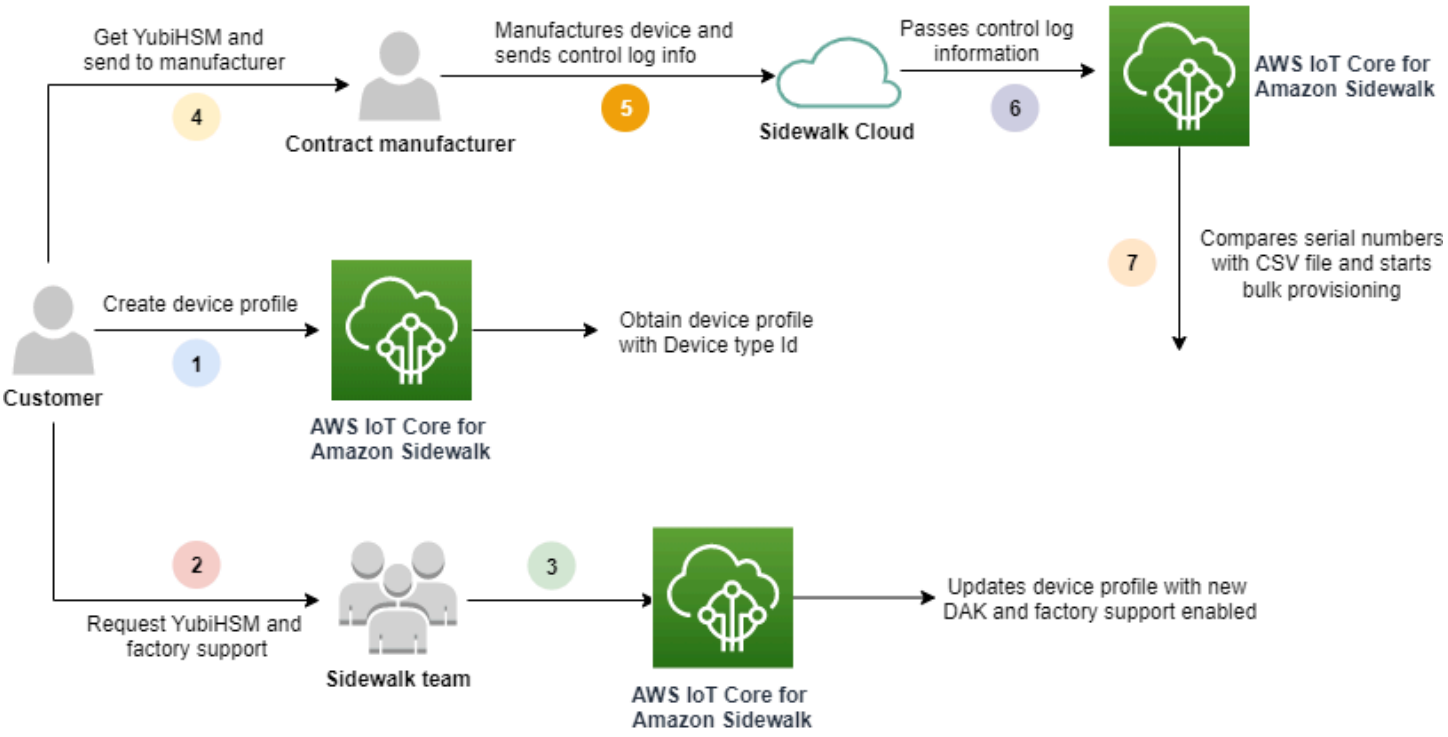
광고 대상 제품 ID(**ApId**)

`ApId` 파라미터는 광고 대상 제품을 식별하는 영숫자 문자열입니다. 대량 프로비저닝하는 Sidewalk 디바이스에 특정 디바이스 프로필을 사용하려는 경우 이 필드를 지정해야 합니다. 그러면 Amazon Sidewalk용 AWS IoT Core가 DAK를 생성하여 YubiHSM 키를 통해 제공합니다. 관련 DAK 정보는 디바이스 프로필에 표시됩니다.

`ApId`를 얻으려면 생성한 디바이스 프로필에 대한 정보를 검색한 후 Amazon Sidewalk 지원 팀에 문의하세요. AWS IoT 콘솔에서, `GetDeviceProfile` API 작업 또는 `get-device-profile` CLI 명령을 사용하여 디바이스 프로파일 정보를 얻을 수 있습니다.

대량 프로비저닝 작동 방식

이 순서도는 Amazon Sidewalk용 AWS IoT Core를 사용하여 대량 프로비저닝이 작동하는 방식을 보여줍니다.



다음 절차는 대량 프로비저닝 프로세스의 각기 다른 단계를 보여줍니다.

1. Sidewalk 디바이스용 디바이스 프로필 생성

   엔드 디바이스를 공장으로 가져가기 전에 먼저 디바이스 프로필을 생성하세요. 디바이스 프로필 및 Sidewalk 엔드 디바이스 추가에 설명된 대로 이 프로필을 사용하여 개별 디바이스를 프로비저닝할 수 있습니다.

2. 프로필에 대한 공장 지원 요청

   엔드 디바이스를 공장으로 가져갈 준비가 되면 Amazon Sidewalk 팀에 YubiHSM 키와 디바이스 프로필에 대한 공장 지원을 요청하세요.

3. DAK 및 공장 지원 프로필 확보

   그러면 Amazon Sidewalk 지원 팀에서 제품 디바이스 증명 키(DAK)로 디바이스 프로필을 업데이트하고 공장 지원을 추가합니다. 디바이스 프로필은 광고 대상 제품 ID(APID)와 새 DAK 및 인증서 정보(예: 인증서 ID)로 자동 업데이트됩니다. 이 프로필을 사용하는 Sidewalk 디바이스는 대량 프로비저닝에 사용할 수 있습니다.

4. YubiHSM 키를 제조업체(CM)에 전송

   이제 엔드 디바이스에 자격이 부여되었으므로 YubiHSM 키를 계약 제조업체(CM)에 보내 제조 프로세스를 시작할 수 있습니다. 자세한 내용은 Amazon Sidewalk 설명서에서 Amazon Sidewalk 디바이스 제조를 참조하세요.

5. 디바이스 제조 및 제어 로그와 일련번호 전송

   CM은 디바이스를 제조하고 제어 로그를 생성합니다. CM은 제조할 디바이스 목록과 해당 Sidewalk 제조 일련번호(SMSN)가 포함된 CSV 파일도 제공합니다. 다음 코드는 샘플 제어 로그입니다. 여기에는 디바이스의 일련번호, APID 및 퍼블릭 디바이스 인증서가 포함됩니다.

```
{
    "controlLogs": [
    {
        "version": "4-0-1",
        "device":
        {
            "serialNumber": "device1",
            "productIdentifier": {
                "advertisedProductId": "abCD"
             },
             "sidewalkData": {
                "SidewalkED25519CertificateChain": "...",
                "SidewalkP256R1CertificateChain": "..."
             }
         }
      }
   ]
}
```

6. Amazon Sidewalk용 AWS IoT Core에 제어 로그 정보 전달

   Amazon Sidewalk 클라우드는 제조업체로부터 제어 로그 정보를 검색하고 이 정보를 Amazon Sidewalk용 AWS IoT Core에 전달합니다. 그러면 일련번호와 함께 디바이스를 생성할 수 있습니다.

7. 일련번호 일치 여부확인 및 대량 프로비저닝 시작

   AWS IoT 콘솔 또는 Amazon Sidewalk용 AWS IoT Core API 작업 `StartWirelessDeviceImportTask`를 사용하여 Amazon Sidewalk용 AWS IoT Core는 Amazon Sidewalk에서 얻은 각 디바이스의 Sidewalk 제조 일련번호(SMSN)를 CSV 파일의 해당 일련번호와 비교합니다. 이 정보가 일치하면 대량 프로비저닝 프로세스를 시작하고 Amazon Sidewalk용 AWS IoT Core로 가져올 디바이스를 생성합니다.

공장 지원이 적용된 디바이스 프로필 생성

Amazon Sidewalk 디바이스를 대량 프로비저닝하려면 먼저 디바이스 프로필을 생성한 다음 Amazon Sidewalk 지원 팀에 문의하여 공장 지원을 요청해야 합니다. 그러면 Amazon Sidewalk 팀에서 새 디바이스 증명 키(DAK)로 디바이스 프로필을 업데이트하고 여기에 공장 지원을 추가합니다. 그러면 이 프로필을 사용하는 Sidewalk 디바이스에는 Amazon Sidewalk용 AWS IoT Core와 함께 사용할 수 있는 자격이 부여되고 대량 프로비저닝을 위해 디바이스를 온보딩할 수 있습니다.

다음 단계는 공장 지원이 적용된 디바이스 프로필을 만드는 방법을 보여줍니다.

1. 를 사용하여 디바이스 프로필을 생성합니다.

   먼저 디바이스 프로필을 생성합니다. 프로필을 만들 때 이름과 선택적 태그를 이름-값 페어로 지정합니다. 필요한 파라미터와 프로필 생성 및 사용에 대한 자세한 내용은 디바이스 생성 및 추가 방법 섹션을 참조하세요.

2. 프로필에 대한 공장 지원 확보

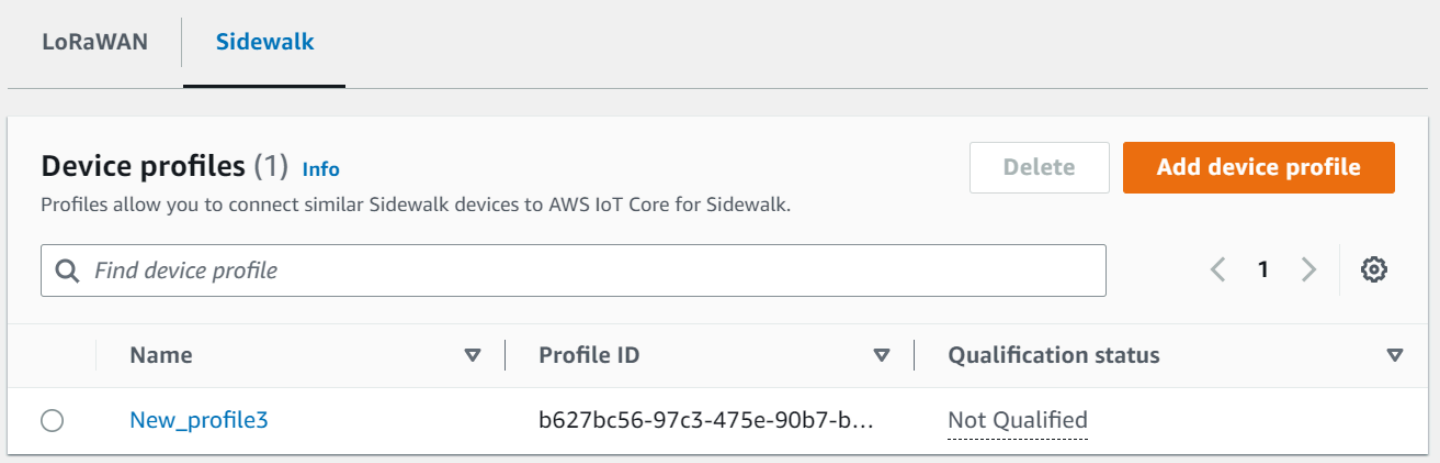
   그런 다음 이 프로필을 사용하는 디바이스에 자격을 부여할 수 있도록 디바이스 프로필에 대한 공장 지원을 확보합니다. 자격 부여를 위해 Amazon Sidewalk 팀에 티켓을 생성하세요. 팀에서 확인하면 광고 대상 제품 ID(ApId)를 받게 되며, 프로필은 공장에서 발급한 DAK로 업데이트됩니다. 이 프로필을 사용하는 Sidewalk 엔드 디바이스에 자격이 부여됩니다.

AWS IoT 콘솔, Amazon Sidewalk용 AWS IoT Core API 작업 또는 AWS CLI를 사용하여 디바이스 프로필을 생성할 수 있습니다.

주제

프로필 생성(콘솔)

AWS IoT 콘솔을 사용하여 디바이스 프로필을 만들려면 프로필 허브의 Sidewalk 탭으로 이동하여 프로필 생성을 선택합니다.



프로필을 만들려면 다음 필드를 지정한 다음 제출을 선택합니다.

- 명칭

  프로필의 이름을 입력합니다.

- Tags

  선택적 태그를 이름-값 페어로 입력하면 프로필을 더 쉽게 식별할 수 있습니다. 또한 태그를 사용하면 청구 요금을 더 쉽게 추적할 수 있습니다.

프로필 정보 보기 및 프로필 자격 부여

생성한 프로필을 프로필 허브에서 볼 수 있습니다. 세부 정보를 보려면 프로필을 선택합니다. 다음에 대한 정보가 표시됩니다.

- 디바이스 프로필 이름, 고유 식별자, 이름-값 페어로 지정한 선택적 태그
- 프로필의 애플리케이션 서버 퍼블릭 키 및 디바이스 유형 ID

- 자격 상태. 공장 지원이 적용되지 않는 디바이스 프로필을 사용하고 있음을 나타냅니다. 공장 지원이 적용되도록 디바이스 프로필에 자격을 부여하려면 Amazon Sidewalk Support에 문의하세요.
- 디바이스 증명 키(DAK) 정보. 디바이스 프로필에 자격이 부여되면 새 DAK가 발급되고 프로필이 새 DAK 정보로 자동으로 업데이트됩니다.

프로필 생성(CLI)

디바이스 프로필을 생성하려면 `CreateDeviceProfile` API 작업 또는 `create-device-profile` CLI 명령을 사용합니다. 예를 들어 다음 명령을 실행하면 Sidewalk 엔드 디바이스에 대한 프로필이 생성됩니다.

```
aws iotwireless create-device-profile \
    --name sidewalk_device_profile --sidewalk {}
```

이 명령을 실행하면 프로필의 Amazon 리소스 이름(ARN)과 ID 등 프로필 세부 정보가 반환됩니다.

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

프로필 정보 보기 및 프로필 자격 부여

`GetDeviceProfile` API 작업 또는 `get-device-profile` CLI 명령을 사용하여 Amazon Sidewalk용 AWS IoT Core 계정에 추가한 디바이스 프로필에 대한 정보를 가져옵니다. 디바이스 프로필에 대한 정보를 검색하려면 프로필 ID를 지정합니다. 그러면 API가 지정된 식별자와 일치하는 디바이스 프로필에 대한 정보를 반환합니다.

다음은 CLI 명령의 예시입니다.

```
aws iotwireless get-device-profile \
    --id "12345678-234a-45bc-67de-e8901234f0a1" > device_profile.json
```

이 명령을 실행하면 디바이스 프로필의 파라미터, 애플리케이션 서버 퍼블릭 키, `DeviceTypeId`, `ApId`, 자격 상태 및 `DAKCertificate` 정보가 반환됩니다.

이 예시에서 자격 상태 및 DAK 정보는 디바이스 프로필에 자격이 없음을 나타냅니다. 프로필에 자격을 부여하려면 Amazon Sidewalk Support에 문의하세요. 그러면 프로필에 디바이스 제한이 없는 새 DAK가 발급됩니다.


```
{
    "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 지원 팀에서 이 정보를 확인하면 다음 예시와 같이 APID와 공장 지원이 적용된 DAK를 받게 됩니다.

> **Note**
>
> `MaxAllowedSignature`의 값이 `-1`이라는 것은 DAK에 디바이스 제한이 없음을 나타냅니다. DAK 파라미터에 대한 자세한 내용은 DAKCertificateMetadata를 참조하세요.


```
{
    "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
    }
}
```

다음 단계

공장 지원이 적용된 DAK가 있는 디바이스 프로필을 생성했으니 이제 팀에게서 받은 YubiHSM 키를 제조업체에 제공합니다. 그러면 디바이스가 공장에서 제조되고 디바이스의 일련번호(SMSN)가 포함된 제어 로그 정보가 Amazon Sidewalk로 전달됩니다. 이 워크플로에 대한 자세한 내용은 Amazon Sidewalk 설명서에서 Amazon Sidewalk 디바이스 제조를 참조하세요.

그런 다음 온보딩할 디바이스의 일련번호를 Amazon Sidewalk용 AWS IoT Core에 제공하여 Sidewalk 디바이스를 대량 프로비저닝할 수 있습니다. Amazon Sidewalk용 AWS IoT Core가 제어 로그를 수신하면 제어 로그의 일련번호를 사용자가 제공한 일련번호와 비교합니다. 일련번호가 일치하면 가져오기 태스크가 디바이스를 Amazon Sidewalk용 AWS IoT Core에 온보딩하기 시작합니다. 자세한 내용은 가져오기 작업을 사용하여 Sidewalk 디바이스 프로비저닝 단원을 참조하십시오.

가져오기 작업을 사용하여 Sidewalk 디바이스 프로비저닝

이 단원에서는 AWS IoT 콘솔, Amazon Sidewalk용 AWS IoT Core API 작업 또는 AWS CLI를 사용하여 Sidewalk 디바이스를 대량으로 프로비저닝하는 방법을 설명즙니다. 다음 섹션에서는 Sidewalk 디바이스를 대량 프로비저닝하는 방법을 설명합니다.

주제

- Sidewalk 대량 프로비저닝 작동 방법
- Sidewalk 대량 프로비저닝의 주요 고려 사항
- CSV 파일 형식

- Sidewalk 대량 프로비저닝 사용 방법
- Sidewalk 디바이스 대량 프로비저닝
- 가져오기 작업 및 디바이스 온보딩 상태 보기

Sidewalk 대량 프로비저닝 작동 방법

다음 단계는 대량 프로비저닝의 작동 방식을 보여줍니다.

1. 무선 디바이스 가져오기 작업 시작

   Sidewalk 디바이스를 대량으로 프로비저닝하려면 가져오기 태스크를 생성하고 Amazon Sidewalk용 AWS IoT Core에 온보딩할 디바이스의 Sidewalk 제조 일련번호(SMSN)를 제공해야 합니다. 제조업체가 Amazon Sidewalk에 제어 로그를 업로드한 후 이메일을 통해 디바이스의 Sidewalk 제조 일련번호(SMSN)를 CSV 파일로 받았습니다. 이 워크플로와 제어 로그를 받는 방법에 대한 자세한 내용은 Amazon Sidewalk 설명서에서 Amazon Sidewalk 디바이스 제조를 참조하세요.

2. 백그라운드에서 가져오기 프로세스 실행

   Amazon Sidewalk용 AWS IoT Core가 가져오기 태스크 요청을 받으면 태스크 설정을 시작하고 시스템을 자주 폴링하는 백그라운드 프로세스를 시작합니다. 백그라운드 프로세스가 가져오기 태스크 지침을 받으면 CSV 파일을 읽기 시작합니다. 동시에 Amazon Sidewalk용 AWS IoT Core는 Amazon Sidewalk로부터 제어 로그가 수신되었는지 확인합니다.

3. 무선 디바이스 레코드 생성

   Amazon Sidewalk에서 제어 로그를 수신하면 Amazon Sidewalk용 AWS IoT Core는 제어 로그의 일련번호가 CSV 파일의 SMSN 값과 일치하는지 확인합니다. 일련번호가 일치하면 Amazon Sidewalk용 AWS IoT Core는 주어진 일련번호에 해당하는 Sidewalk 디바이스에 대한 무선 디바이스 레코드를 생성하기 시작합니다. 모든 디바이스가 온보딩되면 가져오기 작업이 완료됨으로 표시됩니다.

Sidewalk 대량 프로비저닝의 주요 고려 사항

Sidewalk 디바이스를 Amazon Sidewalk용 AWS IoT Core에 대량으로 프로비저닝할 때 고려해야 할 몇 가지 주요 사항은 다음과 같습니다.

- 디바이스 프로필을 생성한 AWS 계정에서 AWS IoT 콘솔 또는 Amazon Sidewalk용 AWS IoT Core API 작업을 사용하여 대량 프로비저닝을 수행해야 합니다.

- Sidewalk 디바이스를 대량 프로비저닝하기 전에 디바이스 프로필에 공장 지원을 나타내는 DAK 정보가 이미 포함되어 있어야 합니다. 그렇지 않으면 AWS IoT 콘솔을 사용한 대량 프로비저닝이나 대량 프로비저닝 API 작업이 실패할 수 있습니다.
- 가져오기 태스크를 시작한 후 CSV 파일을 처리하고, 무선 디바이스를 가져오고, Amazon Sidewalk용 AWS IoT Core에 온보딩하는 데 최소 10분이 걸릴 수 있습니다.
- 무선 디바이스 가져오기 작업은 시작되면 90일 동안 실행됩니다. 이 기간 동안 Amazon Sidewalk로부터 제어 로그를 수신했는지 확인합니다. 90일 이전에 Amazon Sidewalk로부터 제어 로그를 받지 못한 경우, 작업 세부 정보를 볼 때 만료되었다는 메시지와 함께 작업이 완료됨으로 표시됩니다. 가져오기 작업에서 제어 로그를 기다리고 있던 디바이스의 온보딩 상태는 실패로 표시됩니다.
- 이미 생성한 가져오기 작업을 업데이트하려고 시도할 때는 작업에 다른 디바이스를 추가만 할 수 있습니다. 가져오기 작업을 생성한 후 가져오기 작업에 이미 추가된 디바이스에서 작업이 시작되기 전에는 언제든지 새 디바이스를 추가할 수 있습니다. 원래 가져오기 작업에 이미 있는 디바이스의 일련번호가 업데이트 파일에 포함된 경우 해당 일련번호는 무시됩니다.
- 업데이트 작업을 요청하면 가져오기 작업을 생성할 때 사용한 것과 동일한 IAM 역할이 Amazon S3 버킷의 CSV 파일에 액세스하는 것으로 간주됩니다.
- 작업이 이미 성공적으로 완료되었거나 작업이 업데이트되지 않은 경우에만 가져오기 작업을 삭제할 수 있습니다. 잘못된 IAM 역할을 제공했거나 Amazon S3 버킷 파일을 찾을 수 없는 경우 작업이 업데이트되지 않을 수 있습니다. 가져오기 작업이 `PENDING` 상태인 경우 작업을 업데이트하거나 삭제할 수 없습니다.
- 작업으로 가져오는 CSV 파일은 다음 섹션에 설명된 형식을 사용해야 합니다.

CSV 파일 형식

가져오기 태스크에 지정하는 Amazon S3 버킷에 포함된 CSV 파일은 다음 형식을 사용해야 합니다.

- 행 1은 키워드 `smsn`을 사용해야 합니다. 이 키워드는 가져오는 CSV 파일에 가져올 디바이스의 SMSN이 포함되어 있음을 나타냅니다.
- 2행 이후의 행에는 온보딩할 디바이스의 SMSN을 포함해야 합니다. 디바이스 SMSN은 64자리 16진수 문자 형식이어야 합니다.

이 JSON 파일은 샘플 CSV 파일 형식을 보여줍니다.

```
smsn
1C1A10B0AC0A200C012BBAC2CBB1B21CB12C0CA2AC1C1BB22CAA01C1B0B01122
B122C2B1121BACA2221001AC1B22012AAC11112C11C2A100C1C2B012A1100C10
02B222C110B0A210B0A0C2C112CCCAC21C1C0B0AA1221AB1022A2CC11B1B1122
```

```
C2C021CA1C111CCAB1221C0021C1C2AAA0AA1A2A01ABC10CBAACCA2A0121022A
0CB22C01BBC2CA2C0B11001121ACB2ABB0BB0121C2BA101C012CC2B20C011AC0
```

Sidewalk 대량 프로비저닝 사용 방법

다음 단계는 Amazon Sidewalk 대량 프로비저닝을 사용하는 방법을 보여줍니다.

1. 디바이스 일련번호 제공

   Sidewalk 디바이스를 프로비저닝하려면 온보딩할 디바이스의 일련번호를 제공해야 합니다. 다음 방법 중 하나를 사용하여 디바이스를 프로비저닝할 수 있습니다.

   - Sidewalk 제조 일련번호(SMSN)를 사용하여 각 디바이스를 개별적으로 프로비저닝합니다. 이 방법은 적절한 IAM 역할이 포함된 CSV 파일을 업로드하거나 디바이스가 작업에 온보딩할 준비가 될 때까지 기다릴 필요 없이 워크플로를 테스트하고 디바이스를 더 빠르게 온보딩하려는 경우에 유용합니다.
   - CSV 파일에 프로비저닝할 디바이스의 SMSN이 포함된 Amazon S3 버킷 URL을 제공하여 디바이스를 대량으로 프로비저닝합니다. 이 방법은 온보딩할 디바이스의 수량이 많을 때 특히 유용합니다. 이 경우 각 디바이스를 개별적으로 온보딩하는 것은 따분한 일일 수 있습니다. 대신 Amazon S3 버킷에 업로드된 CSV 파일의 경로와 파일에 액세스할 수 있는 IAM 역할만 제공하면 됩니다.
2. 가져오기 작업 및 디바이스 온보딩 상태 확보

   생성하는 각 가져오기 작업에 대해 작업 온보딩 상태 및 작업에 추가된 디바이스의 온보딩 상태에 대한 정보를 검색할 수 있습니다. 또한 작업 또는 디바이스 온보딩이 실패한 이유와 같은 추가 상태 정보도 볼 수 있습니다. 자세한 내용을 알아보려면 다음 섹션을 참조하세요.
3. (선택 사항) 가져오기 작업 업데이트 또는 삭제

   이미 생성한 가져오기 작업을 업데이트하거나 삭제할 수 있습니다.

   - 이미 추가된 디바이스에서 태스크가 시작되기 전에 언제든지 가져오기 태스크를 업데이트하고 태스크에 디바이스를 추가할 수 있습니다. Amazon Sidewalk용 AWS IoT Core는 가져오기 태스크를 생성할 때 사용한 역할과 동일한 IAM 역할을 맡습니다. 작업을 생성할 때 작업에 추가하려는 디바이스의 일련번호가 들어 있는 새 CSV 파일을 지정하세요.

     > **Note**
     >
     > 기존 가져오기 태스크를 업데이트할 때는 태스크에 디바이스를 추가만 할 수 있습니다. Amazon Sidewalk용 AWS IoT Core는 이미 가져오기 태스크에 있는 디바이스와 태스크

에 추가하려는 디바이스 간에 통합 태스크를 수행합니다. 가져오기 작업에 이미 있는 디바이스의 일련번호가 새로운 파일에 포함된 경우 해당 일련번호는 무시됩니다.

- 이미 성공적으로 완료된 가져오기 작업이나 IAM 역할 정보가 잘못된 경우 또는 작업을 생성하거나 업데이트할 때 S3 버킷 파일을 사용할 수 없는 경우 업데이트가 실패한 가져오기 작업을 삭제할 수 있습니다.

주제

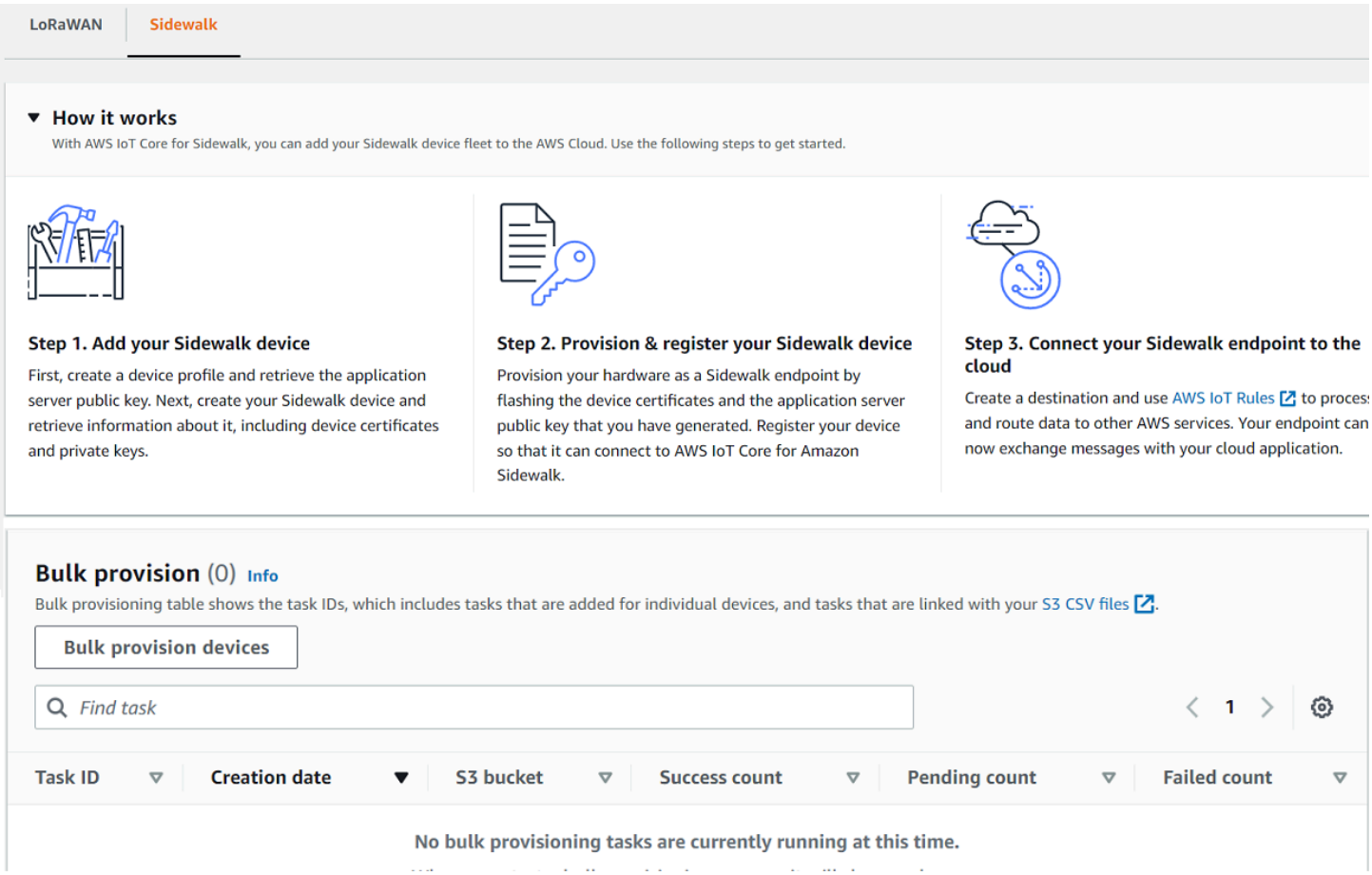
- Sidewalk 디바이스 대량 프로비저닝
- 가져오기 작업 및 디바이스 온보딩 상태 보기

Sidewalk 디바이스 대량 프로비저닝

이 단원에서는 AWS IoT 콘솔과 AWS CLI를 사용하여 Sidewalk 디바이스를 Amazon Sidewalk용 AWS IoT Core에 대량으로 프로비저닝하는 방법을 보여 줍니다.

Sidewalk 디바이스 대량 프로비저닝(콘솔)

AWS IoT 콘솔을 사용하여 Sidewalk 디바이스를 추가하려면 디바이스 허브의 Sidewalk 탭으로 이동하여 디바이스 대량 프로비저닝을 선택한 후 다음 단계를 수행하세요.



1. 가져오기 방법 선택

   Amazon Sidewalk용 AWS IoT Core에 대량으로 온보딩할 디바이스를 가져오는 방법을 지정하세요.

   - SMSN을 사용하여 개별 디바이스를 프로비저닝하려면 개별 공장 지원 디바이스 프로비저닝을 선택합니다.
   - 디바이스 목록과 해당 SMS가 포함된 CSV 파일을 제공하여 디바이스를 대량으로 프로비저닝하려면 S3 버킷 사용을 선택합니다.

2. 온보딩할 디바이스 지정

   디바이스를 온보딩하기 위해 선택한 방법에 따라 디바이스 정보와 일련번호를 추가합니다.

   a. 개별 공장 지원 디바이스 프로비저닝을 선택한 경우 다음 정보를 지정하세요.

      i. 온보딩할 각 디바이스의 이름. 이름은 AWS 계정 및 AWS 리전에서 고유해야 합니다.

      ii. SMSN 입력 필드에 디바이스의 Sidewalk 제조 일련번호(SMSN) 제공

iii. 디바이스에서 다른 AWS 서비스로 메시지를 라우팅하는 IoT 규칙을 설명하는 대상

b. S3 버킷 사용을 선택한 경우:

i. S3 URL 정보로 구성된 S3 버킷 대상 정보를 제공합니다. CSV 파일을 제공하려면 S3 찾아보기를 선택한 다음 사용하려는 CSV 파일을 선택합니다.

Amazon Sidewalk용 AWS IoT Core는 S3 버킷에 있는 CSV 파일의 경로인 S3 URL을 자동으로 채웁니다. 경로의 형식은 `s3://bucket_name/file_name`입니다. Amazon Simple Storage Service 콘솔에서 파일을 보려면 보기를 선택합니다.

ii. 사용자 대신 S3 버킷의 CSV 파일에 대한 액세스를 Amazon Sidewalk용 AWS IoT Core에 허용하는 S3 프로비저닝 역할을 제공합니다. 새로운 서비스 역할을 만들거나 기존 역할을 선택할 수 있습니다.

새 역할을 생성하려면 역할 이름을 제공할 수도 있고 임의의 이름이 자동으로 생성되도록 비워 둘 수도 있습니다.

iii. 디바이스에서 다른 AWS 서비스로 메시지를 라우팅하는 IoT 규칙을 설명하는 대상을 제공합니다.

3. 가져오기 작업 시작

모든 옵션 태그를 이름-값 페어로 제공하고 제출을 선택하여 무선 디바이스 가져오기 작업을 시작합니다.

Sidewalk 디바이스 대량 프로비저닝(CLI)

Sidewalk 디바이스를 Amazon Sidewalk용 AWS IoT Core 계정에 온보딩하려면 디바이스를 개별적으로 추가할지 또는 S3 버킷에 포함된 CSV 파일을 제공할지에 따라 다음 API 작업 중 하나를 사용하세요.

- S3 CSV 파일을 사용하여 디바이스 대량 업로드

  S3 버킷에 CSV 파일을 제공하여 디바이스를 대량으로 업로드하려면 `StartWirelessDeviceImportTask` API 작업 또는 `start-wireless-device-import-task` AWS CLI 명령을 사용합니다. 태스크를 생성할 때 Amazon S3 버킷의 CSV 파일 경로와 CSV 파일에 액세스할 권한을 Amazon Sidewalk용 AWS IoT Core에 부여하는 IAM 역할을 지정합니다.

  태스크가 실행되기 시작하면 Amazon Sidewalk용 AWS IoT Core는 CSV 파일 읽기를 시작하고 파일에 있는 일련번호(SMSN)를 Amazon Sidewalk에서 수신한 제어 로그의 해당 정보와 비교합니다. 일련번호가 일치하면 는 주어진 일련번호에 해당하는 무선 디바이스 레코드를 생성하기 시작합니다.

다음 명령은 가져오기 작업 생성의 예시를 보여줍니다.

```
aws iotwireless start-wireless-device-import-task \
    --cli-input-json "file://task.json"
```

다음은 task.json 파일의 콘텐츠를 보여줍니다.

task.json의 콘텐츠

```
{
    "DestinationName": "Sidewalk_Destination",
    "Sidewalk": {
        "DeviceCreationFile": "s3://import_task_bucket/import_file1",
        "Role": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI"
    }
}
```

이 명령을 실행하면 가져오기 작업의 ID와 ARN이 반환됩니다.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"
    "Id": "a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"
}
```

- SMSN을 사용하여 디바이스 개별 프로비저닝

  SMSN을 사용하여 디바이스를 개별적으로 프로비저닝하려면 StartSingleWirelessDeviceImportTask API 작업 또는 start-single-wireless-device-import-task AWS CLI 명령을 사용하세요. 작업을 생성할 때 Sidewalk 대상과 온보딩하려는 디바이스의 일련번호를 지정하세요.

  일련번호가 Amazon Sidewalk에서 수신한 제어 로그의 해당 정보와 일치하면 작업이 실행되고 무선 디바이스 레코드가 생성됩니다.

  다음 명령은 가져오기 작업 생성의 예시를 보여줍니다.

```
aws iotwireless start-single-wireless-device-import-task \
    --destination-name sidewalk_destination \
```

```
    --sidewalk
 '{"SidewalkManufacturingSn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A
```

이 명령을 실행하면 가져오기 작업의 ID와 ARN이 반환됩니다.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/e2a5995e-743b-41f2-
a1e4-3ca6a5c5249f"
    "Id": "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
}
```

가져오기 작업 업데이트 또는 삭제

가져오기 작업에 디바이스를 더 추가하려는 경우 작업을 업데이트하면 됩니다. 작업이 더 이상 필요하지 않거나 실패한 경우에도 작업을 삭제할 수 있습니다. 작업 업데이트 또는 삭제하는 상황에 대한 자세한 내용은 Sidewalk 대량 프로비저닝 사용 방법 섹션을 참조하세요.

> **Warning**
>
> 삭제 작업은 영구적이며 취소할 수 없습니다. 이미 성공적으로 완료된 가져오기 작업은 삭제해도 해당 작업을 사용하여 이미 온보딩된 엔드 디바이스는 제거되지 않습니다.

가져오기 작업을 업데이트 또는 삭제하는 방법:

- AWS IoT 콘솔 사용

  다음 단계에서는 AWS IoT 콘솔을 사용하여 가져오기 작업을 업데이트하거나 삭제하는 방법을 설명합니다.

  가져오기 작업을 업데이트하는 방법:

  1. AWS IoT 콘솔의 Sidewalk 디바이스 허브로 이동합니다.
  2. 업데이트할 가져오기 작업을 선택한 후 편집을 선택합니다.
  3. 작업에 추가할 디바이스의 일련번호가 들어 있는 다른 S3 파일을 제공한 다음 제출을 선택합니다.

가져오기 작업을 삭제하는 방법:

1. AWS IoT 콘솔의 Sidewalk 디바이스 허브로 이동합니다.
2. 삭제하려는 작업을 선택한 다음 삭제를 선택합니다.

- AWS IoT 무선 API 또는 AWS CLI 사용

  다음 AWS IoT 무선 API 태스크 또는 CLI 명령을 사용하여 가져오기 태스크를 업데이트하거나 삭제합니다.

  - **UpdateWirelessDeviceImportTask** API 또는 **update-wireless-device-import-task** CLI

    이 API 작업은 Amazon S3 CSV 파일의 콘텐츠를 기존 가져오기 작업에 추가합니다. 이전에 작업에 포함되지 않은 디바이스의 일련번호만 추가할 수 있습니다.

  - **DeleteWirelessDeviceImportTask** API 또는 **delete-wireless-device-import-task** CLI

    이 API 작업은 가져오기 작업 ID를 사용하여 삭제 대상으로 표시된 가져오기 작업을 삭제합니다.

가져오기 작업 및 디바이스 온보딩 상태 보기

작업에 추가한 무선 디바이스 가져오기 작업 및 Sidewalk 디바이스에 대해 다음 중 하나의 상태 메시지가 표시될 수 있습니다. 이러한 메시지는 AWS IoT 콘솔에 표시되거나 AWS IoT 무선 API 태스크 또는 AWS CLI 명령을 사용하여 이러한 태스크 및 해당 디바이스에 대한 정보를 검색할 때 표시됩니다.

가져오기 작업의 상태 정보 보기

가져오기 작업을 생성한 후에는 생성한 가져오기 작업과 작업에 추가된 디바이스의 온보딩 상태를 볼 수 있습니다. 온보딩 상태는 온보딩 보류 중인 디바이스 수, 성공적으로 온보딩된 디바이스 수, 온보딩에 실패한 디바이스 수를 나타냅니다.

가져오기 작업이 방금 생성되었으면 보류 중인 개수에는 추가된 디바이스 수에 해당하는 값이 표시됩니다. 작업이 시작되고 CSV 파일을 읽어 무선 디바이스 레코드를 만들면 보류 중인 개수가 줄어들고 디바이스가 성공적으로 온보딩되면 성공 횟수가 증가합니다. 온보딩에 실패하는 디바이스가 있으면 실패한 횟수가 증가합니다.

가져오기 작업 및 디바이스 온보딩 상태를 보는 방법:

- AWS IoT 콘솔 사용

생성한 가져오기 작업과 디바이스의 온보딩 상태 정보 요약의 수를 AWS IoT 콘솔의 Sidewalk devices 허브에서 확인할 수 있습니다. 생성한 가져오기 작업의 세부 정보를 보면 디바이스 온보딩 상태에 대한 추가 정보를 볼 수 있습니다.

- AWS IoT 무선 API 또는 AWS CLI 사용

  디바이스 온보딩 상태를 보려면 다음 AWS IoT 무선 API 작업 또는 해당 AWS CLI 명령을 사용하세요.

  - **ListWirelessDeviceImportTasks** API 또는 **list-wireless-device-import-tasks** CLI

    이 API 태스크는 AWS IoT 무선 계정에 추가된 모든 가져오기 태스크 및 상태에 대한 정보를 반환합니다. 또한 이러한 작업에서 Sidewalk 디바이스의 온보딩 상태를 요약한 개수도 반환합니다.

  - **ListDevicesForWirelessDeviceImportTask** API 또는 **list-devices-for-wireless-device-import-task** CLI

    이 API 작업은 지정된 가져오기 작업 및 상태에 대한 정보, 가져오기 작업에 추가된 모든 Sidewalk 디바이스에 대한 정보와 온보딩 상태 정보를 반환합니다.

  - **GetWirelessDeviceImportTask** API 또는 **get-wireless-device-import-task** CLI

    이 API 작업은 지정된 가져오기 작업 및 상태에 대한 정보와 해당 작업에 포함된 Sidewalk 디바이스의 온보딩 상태 요약 수를 반환합니다.

가져오기 작업 상태

AWS 계정에 생성한 가져오기 작업에 대해 다음 상태 메시지 중 하나가 표시될 수 있습니다. 상태는 가져오기 작업이 처리를 시작했는지, 완료되었는지 또는 실패했는지를 나타냅니다. AWS IoT 콘솔이나 AWS IoT 무선 API 작업의 `StatusReason` 파라미터를 사용하여 추가적인 상태 세부 정보를 검색할 수도 있습니다.

- 초기화 중

  Amazon Sidewalk용 AWS IoT Core가 무선 디바이스 가져오기 태스크 요청을 받았으며 태스크를 설정하는 중입니다.

- 초기화됨

Amazon Sidewalk용 AWS IoT Core가 가져오기 태스크 설정을 완료했으며 일련번호(SMSN)를 사용하여 디바이스를 가져오고 태스크를 계속 처리할 수 있도록 제어 로그가 도착하기를 기다리고 있습니다.

- PENDING

  가져오기 태스크가 대기열에서 처리되기를 기다리고 있습니다. Amazon Sidewalk용 AWS IoT Core가 처리 대기열에 있는 다른 태스크를 평가하고 있습니다.

- 완료

  가져오기 작업이 처리되고 완료되었습니다.

- 실패

  가져오기 작업 또는 디바이스 작업이 실패했습니다. `StatusReason` 파라미터를 사용하여 가져오기 작업이 실패한 이유(예: 검증 예외)를 식별할 수 있습니다.

- 삭제 중

  가져오기 작업이 삭제 대상으로 표시되었으며 삭제가 진행되고 있습니다.

디바이스 온보딩 상태

가져오기 작업에 추가한 Sidewalk 디바이스에 대해 다음 중 하나의 상태 메시지가 표시될 수 있습니다. 상태는 디바이스가 온보딩될 준비가 되었는지, 온보딩되었는지 또는 온보딩에 실패했는지를 나타냅니다. AWS IoT 콘솔이나 AWS IoT 무선 API 작업의 `OnboardingStatusReason` 파라미터인 `ListDevicesForWirelessDeviceImportTask`를 사용하여 추가적인 상태 세부 정보를 검색할 수도 있습니다.

- 초기화됨

  Amazon Sidewalk용 AWS IoT Core가 가져오기 태스크 설정을 완료했으며 일련번호(SMSN)를 사용하여 디바이스를 가져오고 태스크를 계속 처리할 수 있도록 제어 로그가 도착하기를 기다리고 있습니다.

- PENDING

  가져오기 태스크를 대기열에서 처리하고 태스크에 디바이스를 온보딩하기를 기다리고 있습니다. Amazon Sidewalk용 AWS IoT Core가 처리 대기열에 있는 다른 태스크를 평가하고 있습니다.

- 온보딩됨

Sidewalk 디바이스가 가져오기 작업에 성공적으로 온보딩되었습니다.

- FAILED

  가져오기 작업 또는 디바이스 작업이 실패했고 Sidewalk 디바이스가 작업에 온보딩되지 못했습니다. `OnboardingStatusReason` 파라미터를 사용하여 디바이스 온보딩이 실패한 이유에 대한 추가 세부 정보를 검색할 수 있습니다.

AWS IoT Wireless의 보안

AWS에서는 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 매우 민감한 조직의 요구 사항에 부합하도록 빌드가 된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS(와)과 귀하의 공동 책임입니다. 공동 책임 모델은(는) 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 - AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. AWS IoT Wireless에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램의 범위에 속하는 AWS 서비스를 참조하세요.
- 클라우드의 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS IoT Wireless 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 보안 및 규정 준수 목표에 맞게 AWS IoT Wireless를 구성하는 방법을 보여 줍니다. 또한 AWS IoT Wireless 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배우게 됩니다.

내용

AWS IoT Wireless의 데이터 보호

AWS 공동 책임 모델은 AWS IoT Wireless의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 이 인프라에서 호스팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 사용하는 AWS 서비스의 보안

구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하십시오. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조합니다.

데이터를 보호하려면 AWS 계정 보안 인증을 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)을 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이 방식을 사용하면 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS리소스와 통신합니다. TLS 1.2가 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail(으)로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2를 참조합니다.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 AWS IoT Wireless 또는 기타 AWS 서비스 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함시켜서는 안 됩니다.

AWS IoT 무선에서 데이터 암호화

전송 중 및 저장 중인 AWS IoT 무선 데이터는 기본적으로 모두 암호화됩니다. AWS IoT 무선는 AWS KMS key의 고객 관리형 AWS KMS 키를 지원하지 않습니다. 데이터를 암호화할 때 AWS IoT 무선는 AWS 소유 키만 사용합니다.

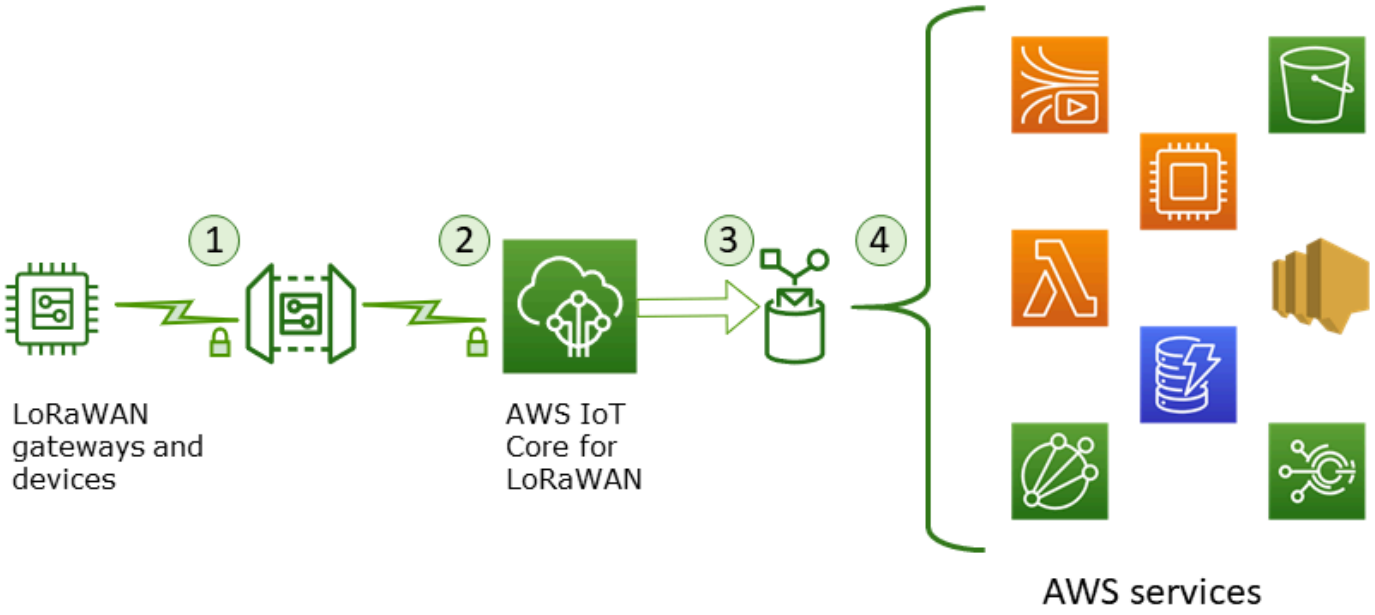
AWS IoT Core for LoRaWAN을 통한 데이터 및 전송 보안

AWS IoT Core for LoRaWAN은 다음 방법을 사용하여 LoRaWAN 디바이스, 게이트웨이 및 AWS IoT Core for LoRaWAN 간의 데이터 및 통신을 보호합니다.

- 디바이스가 LoRaWAN 게이트웨이와 통신할 때 따르는 보안 모범 사례는 LoRaWAN 보안 백서에 설명되어 있습니다.
- AWS IoT Core가 게이트웨이를 AWS IoT Core for LoRaWAN에 연결하고 데이터를 다른 AWS 서비스로 전송하는 데 사용하는 보안. 자세한 내용은 AWS IoT Core의 데이터 보호를 참조하세요.

시스템 전체에서 데이터를 보호하는 방법

이 다이어그램에서 AWS IoT Core for LoRaWAN에 연결된 LoRaWAN 시스템의 핵심 요소를 식별하여 전체 데이터가 어떻게 보호되는지 확인할 수 있습니다.



1. LoRaWAN 무선 디바이스는 전송하기 전에 AES128 CTR 모드를 사용하여 이진 메시지를 암호화합니다.
2. AWS IoT Core for LoRaWAN에 대한 게이트웨이 연결은 AWS IoT의 전송 보안에 설명된 대로 TLS로 보호됩니다. AWS IoT Core for LoRaWAN은 이진 메시지를 해독하고 해독된 이진 메시지 페이로드를 base64 문자열로 인코딩합니다.
3. 이에 따라 base64로 인코딩된 메시지는 디바이스에 할당된 대상에서 설명되어 있는 AWS IoT 규칙에 대한 메시지 페이로드로서 전송됩니다. AWS 내의 데이터는 AWS 소유 키를 사용하여 암호화됩니다.
4. AWS IoT 규칙은 메시지 데이터를 규칙 구성에 설명된 서비스로 전송합니다. AWS 내의 데이터는 AWS 소유 키를 사용하여 암호화됩니다.

LoRaWAN 디바이스 및 게이트웨이 전송 보안

LoRaWAN 디바이스 및 AWS IoT Core for LoRaWAN은 미리 공유된 루트 키를 저장합니다. 세션 키는 프로토콜에 따라 LoRaWAN 디바이스와 AWS IoT Core for LoRaWAN 모두에서 파생됩니다. 대칭 세션 키는 표준 AES-128 CTR 모드에서 암호화 및 복호화에 사용됩니다. 또한 4바이트 메시지 무결성 코드(MIC)는 표준 AES-128 CMAC 알고리즘에 따라 데이터 무결성을 검사하는 데 사용됩니다. 세션 키는 조인/리조인 프로세스를 사용하여 업데이트할 수 있습니다.

LoRa 게이트웨이에 대한 보안 사례는 LoRaWAN 사양에 설명되어 있습니다. LoRa 게이트웨이는 `Basics Station`을 사용하여 웹 소켓을 통해 AWS IoT Core for LoRaWAN에 연결합니다. `Basics Station`은 버전 2.0.4 이상만 지원합니다.

웹 소켓 연결이 설정되기 전에 AWS IoT Core for LoRaWAN은 TLS 서버 및 클라이언트 인증 모드를 사용하여 게이트웨이를 인증합니다. LoRaWAN 프로토콜의 기밀성을 보장하기 위해 TLS 버전 1.2가 사용됩니다. TLS 지원은 다수의 프로그래밍 언어 및 운영 체제를 지원합니다. AWS의 데이터는 특정 AWS 서비스에 의해 암호화됩니다. 다른 AWS 서비스의 데이터 암호화에 대한 자세한 내용은 해당 서비스의 보안 설명서를 참조하세요.

AWS IoT Core for LoRaWAN은 TLS 인증에 사용되는 인증서와 키를 구성 및 업데이트하는 구성 및 업데이트 서버(CUPS)를 유지 관리하기도 합니다.

AWS IoT Wireless를 위한 자격 증명 및 액세스 관리

AWS Identity and Access Management(IAM)은 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 AWS IoT Wireless 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 소유)를 받을 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

- 고객
- 보안 인증을 통한 인증
- 정책을 사용한 액세스 관리
- AWS IoT Wireless가 IAM과 함께 작동하는 방식
- AWS IoT Wireless 자격 증명 기반 정책 예제
- AWS IoT 무선의 AWS 관리형 정책
- AWS IoT Wireless 자격 증명 및 액세스 문제 해결

고객

AWS Identity and Access Management(IAM)를 사용하는 방법은 AWS IoT Wireless에서 수행하는 작업에 따라 달라집니다.

서비스 사용자 - AWS IoT Wireless 서비스를 사용하여 작업을 수행하는 경우 필요한 보안 인증 정보와 권한을 관리자가 제공합니다. 더 많은 AWS IoT Wireless 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AWS IoT Wireless의 기능에 액세스할 수 없는 경우 AWS IoT Wireless 자격 증명 및 액세스 문제 해결 단원을 참조하세요.

서비스 관리자 - 회사에서 AWS IoT Wireless 리소스를 책임지고 있는 경우 AWS IoT Wireless에 대한 전체 액세스를 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 AWS IoT Wireless 기능과 리소스를 결정합니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사가 AWS IoT Wireless에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 AWS IoT Wireless가 IAM과 함께 작동하는 방식 단원을 참조하세요.

IAM 관리자 - IAM 관리자라면 AWS IoT Wireless 액세스 권한 관리에 대한 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AWS IoT Wireless 자격 증명 기반 정책 예제를 보려면 AWS IoT Wireless 자격 증명 기반 정책 예제 단원을 참조하세요.

보안 인증을 통한 인증

인증은 ID 보안 인증을 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자나 IAM 사용자로 또는 IAM 역할을 수임하여 인증(AWS에 로그인)되어야 합니다.

보안 인증 소스를 통해 제공된 보안 인증 정보를 사용하여 페더레이션형 ID로 AWS에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증, Google 또는 Facebook 보안 인증이 페더레이션형 ID의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS에 액세스하면 간접적으로 역할을 수임합니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. AWS에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인사용 설명서의 AWS 계정에 로그인하는 방법을 참조하세요.

AWS에 프로그래밍 방식으로 액세스하는 경우, AWS에서는 보안 인증 정보를 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK) 및 명령줄 인터페이스(CLI)를 제공합니다. AWS

도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 서명을 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS은(는) 다중 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center사용 설명서의 다중 인증 및 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하세요.

AWS 계정 루트 사용자

AWS 계정를 생성할 때는 해당 계정의 모든 AWS 서비스 및 리소스에 대해 완전한 액세스 권한이 있는 단일 로그인 ID로 시작합니다. 이 보안 인증은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에는 루트 사용자를 가급적 사용하지 않는 것이 좋습니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 이 정보를 사용합니다. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 Tasks that require root user credentials를 참조하십시오.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가지고 있는 AWS 계정 내 보안 인증입니다. 가능하다면 암호 및 액세스 키와 같은 장기 보안 인증 정보가 있는 IAM 사용자를 생성하는 대신, 임시 보안 인증 정보를 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증 정보가 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체 섹션을 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 보안 인증입니다. 귀하는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증을 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 정보는 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님) 섹션을 참조하세요.

IAM 역할

> **Note**
>
> AWS IoT 무선는 서비스 역할 및 서비스 연결 역할을 지원하지 않습니다.

IAM 역할은 특정 권한을 가지고 있는 AWS 계정계정 내 ID입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 전환하여 AWS Management Console에서 IAM 역할을 임시로 수임할 수 있습니다. AWS CLI 또는 AWS API 태스크를 직접적으로 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM 역할 사용를 참조하세요.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션형 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션형 ID가 인증되면 이 ID는 역할과 연결되며 역할에 의해 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 서드 파티 자격 증명 공급자의 역할 만들기 부분을 참조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 보안 인증 정보에서 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 집합을 IAM의 역할과 연결합니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트 섹션을 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스를 사용하면 역할을(프록시로 사용하는 대신) 리소스에 정책을 직접 연결할 수 있습니다. 교차 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이 섹션을 참조하세요.
- 교차 서비스 액세스 – 일부 AWS 서비스은(는) 다른 AWS 서비스의 기능을 사용합니다. 예를 들어 서비스에서 직접적으로 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 보안 주체의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하십시오.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.

- 서비스 연결 역할 – 서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 링크 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수는 없습니다.
- Amazon EC2에서 실행 중인 애플리케이션 – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 보안 인증을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여 섹션을 참조하세요.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하세요.

정책을 사용한 액세스 관리

정책을 생성하고 AWSID 또는 리소스에 연결하여 AWS내 액세스를 제어합니다. 정책은 ID 또는 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS은(는) 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되는지 또는 거부되는지를 결정합니다. 대부분의 정책은 AWS에 JSON 설명서로서 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

관리자는 AWSJSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole`태스크를 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console, AWS CLI또는 AWSAPI에서 역할 정보를 가져올 수 있습니다.

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지

를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 AWS 계정에 속한 다수의 사용자, 그룹 및 역할에 독립적으로 추가할 수 있는 정책입니다. 관리형 정책에는 AWS관리형 정책과 고객 관리형 정책이 포함되어 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 제어할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스이(가) 포함될 수 있습니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3, AWS WAF및 Amazon VPC는 ACL을 지원하는 대표적인 서비스입니다. ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 유형

AWS은(는) 비교적 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 유형은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 ID 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 ID 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책

중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계 섹션을 참조하세요.

- 서비스 제어 정책(SCP) – SCP는 AWS Organizations에서 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations은(는) 기업이 소유하는 여러 개의 AWS 계정을(를) 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 계정 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자을 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 정보는 AWS Organizations사용 설명서의 SCP 작동 방식을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 ID 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지 여부를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하십시오.

AWS IoT Wireless가 IAM과 함께 작동하는 방식

IAM을 사용하여 AWS IoT Wireless에 대한 액세스를 관리하려면 먼저 어떤 IAM 기능을 AWS IoT Wireless에 사용할 수 있는지를 이해해야 합니다. AWS IoT Wireless 및 기타 AWS 서비스에서 IAM을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서에서 IAM으로 작업하는 AWS 서비스를 참조하세요.

AWS IoT 무선을 통해 사용할 수 있는 IAM 기능

IAM 특성	AWS IoT 무선 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예

IAM 특성	AWS IoT 무선 지원
정책 조건 키	예
ACL	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
보안 주체 권한	예
서비스 역할	아니요
서비스 링크 역할	아니요

주제

- AWS IoT Wireless 자격 증명 기반 정책
- AWS IoT 무선 내 리소스 기반 정책
- 정책 작업
- 정책 리소스
- 조건 키
- 액세스 제어 목록(ACL)
- AWS IoT 무선를 사용한 ABAC
- AWS IoT 무선에서 임시 보안 인증 정보 사용
- AWS IoT 무선의 서비스 간 보안 주체 권한
- 서비스 역할
- AWS IoT 무선에 대한 서비스 연결 역할

AWS IoT Wireless 자격 증명 기반 정책

자격 증명 기반 정책 지원	예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 자격 증명 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

예제

AWS IoT Wireless 자격 증명 기반 정책의 예제를 보려면 AWS IoT Wireless 자격 증명 기반 정책 예제 단원을 참조하세요.

AWS IoT 무선 내 리소스 기반 정책

리소스 기반 정책 지원	아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 제어할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스이(가) 포함될 수 있습니다.

크로스 계정 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 신뢰 관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 보안 주체와 리소스가 서로 다른 AWS 계정에 있는 경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 엔터티(사용자 또는 역할)에도 리소스 액세스 권한을 부여해야 합니다. 엔터티에 자격 증명 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

정책 작업

정책 작업 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다. 일반적으로 정책 작업의 이름은 연결된 AWS API 작업의 이름과 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함합니다.

AWS IoT Wireless의 정책 작업은 작업 앞에 `iotwireless:` 접두사를 사용합니다. 예를 들어, `ListWirelessDevices` API를 사용하여 AWS 계정에 등록된 모든 무선 디바이스를 나열할 수 있는 권한을 부여하려면 `iotwireless:ListWirelessDevices` 작업을 정책에 포함합니다. 정책 문에는 `Action` 또는 `NotAction` 요소가 포함되어야 합니다. AWS IoT Wireless는 이 서비스로 수행할 수 있는 태스크를 설명하는 고유한 작업 세트를 정의합니다.

명령문 하나에 여러 태스크를 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "iotwireless:ListMulticastGroups",
      "iotwireless:ListFuotaTasks"
   ]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `Get`(이)라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "iotwireless:Get*"
```

AWS IoT Wireless 작업 목록을 보려면 IAM 사용 설명서의 AWS IoT Wireless에서 정의한 작업을 참조하세요.

정책 리소스

정책 리소스 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 문장에는 Resource 또는 NotResource 요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 작업을 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

AWS IoT 무선 서비스에는 다음 ARN이 있습니다.

```
arn:${Partition}:iotwireless:${Region}:${Account}:${Resource}/${Resource-id}
```

ARN 형식에 대한 자세한 내용은 Amazon 리소스 이름(ARNs) 및 AWS 서비스 네임스페이스를 참조하세요.

예를 들어, 문에서 네트워크 분석기 구성인 NAConfig1을 지정하려면 다음 ARN을 사용합니다.

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:NetworkAnalyzerConfiguration/
NAConfig1"
```

특정 계정에 속하는 모든 FUOTA 작업을 지정하려면 와일드카드(*)를 사용합니다.

```
"Resource": "arn:aws:iotwireless:us-east-1:123456789012:FuotaTask/*"
```

리소스를 나열하기 위한 작업과 같은 일부 AWS IoT Wireless 작업은 특정 리소스에서 수행할 수 없습니다. 이러한 경우, 와일드카드(*)를 사용해야 합니다.

```
"Resource": "*"
```

다양한 AWS IoT 무선 API 작업에는 여러 리소스가 관여합니다. 예를 들어, AssociateWirelessDeviceWithThing은 무선 디바이스를 AWS IoT 사물과 연결하므로 IAM 사용자는 해당 디바이스와 IoT 사물을 사용할 권한이 있어야 합니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

```
"Resource": [
      "WirelessDevice",
```

```
        "thing"
```

AWS IoT Wireless 리소스 유형 및 해당 ARN의 목록을 보려면 IAM 사용 설명서의 AWS IoT Wireless에서 정의된 리소스를 참조하세요. 각 리소스의 ARN에 어떤 작업을 지정할 수 있는지 알아보려면 AWS IoT Wireless에서 정의된 작업을 참조하세요.

조건 키

서비스별 정책 조건 키 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 선택 사항입니다. 같음이나 미만 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우 AWS는 논리적 `AND` 태스크를 사용하여 평가합니다. 단일 조건 키의 여러 값을 지정하는 경우 AWS는 논리적 `OR` 태스크를 사용하여 조건을 평가합니다. 문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS은(는) 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

AWS IoT Wireless에서는 자체 조건 키 집합을 정의하고 일부 전역 조건 키 사용도 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요. AWS IoT Wireless 조건 키 목록을 보려면 IAM 사용 설명서의 AWS IoT Wireless의 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 AWS IoT Wireless에서 정의된 작업을 참조하세요.

액세스 제어 목록(ACL)

ACL 지원	아니요

ACL(액세스 제어 목록)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

AWS IoT 무선를 사용한 ABAC

ABAC 지원(정책의 태그)	예

ABAC(속성 기반 액세스 제어)는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연결할 수 있습니다. ABAC의 첫 번째 단계로 엔터티 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그를 기반으로 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC란 무엇입니까?를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

AWS IoT Wireless 리소스에 태그를 연결하거나 AWS IoT Wireless에 대한 요청에서 태그를 전달할 수 있습니다. 태그를 기반으로 액세스를 제어하려면 `YOUR-SERVICE-PREFIX:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다. AWS IoT Wireless 리소스 태깅에 대한 자세한 내용은 AWS IoT 무선 리소스에 태그 지정 단원을 참조하세요.

AWS IoT 무선에서 임시 보안 인증 정보 사용

임시 보안 인증 지원	예

일부 AWS 서비스은(는) 임시 보안 인증을 사용하여 로그인할 때 작동하지 않습니다. 임시 보안 인증으로 작동하는 AWS 서비스를 비롯한 추가 정보는 IAM 사용 설명서의 IAM을 사용하는 AWS 서비스을(를) 참조하세요.

사용자 이름과 암호를 제외한 다른 방법을 사용하여 AWS Management Console에 로그인하면 임시 보안 인증을 사용하는 것입니다. 예를 들어 회사의 Single Sign-On(SSO) 링크를 사용하여 AWS에 액세스하면 해당 프로세스에서 자동으로 임시 보안 인증을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 보안 인증을 수동으로 만들 수 있습니다 그런 다음 이러한 임시 보안 인증을 사용하여 AWS에 액세스할 수 있습니다. AWS에서는 장기 액세스 키를 사용하는 대신 임시 보안 인증을 동적으로 생성할 것을 권장합니다. 자세한 정보는 IAM의 임시 보안 보안 인증 섹션을 참조하세요.

AWS IoT 무선의 서비스 간 보안 주체 권한

전달 액세스 세션(FAS) 지원	예

IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 보안 주체의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하십시오.

서비스 역할

서비스 역할 지원	아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM role(IAM 역할)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.

AWS IoT 무선에 대한 서비스 연결 역할

서비스 연결 역할 지원	아니요

서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 링크 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

AWS IoT Wireless 자격 증명 기반 정책 예제

기본적으로 IAM 사용자 및 역할은 AWS IoT Wireless 리소스를 생성 또는 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS CLI또는 AWSAPI를 사용해 태스크를 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 특정 API 태스크를 수행할 수 있는 권한을 사용자와 역할에게 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 정책을 연결해야 합니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 JSON 탭에서 정책 생성을 참조하세요.

주제

- 정책 모범 사례
- AWS IoT Wireless 콘솔 사용
- 사용자가 자신이 권한을 볼 수 있도록 허용
- AWS IoT 무선 무선 디바이스 작업을 수행하는 데 필요한 권한

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS IoT Wireless 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. 자격 증명 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책으로 시작하고 최소 권한을 향해 나아가기 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. AWS 계정에서 사용할 수 있습니다. 사용 사례에 고유한 AWS 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- Apply least-privilege permissions(최소 권한 적용) – IAM 정책을 사용하여 권한을 설정하는 경우 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 least-privilege permissions(최소 권한)으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.

- Use conditions in IAM policies to further restrict access(IAM 정책의 조건을 사용하여 액세스 추가 제한) – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 생성할 수 있습니다. 특정 AWS 서비스(예: AWS CloudFormation)를 통해 사용되는 경우에만 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 검증하여 안전하고 기능적인 권한 보장 – IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 검증합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 권장 사항을 제공하여 안전하고 기능적인 정책을 생성하도록 돕습니다. 자세한 정보는 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하세요.
- Require multi-factor authentication(MFA) (다중 인증 필요) – AWS 계정 계정에 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 설정합니다. API 작업을 호출할 때 MFA를 요구하려면 정책에 MFA 조건을 추가합니다. 자세한 정보는 IAM 사용 설명서의 MFA 보호 API 액세스 구성을 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

AWS IoT Wireless 콘솔 사용

AWS IoT Wireless 콘솔에 액세스하려면 최소한의 권한 세트가 있어야 합니다. 이러한 권한은 AWS 계정에서 AWS IoT Wireless 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(IAM 사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

해당 엔터티가 AWS IoT Wireless 콘솔을 여전히 사용할 수 있도록 하려면 AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하십시오.

```
AWSIoTWirelessFullAccess
```

AWS CLI 또는 AWSAPI만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자가 자신이 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 보안 인증에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI나 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "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 무선 무선 디바이스 작업을 수행하는 데 필요한 권한

자격 증명 기반 정책의 조건을 사용하여 AWS IoT Wireless 작업에 대한 액세스를 제어할 수 있습니다. 이 예제에서는 디바이스 생성 및 관리를 허용하는 정책을 생성할 수 있는 방법을 보여 줍니다. 하지만 사물 태그 Owner에 해당 사용자의 사용자 이름 값이 있는 경우에만 권한이 부여됩니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

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

```
 "Statement": [{
    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
              "iotwireless:CreateWirelessDevice",
              "iotwireless:GetWirelessDevice",
              "iotwireless:ListWirelessDevices",
              "iotwireless:UpdateWirelessDevice",
              "iotwireless:DeleteWirelessDevice"
           ],
    "Resource": "*"
    }
 ]
}
```

정책에는 `CreateWirelessDevice`, `GetWirelessDevice`, `ListWirelessDevices`, `UpdateWirelessDevice` 및 `DeleteWirelessDevice` 작업을 사용할 수 있는 권한을 부여하는 문이 하나 있습니다. AWS IoT 무선는 이러한 메서드를 직접적으로 호출하여 무선 디바이스를 생성하고 관리합니다.

자격 증명 기반 정책에서 권한을 가질 보안 주체를 지정하지 않으므로 이 정책은 보안 주체 요소를 지정하지 않습니다. 정책을 사용자에게 연결할 경우 사용자는 암시적인 보안 주체입니다. IAM 역할에 권한 정책을 연결하면 역할의 신뢰 정책에서 식별된 보안 주체가 권한을 얻습니다.

AWS IoT 무선의 AWS 관리형 정책

사용자, 그룹 또는 역할에 권한을 추가할 때 정책을 직접 작성하는 것보다 AWS관리형 정책을 사용하는 것이 더욱 편리합니다. 팀에 필요한 권한만 제공하는 IAM 고객 관리형 정책을 생성하려면 시간과 전문 지식이 필요합니다. 빠르게 시작하려면 AWS 관리형 정책을 사용하면 됩니다. 이 정책은 일반적인 사용 사례를 다루며 사용자의 AWS 계정에서 사용할 수 있습니다. AWS 관리형 정책에 대한 자세한 내용은 IAM 사용 설명서에서 AWS 관리형 정책을 참조하세요.

AWS 서비스 유지 관리 및 AWS관리형 정책 업데이트입니다. AWS 관리형 정책에서는 권한을 변경할 수 없습니다. 서비스에서 때때로 추가 권한을 AWS 관리형 정책에 추가하여 새로운 기능을 지원합니다. 이 유형의 업데이트는 정책이 연결된 모든 보안 인증(사용자, 그룹 및 역할)에 적용됩니다. 서비스는 새로운 기능이 시작되거나 새 태스크를 사용할 수 있을 때 AWS 관리형 정책에 업데이트됩니다. 서비스는 AWS관리형 정책에서 권한을 제거하지 않기 때문에 정책 업데이트로 인해 기존 권한이 손상되지 않습니다.

또한 AWS는 여러 서비스의 직무에 대한 관리형 정책을 지원합니다. 예를 들어 ReadOnlyAccess라는 이름의 AWS 관리형 정책은 모든 AWS 서비스 및 리소스에 대한 읽기 전용 액세스 권한을 제공합니다. 서비스에서 새 기능을 시작하면 AWS가 새 작업 및 리소스에 대한 읽기 전용 권한을 추가합니다. 직무 정책의 목록과 설명은 IAM 사용 설명서의 직무에 관한 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AWSIoTWirelessDataAccess

`AWSIoTWirelessDataAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 연결된 자격 증명에 `SendDataToWirelessDevice` API를 사용하여 LoRaWAN 및 Sidewalk 디바이스에 데이터를 전송할 수 있는 권한을 부여합니다. AWS Management Console에서 이 정책을 보려면 AWSIoTWirelessDataAccess를 확인하세요.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `iotwireless` – AWS IoT 무선 데이터를 검색합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iotwireless:SendDataToWirelessDevice"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 관리형 정책: AWSIoTWirelessFullAccess

`AWSIoTWirelessFullAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 연결된 자격 증명에 모든 AWS IoT 무선 작업에 액세스할 수 있는 권한을 부여합니다. AWS Management Console에서 이 정책을 보려면 AWSIoTWirelessFullAccess를 확인하세요.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `iotwireless` – AWS IoT 무선 데이터를 검색하고 모든 AWS IoT 무선 작업을 수행합니다.

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

AWS 관리형 정책: AWSIoTWirelessFullPublishAccess

`AWSIoTWirelessFullPublishAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 연결된 자격 증명에 사용자를 대신하여 AWS IoT 규칙에 게시할 수 있는 제한된 권한을 부여합니다. AWS Management Console에서 이 정책을 보려면 AWSIoTWirelessFullPublishAccess를 확인하세요.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `iot` – 엔드포인트 URL을 가져와 AWS IoT 규칙 엔진에 게시하는 작업을 수행합니다.

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

AWS 관리형 정책: AWSIoTWirelessLogging

`AWSIoTWirelessLogging` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 연결된 자격 증명에 Amazon CloudWatch Logs 로그 그룹을 생성하고 해당 그룹에 로그를 스트리밍할 수 있는 권한을 부여합니다. 이 정책은 CloudWatch 로깅 역할에 연결됩니다. AWS Management Console에서 이 정책을 보려면 `AWSIoTWirelessLogging`를 확인하세요.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `logs` - CloudWatch 로그를 검색합니다. 또한 CloudWatch Logs 그룹을 생성하고 해당 그룹에 로그를 스트리밍할 수 있도록 허용합니다.

```
{
    "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 관리형 정책: AWSIoTWirelessReadOnlyAccess

`AWSIoTLogging` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 연결된 자격 증명에 AWS IoT 무선 작업에 읽기 전용으로 액세스할 수 있는 권한을 부여합니다. AWS Management Console에서 이 정책을 보려면 `AWSIoTWirelessReadOnlyAccess`를 확인하세요.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `logsGet` – AWS IoT 무선 `List` 및 API 작업을 수행합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iotwireless:List*",
                "iotwireless:Get*"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

AWS 관리형 정책: AWSIoTWirelessGatewayCertManager

`AWSIoTWirelessGatewayCertManager` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 연결된 자격 증명에 AWS IoT 인증서를 생성, 나열 및 설명할 수 있는 권한을 부여합니다. AWS Management Console에서 이 정책을 보려면 `AWSIoTWirelessGatewayCertManager`를 확인하세요.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `iot` – 인증서를 생성, 설명 및 나열하는 작업을 수행합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IoTWirelessGatewayCertManager",
            "Effect": "Allow",
            "Action": [
                "iot:CreateKeysAndCertificate",
                "iot:DescribeCertificate",
                "iot:ListCertificates"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 관리형 정책으로 AWS IoT 무선 업데이트

이 서비스가 이러한 변경 내용을 추적하기 시작한 이후부터 AWS IoT 무선의 AWS 관리형 정책 업데이트에 관한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 AWS IoT 무선 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경 사항	설명	날짜
AWS IoT 무선에서 변경 사항 추적 시작	AWS IoT 무선이(가) AWS 관리형 정책에 대한 변경 내용 추적을 시작했습니다.	2022년 5월 18일

AWS IoT Wireless 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 AWS IoT Wireless 및 IAM에서 작업할 때 발생할 수 있는 공통적인 문제를 진단하고 수정할 수 있습니다.

주제

- AWS IoT Wireless에서 작업을 수행할 권한이 없음
- 액세스 키를 보기를 원함
- 관리자로서 다른 사용자가 AWS IoT Wireless에 액세스할 수 있기를 원함
- AWS 계정 외부의 사람이 AWS IoT Wireless 리소스에 액세스할 수 있기를 원함

AWS IoT Wireless에서 작업을 수행할 권한이 없음

AWS Management Console에서 작업을 수행할 권한이 없다는 메시지가 나타나는 경우 관리자에게 문의하여 도움을 받아야 합니다. 관리자는 사용자 이름과 비밀번호를 제공한 사람입니다.

다음 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 *`WirelessDevice`*에 대한 세부 정보를 보려고 하지만 `YOUR-SERVICE-PREFIX:`*`GetWirelessDevice`* 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: YOUR-
SERVICE-PREFIX:GetWirelessDevice on resource: my-LoRaWAN-device
```

이 경우 Mateo는 *`my-LoRaWAN-device`* 작업을 사용하여 `YOUR-SERVICE-PREFIX:`*`GetWirelessDevice`* 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

액세스 키를 보기를 원함

IAM 사용자 액세스 키를 생성한 후에는 언제든지 액세스 키 ID를 볼 수 있습니다. 하지만 보안 액세스 키는 다시 볼 수 없습니다. 보안 액세스 키를 잃어버린 경우 새로운 액세스 키 페어를 생성해야 합니다.

액세스 키는 액세스 키 ID(예: `AKIAIOSFODNN7EXAMPLE`)와 보안 액세스 키(예: `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`)의 두 가지 부분으로 구성됩니다. 사용자 이름 및 암호와 같이 액세스 키 ID와 보안 액세스 키를 함께 사용하여 요청을 인증해야 합니다. 사용자 이름과 암호를 관리하는 것처럼 안전하게 액세스 키를 관리합니다.

> **Important**
>
> 정식 사용자 ID를 찾는 데 도움이 되더라도 액세스 키를 타사에 제공하지 마시기 바랍니다. 이로 인해 다른 사람에게 AWS 계정에 대한 영구 액세스를 제공하게 될 수 있습니다.

액세스 키 페어를 생성할 때는 액세스 키 ID와 보안 액세스 키를 안전한 위치에 저장하라는 메시지가 나타납니다. 보안 액세스 키는 생성할 때만 사용할 수 있습니다. 하지만 보안 액세스 키를 잃어버린 경우 새로운 액세스 키를 IAM 사용자에게 추가해야 합니다. 최대 두 개의 액세스 키를 가질 수 있습니다. 이미 두 개가 있는 경우 새로 생성하려면 먼저 키 페어 하나를 삭제해야 합니다. 지침을 보려면 IAM 사용 설명서의 액세스 키 관리를 참조하세요.

관리자로서 다른 사용자가 AWS IoT Wireless에 액세스할 수 있기를 원함

다른 사용자가 AWS IoT Wireless에 액세스하도록 하려면 액세스 권한이 필요한 사용자 또는 애플리케이션에 대한 IAM 엔터티(사용자 또는 역할)를 만들어야 합니다. 다른 사용자들은 해당 엔터티에 대한 보안 인증을 사용해 AWS에 액세스합니다. 그런 다음 AWS IoT Wireless에 대한 올바른 권한을 부여하는 정책을 엔터티에 연결해야 합니다.

바로 시작하려면 IAM 사용 설명서의 첫 번째 IAM 위임 사용자 및 그룹 생성을 참조하세요.

AWS 계정 외부의 사람이 AWS IoT Wireless 리소스에 액세스할 수 있기를 원함

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- AWS IoT Wireless에서 이러한 기능을 지원하는지 여부를 알아보려면 AWS IoT Wireless가 IAM과 함께 작동하는 방식 단원을 참조하세요.
- 소유하고 있는 AWS 계정의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공을 참조하십시오.
- 리소스에 대한 액세스 권한을 서드 파티 AWS 계정에게 제공하는 방법을 알아보려면 IAM 사용 설명서의 서드 파티가 소유한 AWS 계정에 대한 액세스 제공을 참조하세요.
- 자격 증명 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 페더레이션)을 참조하세요.
- 크로스 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하십시오.

AWS IoT Wireless의 규정 준수 검증

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 AWS IoT Wireless의 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램의 범위 내에 있는 AWS 서비스 목록은 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하세요. 일반적인 내용은 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact을(를) 사용하여 서드 파티 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

AWS IoT Wireless 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표와 관련 법률 및 규정에 따라 결정됩니다. AWS는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 – 이 배포 안내서에서는 아키텍처 고려 사항에 관해 설명하고 AWS에서 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- HIPAA 보안 및 규정 준수 아키텍팅 백서 - 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 생성하는 방법을 설명합니다.
- AWS 규정 준수 리소스 - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 안내서 콜렉션입니다.
- AWS Config 개발자 설명서의 규칙을 사용하여 리소스 평가 – AWS Config를 사용하여 리소스 구성이 내부 사례, 업계 지침, 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub: 이 AWS 서비스는 보안 산업 표준 및 모범 사례 규정 준수 여부를 확인하는 데 도움이 되도록 AWS 내 보안 상태를 종합적으로 보여줍니다.

AWS IoT Wireless의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며, 이러한 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 정보는 AWS 글로벌 인프라를 참조하세요.

AWS IoT Wireless의 인프라 보안

관리형 서비스로서 AWS IoT Wireless는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차에 의해 보호됩니다.

AWS에서 게시한 API 직접 호출을 사용하여 네트워크를 통해 AWS IoT Wireless에 액세스합니다. 클라이언트가 전송 계층 보안(TLS) 1.0 이상을 지원해야 합니다. TLS 1.2 이상을 권장합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 PFS(전달 완전 보안, Perfect Forward Secrecy)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 보안 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

Amazon CloudWatch Logs를 사용하여 AWS IoT 무선 리소스 모니터링

모니터링은 AWS IoT Wireless 및 기타 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. LoRaWAN 및 Sidewalk 디바이스 모두에 대한 모니터링을 사용할 수 있으며, AWS IoT 무선에 온보딩된 시점부터 정보성 메시지와 오류를 확인할 수 있습니다.

다중 지점 실패가 발생할 경우 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분에서 모니터링 데이터를 수집하는 것이 좋습니다. 먼저 다음 질문에 답하는 모니터링 계획을 수립합니다. 어떻게 답해야 할지 잘 모르는 경우에도 계속해서 로깅을 활성화하고 기준 성능을 설정할 수 있습니다.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

다음 단계에서는 다양한 시간과 다양한 부하 조건에서 성능을 측정하여 환경에서 로깅을 활성화하고 정상적인 AWS IoT Wireless 성능의 기준선을 설정합니다. AWS IoT Wireless을 모니터링할 때 과거 모니터링 데이터를 현재 성능 데이터와 비교할 수 있도록 유지합니다. 이를 통해 정상적인 성능 패턴과 성능 이상을 식별하고 문제 해결 방법을 고안할 수 있습니다.

모니터링 도구

AWS IoT Wireless를 관찰하고, 문제 발생 시 보고하고, 적절한 경우 자동 조치를 취하는 다음과 같은 모니터링 도구를 사용할 수 있습니다.

- Amazon CloudWatch는 AWS에서 실행하는 AWS 리소스와 애플리케이션을 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 맞춤 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임계값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 CloudWatch에서 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 추적하고 필요할 때 자동으로 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.
- 네트워크 분석기는 LoRaWAN 디바이스 및 게이트웨이를 포함한 LoRaWAN 리소스를 모니터링할 수 있습니다. 따라서 추적 메시지 수신을 시작하여 적시에 로그 정보를 받기 위해 연결을 설정하는

데 걸리는 시간이 줄어듭니다. 자세한 내용은 네트워크 분석기를 사용하여 무선 리소스 플릿 실시간 모니터링 단원을 참조하십시오.

Amazon CloudWatch를 사용하여 리소스를 모니터링하는 방법

원시 데이터를 수집하여 읽기 가능한 실시간에 가까운 지표로 처리하는 CloudWatch를 사용하여 AWS IoT Wireless를 모니터링할 수 있습니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임계값을 주시하다가 해당 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.

AWS IoT 무선 리소스를 로깅하고 모니터링하려면 다음 단계를 수행하세요.

1. 로깅 역할을 생성하여 AWS IoT 무선에 대한 로깅 역할 및 정책 생성에 설명된 대로 AWS IoT 무선 리소스를 로깅합니다.
2. CloudWatch Logs 콘솔의 로그 메시지는 기본 로그 수준이 `ERROR`이며, 이는 오류 정보만 포함하는 요약된 정보입니다. 더 상세한 메시지를 보려면 CLI를 사용하여 AWS IoT 무선 리소스에 대한 로깅 구성에 설명된 대로 로깅을 먼저 구성하는 것이 좋습니다.
3. 그런 다음 CloudWatch Logs 콘솔에서 로그 항목을 확인하여 리소스를 모니터링할 수 있습니다. 자세한 내용은 CloudWatch AWS IoT 무선 로그 항목 보기 단원을 참조하십시오.
4. 로그 그룹을 사용하여 필터 표현식을 만들 수 있지만, 먼저 간단한 필터를 생성하고 로그 그룹에서 로그 항목을 확인한 다음 CloudWatch Insights로 이동하여 모니터링 중인 리소스 또는 이벤트에 따라 로그 항목을 필터링하는 쿼리를 생성하는 것이 좋습니다. 자세한 내용은 CloudWatch 인사이트를 사용하여 AWS IoT 무선의 로그 필터링 단원을 참조하십시오.

AWS IoT 무선의 로깅 구성

AWS IoT 활동을 모니터링 및 로그하기 전에 먼저 CLI 또는 API를 사용하여 AWS IoT 무선 리소스에 대한 로깅을 활성화합니다.

AWS IoT 무선 로깅 구성 방법을 고려할 때 달리 지정하지 않는 한, 기본 로깅 구성에 따라 AWS IoT 활동이 기록되는 방식이 결정됩니다. 시작할 때 기본 로그 수준이 `INFO`인 자세한 로그를 얻고 싶을 수 있습니다.

초기 로그를 검토한 후 기본 로그 수준을 (요약된 정보인) `ERROR`로 변경하고, 더 많은 주의가 필요할 수 있는 리소스에는 더 상세한 리소스별 로그 수준을 설정할 수 있습니다. 로그 수준은 언제든 변경할 수 있습니다.

다음 주제에서는 AWS IoT 무선 리소스에 대한 로깅을 구성하는 방법을 보여줍니다.

주제

AWS IoT 무선에 대한 로깅 역할 및 정책 생성

다음은 AWS IoT 무선 리소스에 대해서만 로깅 역할을 생성하는 방법을 보여줍니다. AWS IoT Core에 대한 로깅 역할을 만들려는 경우 https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html을(를) 참조하세요.

AWS IoT 무선에 대한 로깅 역할 생성

로깅을 사용하려면 먼저 사용자를 대신하여 AWS IoT 무선 활동을 모니터링할 수 있는 권한을 AWS에 부여하는 IAM 역할과 정책을 생성해야 합니다.

로깅을 위한 IAM 역할 생성

AWS IoT 무선에 대한 로깅 역할을 생성하려면 IAM의 역할 허브(Roles hub of the IAM) 콘솔을 열고 역할 생성(Create role)을 선택합니다.

1. Select type of trusted entity(신뢰할 수 있는 엔터티 유형 선택) 아래에서 다른 Another AWS account (AWS 계정)를 선택합니다.
2. 계정 ID에서 AWS계정 ID를 입력한 후 다음: 권한을 선택합니다.
3. 검색 상자에 **`AWSIoTWirelessLogging`**을(를) 입력합니다.
4. AWSIoTWirelessLogging이라는 정책 옆에 있는 확인란을 선택한 후 다음: 태그를 선택합니다.
5. 다음: 검토를 선택합니다.
6. 역할 이름에 **`IoTWirelessLogsRole`**을 입력한 다음 역할 생성을 선택합니다.

IAM 역할의 신뢰 관계 편집

이전 단계를 실행한 후 표시되는 확인 메시지에서, 생성한 역할 이름인 IoTWirelessLogsRole을 선택합니다. 그런 다음 역할을 편집하여 다음 신뢰 관계를 추가합니다.

1. IoTWirelessLogsRole 역할의 요약 섹션에서 신뢰 관계 탭을 선택한 다음신뢰 관계 편집을 선택합니다.

2. 정책 문서에서 `Principal` 속성을 다음 예시처럼 변경합니다.

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

`Principal` 속성을 변경한 후 전체 정책 문서가 다음 예시와 같은 형식이어야 합니다.

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

3. 변경 사항을 저장하고 종료하려면 신뢰 정책 업데이트(Update Trust Policy)를 선택합니다.

AWS IoT 무선에 대한 로깅 정책

다음 정책 문서는 AWS IoT 무선이 사용자 대신 CloudWatch에 로그 항목을 제출하도록 허용하는 역할 정책 및 신뢰 정책을 제공합니다.

> **Note**
>
> 이 AWS 관리형 정책 문서는 로깅 역할인 IoTWirelessLogsRole을 생성할 때 자동으로 생성되었습니다.

역할 정책

다음은 역할 정책 문서를 보여줍니다.

```
{
```

```
    "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 무선 활동만 로깅하는 신뢰 정책

다음은 AWS IoT 무선 활동만 로깅하는 신뢰 정책을 보여줍니다.

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

AWS IoT Core 활동도 로그하는 IAM 역할을 생성한 경우 정책 문서를 사용하여 두 활동을 모두 로깅할 수 있습니다. AWS IoT Core에 대한 로깅 역할 생성에 대한 자세한 내용은 https://docs.aws.amazon.com/iot/latest/developerguide/create-logging-role.html을(를) 참조하세요.

다음 단계

AWS IoT 무선 리소스를 로깅할 로깅 역할을 생성하는 방법을 배웠습니다. 기본적으로 로그의 로그 수준은 ERROR이므로 오류 정보만 보려면 CloudWatch AWS IoT 무선 로그 항목 보기(으)로 이동해서 로그 항목 조회를 통해 무선 리소스를 모니터링할 수 있습니다.

로그 항목에서 자세한 정보를 보려는 경우 리소스 또는 다른 이벤트 유형에 대한 기본 로그 수준을 구성할 수 있습니다(예: 로그 수준을 INFO로 설정). 리소스 로깅 구성에 대한 자세한 내용은 AWS IoT 무선 리소스에 대한 로깅 구성을(를) 참조하세요.

AWS IoT 무선 리소스에 대한 로깅 구성

AWS IoT 무선 리소스에 대한 로깅을 구성하려면 API 또는 CLI를 사용합니다. AWS IoT 무선 리소스 모니터링을 시작할 때 기본 구성을 사용할 수 있습니다. 이렇게 하려면 이 주제를 건너뛰고 CloudWatch Logs를 사용한 AWS IoT 무선 모니터링(으)로 이동하여 로그를 모니터링할 수 있습니다.

로그 모니터링을 시작한 후 CLI를 사용하여 로그 수준을 더 상세한 옵션으로 변경할 수 있습니다(예: INFO 및 ERROR 정보를 제공하고 더 많은 리소스에 대한 로깅 활성화).

AWS IoT 무선 및 로그 수준

API 또는 CLI를 사용하기 전에 다음 표를 사용하여 로깅을 구성할 수 있는 다양한 로그 수준과 리소스에 대해 알아보세요. 이 표에서는 리소스를 모니터링할 때 CloudWatch Logs에 표시되는 파라미터를 보여줍니다. 리소스에 대한 로깅을 구성하는 방법은 콘솔에 표시되는 로그를 결정합니다.

샘플 CloudWatch Logs가 어떤 모습인지, 그리고 이러한 파라미터를 사용하여 AWS IoT 무선 리소스에 대한 유용한 정보를 로깅할 수 있는 방법에 대한 자세한 내용은 CloudWatch AWS IoT 무선 로그 항목 보기 섹션을 참조하세요.

로그 수준 및 리소스

명칭	가능한 값	설명
logLevel	INFO,ERROR 또는 DISABLED	• ERROR: 작업을 실패하게 만든 오류를 표시합니다. 로그에 ERROR 정보만 포함됩니다. • INFO: 사물 흐름에 대한 상위 수준 정보를 제공합니다. 로그에 INFO 및 ERROR 정보가 포함됩니다. • DISABLED: 모든 로깅을 비활성화합니다.

명칭	가능한 값	설명
resource	WirelessGateway 또는 WirelessDevice	리소스 유형(WirelessGateway 또는 WirelessDevice)입니다.
wirelessGatewayType	LoRaWAN	resource이(가) WirelessGateway 일 때 무선 게이트웨이의 유형(항상 LoRaWAN)입니다.
wirelessDeviceType	LoRaWAN 또는 Sidewalk	resource이(가) WirelessDevice 일 때 무선 디바이스의 유형(LoRaWAN 또는 Sidewalk)입니다.
wirelessGatewayId	-	resource이(가) WirelessGateway 일 때 무선 게이트웨이의 식별자입니다.
wirelessDeviceId	-	resource이(가) WirelessDevice 일 때 무선 디바이스의 식별자입니다.
event	Join, Rejoin, Registration , Uplink_data , Downlink_data , CUPS_Request 및 Certificate	로그하려는 리소스가 무선 디바이스인지 또는 무선 게이트웨이인지에 따라, 로그되는 이벤트의 유형입니다. 자세한 내용은 CloudWatch AWS IoT 무선 로그 항목 보기 단원을 참조하십시오.

AWS IoT 무선 로깅 API

다음 API 작업을 통해 리소스 로깅을 구성할 수 있습니다. 이 표에서는 API 작업을 사용하기 위해 생성해야 하는 샘플 IAM 정책도 보여 줍니다. 다음 섹션에서는 API를 사용하여 리소스의 로그 수준을 구성하는 방법에 대해 설명합니다.

API 작업 로깅

API 이름	설명	샘플 IAM 정책
GetLogLevelsByResourceTypes	현재 기본 로그 수준 또는 리소스 유형별 로그 수준을 반환합니다. 여기에는 무선 디바이스 또는 무선 게이	{ "Version": "2012-10-17",

API 이름	설명	샘플 IAM 정책
	트웨이에 대한 로그 옵션이 포함될 수 있습니다.	"Statement": [{ "Effect": "Allow", "Action": ["iotwireless:GetLogLevelsByResourceTypes"], "Resource": ["*"] }] }

API 이름	설명	샘플 IAM 정책

GetResourceLogLevel	지정된 리소스 식별자 및 리소스 유형에 대한 로그 수준 재정의를 반환합니다. 리소스는 무선 디바이스 또는 무선 게이트웨이일 수 있습니다.	{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Action": [
		"iotwireless:GetResourceLogLevel"
		],
		"Resource": [
		"arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc537-ab12-cd3a-d00e-1f0e20c1204a",
		]
		}
		]
		}

API 이름	설명	샘플 IAM 정책
PutResourceLogLevel	지정된 리소스 식별자 및 리소스 유형에 대한 로그 수준 재정의를 설정합니다. 리소스는 무선 게이트웨이 또는 무선 디바이스일 수 있습니다. **Note** 이 API는 계정당 200개의 로그 수준 재정의로 제한됩니다.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:PutResourceLogLevel" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc537-ab12-cd3a-d00e-1f0e20c1204a", ] } ] }`

API 이름	설명	샘플 IAM 정책
ResetAllResourceLogLevels	무선 게이트웨이와 무선 디바이스를 모두 포함하는 모든 리소스에 대한 로그 수준 재정의를 제거합니다. **Note** 이 API는 UpdateLogLevelsByResourceTypes API를 사용하여 설정된 로그 수준에 영향을 주지 않습니다	`{ "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 이름	설명	샘플 IAM 정책
ResetResourceLogLevel	지정된 리소스 식별자 및 리소스 유형에 대한 로그 수준 재정의를 제거합니다. 리소스는 무선 게이트웨이 또는 무선 디바이스일 수 있습니다.	{ "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 이름	설명	샘플 IAM 정책

UpdateLogLevelsByResourceTypes	기본 로그 수준 또는 리소스 유형별 로그 수준을 설정합니다. 무선 디바이스 또는 무선 게이트웨이에 대한 로그 옵션에 대해 이 API를 사용하고 CloudWatch에 표시될 로그 메시지를 제어할 수 있습니다.
	
	Note
	이벤트는 선택 사항이며 이벤트 유형은 리소스 유형에 연결됩니다. 자세한 내용은 이벤트 및 리소스 유형 단원을 참조하십시오.
		<pre>{
		    "Version": "2012-10-17",
		    "Statement": [
		        {
		            "Effect": "Allow",
		            "Action": [
		"iotwireless:UpdateLogLevelsByResourceTypes"
		            ],
		            "Resource": [
		                "*"
		            ]
		        }
		    ]
		}</pre>

CLI를 사용하여 리소스의 로그 수준 구성

이 섹션에서는 API 또는 AWS CLI를 사용하여 AWS IoT 무선 리소스에 대한 로그 수준을 구성하는 방법에 대해 설명합니다.

CLI를 사용하기 전에 다음을 수행하세요.

- 앞에서 설명한 대로 CLI 명령을 실행할 API에 대한 IAM 정책을 생성했는지 확인합니다.
- 사용하려는 역할의 Amazon 리소스 이름(ARN)이 필요합니다. 로깅에 사용할 역할을 만들어야 하는 경우 계속하기 전에 AWS IoT 무선에 대한 로깅 역할 및 정책 생성 단원을 참조하세요.

AWS CLI를 사용하는 이유

기본적으로 AWS IoT 무선에 대한 로깅 역할 및 정책 생성에 설명된 대로 IAM 역할 `IoTWirelessLogsRole`을 생성하는 경우 AWS Management Console에서 기본 로그 수준이 `ERROR`인 CloudWatch Logs가 표시됩니다. 모든 리소스 또는 특정 리소스에 대한 기본 로그 수준을 변경하려면 AWS IoT 무선 로깅 API 또는 CLI를 사용합니다.

AWS CLI 사용 방법

API 작업은 모든 리소스에 대해 로그 수준을 구성할지 또는 특정 리소스에 대한 로그 수준을 구성할지에 따라 다음 유형으로 분류할 수 있습니다.

- API 작업 `GetLogLevelsByResourceTypes` 및 `UpdateLogLevelsByResourceTypes`는 계정에서 특정 유형의 모든 리소스(예: 무선 게이트웨이, LoRaWAN 또는 Sidewalk 디바이스)에 대한 로그 수준을 검색하고 업데이트할 수 있습니다.
- API 작업 `GetResourceLogLevel`, `PutResourceLogLevel` 및 `ResetResourceLogLevel`은 리소스 식별자를 사용하여 지정한 개별 리소스의 로그 수준을 검색, 업데이트 및 재설정할 수 있습니다.
- API 작업 `ResetAllResourceLogLevels`는 `PutResourceLogLevel` API를 사용하여 로그 수준 재정의를 지정한 모든 리소스에 대해 로그 수준 재정의를 `null`로 재설정합니다.

CLI를 사용하여 AWS IoT에 대한 리소스별 로그인을 구성하려면

> **Note**
>
> 여기에 표시된 CLI 명령에 해당하는 AWS API의 메서드를 사용하여 API로 이 절차를 수행할 수도 있습니다.

1. 기본적으로 모든 리소스의 로그 수준은 `ERROR`로 설정됩니다. 계정의 모든 리소스에 대해 기본 로그 수준 또는 리소스 유형별 로그 수준을 설정하려면 update-log-levels-by-resource-types 명령을 사용합니다. 다음 예제에서는 JSON 파일 `Input.json`을 생성해서 CLI 명령에 대한 입력으로 제공하는 방법을 보여줍니다. 이 명령을 사용하여 로깅을 선택적으로 비활성화하거나 특정 유형의 리소스 및 이벤트에 대한 기본 로그 수준을 재정의할 수 있습니다.

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

```
}
```

여기서 각 항목은 다음과 같습니다.

WirelessDeviceLogOptions

무선 디바이스에 대한 로그 옵션 목록입니다. 각 로그 옵션에는 무선 디바이스 유형(Sidewalk 또는 LoRaWAN)과 무선 디바이스 이벤트 로그 옵션 목록이 포함됩니다. 각 무선 디바이스 이벤트 로그 옵션에는 선택적으로 이벤트 유형 및 해당 로그 수준이 포함될 수 있습니다.

WirelessGatewayLogOptions

무선 게이트웨이에 대한 로그 옵션 목록입니다. 각 로그 옵션에는 무선 게이트웨이 유형(LoRaWAN)과 무선 게이트웨이 이벤트 로그 옵션 목록이 포함됩니다. 각 무선 게이트웨이 이벤트 로그 옵션에는 선택적으로 이벤트 유형 및 해당 로그 수준이 포함될 수 있습니다.

DefaultLogLevel

모든 리소스에 사용할 로그 수준입니다. 유효 값은 `ERROR`, `INFO` 및 `DISABLED`입니다. 기본 값은 `INFO`입니다.

LogLevel

개별 리소스 유형 및 이벤트에 사용할 로그 수준입니다. 이러한 로그 수준은 기본 로그 수준(예: LoRaWAN 게이트웨이의 경우 로그 수준 `INFO`, 두 이벤트 유형의 경우 로그 수준 `DISABLED` 및 `ERROR`)을 재정의합니다.

다음 명령을 실행하여 `Input.json` 파일을 명령에 대한 입력으로 제공하세요. 이 명령은 출력을 생성하지 않습니다.

```
aws iotwireless update-log-levels-by-resource-types \
    --cli-input-json Input.json
```

무선 디바이스 및 무선 게이트웨이에 대한 로그 옵션을 제거하려면 다음 명령을 실행합니다.

```
{
    "DefaultLogLevel":"DISABLED",
    "WirelessDeviceLogOptions": [],
    "WireslessGatewayLogOptions":[]
}
```

2. update-log-levels-by-resource-types 명령은 출력을 반환하지 않습니다. get-log-levels-by-resource-types 명령을 사용하여 리소스별 로깅 정보를 검색할 수 있습니다. 이 명령은 기본 로그 수준과 무선 디바이스 및 무선 게이트웨이 로그 옵션을 반환합니다.

> **Note**
>
> get-log-levels-by-resource-types 명령은 CloudWatch 콘솔에서 로그 수준을 직접 검색할 수 없습니다. get-log-levels-by-resource-types 명령을 사용하면 update-log-levels-by-resource-types 명령을 통해 리소스에 대해 지정한 최신 로그 수준 정보를 가져올 수 있습니다.

```
aws iotwireless get-log-levels-by-resource-types
```

다음 명령을 실행하면 update-log-levels-by-resource-types로 지정한 최신 로깅 정보를 반환합니다. 예를 들어, 무선 디바이스 로그 옵션을 제거한 경우 get-log-levels-by-resource-types를 실행하면 이 값이 null로 반환됩니다.

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions": null,
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
            }
          ]
        }
      ]
}
```

3. 개별 무선 게이트웨이 또는 무선 디바이스 리소스에 대한 로그 수준을 제어하려면 다음 CLI 명령을 사용합니다.

   - put-resource-log-level
   - get-resource-log-level
   - reset-resource-log-level

   이러한 CLI를 사용하는 경우를 예로 들어 계정에서 많은 수의 무선 디바이스 또는 게이트웨이가 로그되고 있다고 가정해 보겠습니다. 일부 무선 디바이스에 대해서만 오류를 해결하려면 `DefaultLogLevel`을 `DISABLED`로 설정하여 모든 무선 디바이스에 대한 로깅을 비활성화하고 put-resource-log-level을 사용하여 계정에 있는 해당 디바이스에 대해서만 `LogLevel`을 `ERROR`로 설정할 수 있습니다.

   ```
   aws iotwireless put-resource-log-level \
       --resource-identifier
       --resource-type WirelessDevice
       --log-level ERROR
   ```

   이 예에서 명령은 지정된 무선 디바이스 리소스에 대해서만 로그 수준을 `ERROR`로 설정하고 다른 모든 리소스에 대한 로그는 비활성화됩니다. 이 명령은 출력을 생성하지 않습니다. 이 정보를 검색하고 로그 수준이 설정되었는지 확인하려면 get-resource-log-level 명령을 사용하세요.

4. 이전 단계에서 문제를 디버깅하고 오류를 해결한 후 reset-resource-log-level 명령을 실행하여 해당 리소스의 로그 수준을 `null`로 재설정할 수 있습니다. `put-resource-log-level` 명령을 사용하여 여러 디바이스에 대한 오류 문제 해결과 같이 둘 이상의 무선 디바이스 또는 게이트웨이 리소스에 대한 로그 수준 재정의를 설정한 경우 reset-all-resource-log-levels 명령을 사용하여 해당 모든 리소스에 대해 로그 수준 재정의를 다시 `null`로 재설정할 수 있습니다.

   ```
   aws iotwireless reset-all-resource-log-levels
   ```

   이 명령은 출력을 생성하지 않습니다. 리소스에 대한 로깅 정보를 검색하려면 get-resource-log-level 명령을 실행합니다.

다음 단계

로깅 역할을 생성하고 AWS IoT 무선 API를 사용하여 AWS IoT Core for LoRaWAN 리소스에 대한 로깅을 구성하는 방법을 배웠습니다. 그런 다음 로그 항목 모니터링에 대해 알아보려면 CloudWatch Logs를 사용한 AWS IoT 무선 모니터링(으)로 이동하세요.

CloudWatch Logs를 사용한 AWS IoT 무선 모니터링

AWS IoT Core for LoRaWAN에서 기본적으로 활성화된 CloudWatch 로그 항목은 50개 이상입니다. 각 로그 항목은 이벤트 유형, 로그 수준 및 리소스 유형을 설명합니다. 자세한 내용은 AWS IoT 무선 및 로그 수준 단원을 참조하십시오.

AWS IoT 무선 리소스 모니터링 방법

AWS IoT 무선에 대해 로깅이 활성화되면 AWS IoT 무선은 각 메시지가 디바이스에서 AWS IoT를 통해 전달될 때 각 메시지에 대한 진행 이벤트를 전송합니다. 기본적으로 AWS IoT 무선 로그 항목의 기본 로그 수준은 오류(error)입니다. AWS IoT 무선에 대한 로깅 역할 및 정책 생성에 설명된 대로 로깅을 활성화하면 CloudWatch 콘솔에 기본 로그 수준이 `ERROR`인 메시지가 표시됩니다. 이 로그 수준을 사용하면 사용 중인 모든 무선 디바이스 및 게이트웨이 리소스에 대한 오류 정보만 메시지에 표시됩니다.

로그에서 로그 수준 `INFO` 등으로 추가 정보를 표시하게 하거나 일부 디바이스에 대한 로그를 비활성화하고 일부 디바이스에 대해서만 로그 메시지를 표시하려는 경우 AWS IoT 무선 로깅 API를 사용할 수 있습니다. 자세한 내용은 CLI를 사용하여 리소스의 로그 수준 구성 단원을 참조하십시오.

필터 표현식을 만들어 필요한 메시지만 표시할 수도 있습니다.

콘솔에서 AWS IoT 무선 로그를 보기 전에

/aws/iotwireless 로그 그룹이 CloudWatch 콘솔에 나타나도록 하려면 다음을 수행해야 합니다.

- AWS IoT 무선에서 로깅 가능. AWS IoT 무선에서 로깅을 활성화하는 방법에 대한 자세한 내용은 AWS IoT 무선의 로깅 구성 섹션을 참조하세요.
- AWS IoT 무선 작업을 수행하여 일부 로그 항목 작성.

필터 표현식을 보다 효과적으로 생성하고 사용하려면 다음 항목에 설명된 대로 CloudWatch 인사이트를 사용해 보는 것이 좋습니다. 또한 여기에 제시된 순서대로 주제를 따르는 것이 좋습니다. 그러면 먼저 CloudWatch Log 그룹을 사용하여 콘솔에서 로그 항목을 보는 데 사용할 수 있는 다양한 리소스 유형, 해당 이벤트 유형 및 로그 수준에 대해 알아볼 수 있습니다. 그런 다음 CloudWatch Insights를 사용

하여 리소스에서 더 많은 유용한 정보를 얻는 방법으로 필터 표현식을 만드는 방법을 배울 수 있습니다.

주제

- CloudWatch AWS IoT 무선 로그 항목 보기
- CloudWatch 인사이트를 사용하여 AWS IoT 무선의 로그 필터링

CloudWatch AWS IoT 무선 로그 항목 보기

AWS IoT 무선에 대한 로깅 역할 및 정책 생성에 설명된 대로 AWS IoT 무선에 대한 로깅을 구성하고 일부 로그 항목을 작성한 후 다음 단계를 수행하여 CloudWatch 콘솔에서 로그 항목을 볼 수 있습니다.

CloudWatch Logs 그룹에서 AWS IoT 로그 보기

CloudWatch 콘솔에서 CloudWatch Logs는 /aws/iotwireless라는 로그 그룹에 나타납니다. CloudWatch Logs에 대한 자세한 내용은 CloudWatch Logs를 참조하세요.

CloudWatch 콘솔에서 AWS IoT 로그를 보려면

CloudWatch 콘솔로 이동하여 탐색 창에서 로그 그룹을 선택합니다.

1. 필터 텍스트 상자에 **/aws/iotwireless**를 입력한 다음 /aws/iotwireless 로그 그룹을 선택합니다.
2. 계정에 대해 생성된 AWS IoT Core for LoRaWAN 로그의 전체 목록을 보려면 모두 검색(Search all)을 선택합니다. 개별 로그 스트림을 보려면 확장 아이콘을 선택합니다.
3. 로그 스트림을 필터링하기 위해 이벤트 필터링 텍스트 상자에 쿼리를 입력할 수도 있습니다. 몇 가지 시도해볼 만한 쿼리는 다음과 같습니다.

   - `{ $.logLevel = "ERROR" }`

     이 필터를 사용하여 로그 수준이 ERROR인 모든 로그를 찾고 개별 오류 스트림을 확장하여 오류 메시지를 읽을 수 있습니다. 그러면 오류를 해결하는 데 도움이 됩니다.

   - `{ $.resource = "WirelessGateway" }`

     로그 수준에 관계없이 WirelessGateway 리소스에 대한 모든 로그를 찾습니다.

   - `{ $.event = "CUPS_Request" && $.logLevel = "ERROR" }`

     이벤트 유형이 CUPS_Request이고, 로그 수준이 ERROR인 로그를 모두 찾습니다.

이벤트 및 리소스 유형

다음 표에서는 로그 항목을 볼 수 있는 다양한 유형의 이벤트를 보여 줍니다. 이벤트 유형은 리소스 유형이 무선 디바이스인지 또는 무선 게이트웨이인지에 따라 달라집니다. 리소스 및 이벤트 유형에 대해 기본 로그 수준을 사용하거나 각각에 대해 로그 수준을 지정하여 기본 로그 수준을 재정의할 수 있습니다.

사용된 리소스를 기반으로 하는 이벤트 유형

Resource	리소스 유형	이벤트 유형	
무선 게이트웨이	LoRaWAN	• CUPS_Request • 인증서	
무선 디바이스	LoRaWAN	• 조인 • 리조인 • Uplink_Data • Downlink_Data	
무선 디바이스	Sidewalk	• 등록 • Uplink_Data • Downlink_Data	

다음 주제에서는 이러한 이벤트 유형 및 무선 게이트웨이 및 무선 디바이스의 로그 항목에 대한 자세한 정보를 제공합니다.

주제

- 무선 게이트웨이 및 무선 디바이스 리소스에 대한 로그 항목

무선 게이트웨이 및 무선 디바이스 리소스에 대한 로그 항목

로깅을 활성화한 후에는 무선 게이트웨이 및 무선 디바이스에 대한 로그 항목을 볼 수 있습니다. 다음 섹션에서는 리소스 및 이벤트 유형에 따라 다양한 종류의 로그 항목에 대해 설명합니다.

무선 게이트웨이 로그 항목

이 섹션에서는 CloudWatch 콘솔에 표시되는 무선 게이트웨이 리소스에 대한 샘플 로그 항목 중 일부를 보여줍니다. 이러한 로그 메시지는 이벤트 유형이 `CUPS_Request` 또는 `Certificate`일 수 있으

며 리소스 수준 또는 이벤트 수준에서 INFO, ERROR, 또는 DISABLED의 로그 수준을 표시하도록 구성할 수 있습니다. 오류 정보만 보려면 로그 수준을 ERROR로 설정합니다. ERROR 로그 항목의 메시지에는 실패한 이유에 대한 정보가 포함됩니다.

무선 게이트웨이 리소스의 로그 항목은 다음 이벤트 유형에 따라 분류할 수 있습니다.

- CUPS_Request

  게이트웨이에서 실행되는 LoRa Basic Station은 주기적으로 CUPS(Configuration and Update Server)에 업데이트 요청을 전송합니다. 이 이벤트 유형의 경우 무선 게이트웨이 리소스에 대한 CLI를 구성할 때 로그 수준을 INFO로 설정하면 로그에 다음과 같은 내용이 표시됩니다.

  - 이벤트가 성공하면 logLevel이 INFO인 로그 메시지가 표시됩니다. 메시지에는 게이트웨이에 전송된 CUPS 응답에 대한 세부 정보와 게이트웨이 세부 정보가 포함됩니다. 다음 예제는 이 로그 항목을 보여 줍니다. 로그 항목의 logLevel 및 기타 필드에 대한 자세한 내용은 AWS IoT 무선 및 로그 수준 단원을 참조하세요.

    ```
    {
        "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,"
    }
    ```

  - 오류가 있는 경우 logLevel이 ERROR인 로그 항목이 표시되고 메시지에 오류 세부 정보가 포함됩니다. CUPS_Request 이벤트에 대해 오류가 발생할 수 있는 경우의 예로는 CUPS CRC 누락, AWS IoT Core for LoRaWAN에서 게이트웨이의 TC Uri 불일치, IoTWirelessGatewayCertManagerRole 누락 또는 무선 게이트웨이 레코드를 가져올 수 없음 등이 있습니다. 다음 예제는 CRC 누락 로그 항목을 보여 줍니다. 이 오류를 해결하려면 게이트웨이 설정을 확인하여 올바른 CUPS CRC를 입력했는지 확인하세요.

    ```
    {
        "timestamp": "2021-05-13T16:56:08.853Z",
        "resource": "WirelessGateway",
        "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
        "wirelessGatewayType": "LoRaWAN",
    ```

```
    "gatewayEui": "feffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "ERROR",
    "message": "The CUPS CRC is missing from the request. Check your gateway setup
 and enter the CUPS CRC,"
}
```

- 인증서

  이 로그 항목은 무선 게이트웨이가 AWS IoT에 대한 연결을 인증하기 위한 올바른 인증서를 제공했는지 여부를 확인하는 데 도움이 됩니다. 이 이벤트 유형의 경우 무선 게이트웨이 리소스에 대한 CLI를 구성할 때 로그 수준을 INFO로 설정하면 로그에 다음과 같은 내용이 표시됩니다.

  - 이벤트가 성공하면 `logLevel`이 `INFO`인 로그 메시지가 표시됩니다. 메시지에는 인증서 ID 및 무선 게이트웨이 식별자에 대한 세부 정보가 포함됩니다. 다음 예제는 이 로그 항목을 보여 줍니다. 로그 항목의 `logLevel` 및 기타 필드에 대한 자세한 내용은 AWS IoT 무선 및 로그 수준 단원을 참조하세요.

    ```
    {
        "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)"
    }
    ```

  - 오류가 있는 경우 `logLevel`이 `ERROR`인 로그 항목이 표시되고 메시지에 오류 세부 정보가 포함됩니다. `Certificate` 이벤트에 대해 오류가 발생할 수 있는 경우의 예로는 잘못된 인증서 ID, 무선 게이트웨이 식별자 또는 무선 게이트웨이 식별자와 인증서 ID 간의 불일치 등이 있습니다. 다음 예는 잘못된 무선 게이트웨이 식별자로 인한 `ERROR`를 보여줍니다. 오류를 해결하려면 게이트웨이 식별자를 확인합니다.

    ```
    {
        "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)"
}
```

무선 디바이스 로그 항목

이 섹션에서는 CloudWatch 콘솔에 표시되는 무선 디바이스 리소스에 대한 샘플 로그 항목 중 일부를 보여줍니다. 이러한 로그 메시지의 이벤트 유형은 LoRaWAN 또는 Sidewalk 디바이스를 사용하는지 여부에 따라 다릅니다. 각 무선 디바이스 리소스 또는 이벤트 유형은 `INFO`, `ERROR`, 또는 `DISABLED`의 로그 수준을 표시하도록 구성할 수 있습니다.

> **Note**
>
> 요청에 LoRaWAN 및 Sidewalk 무선 메타데이터가 동시에 포함되어서는 안 됩니다. 이 시나리오에서 `ERROR` 로그 항목을 방지하려면 LoRaWAN 또는 Sidewalk 무선 데이터를 지정하세요.

LoRaWAN 디바이스 로그 항목

LoRaWAN 무선 디바이스의 로그 항목은 다음 이벤트 유형에 따라 분류할 수 있습니다.

- **`Join` 및 `Rejoin`**

  LoRaWAN 디바이스를 추가하고 AWS IoT Core for LoRaWAN에 연결할 때 디바이스에서 업링크 데이터를 전송하기 전에 `activation` 또는 `join procedure`라는 프로세스를 완료해야 합니다. 자세한 내용은 AWS IoT Core for LoRaWAN에 무선 디바이스 추가 단원을 참조하십시오.

  이 이벤트 유형의 경우 무선 게이트웨이 리소스에 대한 CLI를 구성할 때 로그 수준을 `INFO`로 설정하면 로그에 다음과 같은 내용이 표시됩니다.

  - 이벤트가 성공하면 `logLevel`이 `INFO`인 로그 메시지가 표시됩니다. 메시지에는 조인 또는 리조인 요청의 상태에 대한 세부 정보가 포함됩니다. 다음 예제는 이 로그 항목을 보여 줍니다. 로그 항목의 `logLevel` 및 기타 필드에 대한 자세한 내용은 AWS IoT 무선 및 로그 수준 단원을 참조하세요.

    ```
    {
        "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"
}
```

- 오류가 있는 경우 `logLevel`이 `ERROR`인 로그 항목이 표시되고 메시지에 오류 세부 정보가 포함됩니다. `Join` 및 `Rejoin` 이벤트에 대해 오류가 발생할 수 있는 경우의 예로는 잘못된 LoRaWAN 리전 설정 또는 잘못된 MIC(메시지 무결성 코드) 검사 등이 있습니다. 다음 예제에서는 MIC 검사로 인한 조인 오류를 보여줍니다. 오류를 해결하려면 올바른 루트 키를 입력했는지 확인하세요.

```
{
    "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 및 Downlink_Data

  이벤트 유형 `Uplink_Data`는 페이로드가 LoRaWAN 또는 Sidewalk 디바이스에서 AWS IoT로 전송될 때 AWS IoT 무선가 생성하는 메시지에 사용됩니다. 이벤트 유형 `Downlink_Data`는 AWS IoT에서 무선 디바이스로 전송되는 다운링크 메시지와 관련된 메시지에 사용됩니다.

  이 이벤트 유형의 경우 무선 디바이스에 대한 CLI를 구성할 때 로그 수준을 `INFO`로 설정하면 로그에 다음과 같은 내용이 표시됩니다.

  - 이벤트가 성공하면 `logLevel`이 `INFO`인 로그 메시지가 표시됩니다. 메시지에는 전송된 업링크 또는 다운링크 메시지의 상태 및 무선 디바이스 식별자에 대한 세부 정보가 포함됩니다. 다음은 Sidewalk 디바이스에 대한 이 로그 항목의 예를 보여 줍니다. 로그 항목의 `logLevel` 및 기타 필드에 대한 자세한 내용은 AWS IoT 무선 및 로그 수준 단원을 참조하세요.

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

- 오류가 있는 경우 `logLevel`이 `ERROR`인 로그 항목이 표시되고 메시지에 오류 세부 정보가 포함되어 오류를 해결하는 데 도움이 됩니다. `Registration` 이벤트에 대해 오류가 발생할 수 있는 경우의 예로는 인증 문제, 유효하지 않거나 너무 많은 요청, 페이로드를 암호화하거나 해독할 수 없는 경우, 지정된 ID를 사용하여 무선 디바이스를 찾을 수 없는 경우 등이 있습니다. 다음 예는 메시지를 처리하는 동안 발생하는 권한 오류를 보여줍니다.

```
{
    "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 디바이스 로그 항목

Sidewalk 디바이스의 로그 항목은 다음 이벤트 유형에 따라 분류할 수 있습니다.

- **`Registration`**

  이 로그 항목은 AWS IoT 무선에 등록하는 모든 Sidewalk 디바이스의 상태를 모니터링하는 데 도움이 됩니다. 이 이벤트 유형의 경우 무선 디바이스 리소스에 대한 CLI를 구성할 때 로그 수준을 `INFO`로 설정하면 `logLevel`이 `INFO` 및 `ERROR`인 로그 메시지가 표시됩니다. 메시지에는 시작부터 완료까지 등록 진행에 대한 세부 정보가 포함됩니다. `ERROR` 로그 메시지에는 디바이스 등록 관련 문제를 해결하는 방법에 대한 정보가 포함됩니다.

다음은 로그 수준이 INFO인 로그 메시지에 대한 예를 보여줍니다. 로그 항목의 logLevel 및 기타 필드에 대한 자세한 내용은 AWS IoT 무선 및 로그 수준 단원을 참조하세요.

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "8d0b2775-e19b-4b2a-a351-cb8a2734a504",
    "wirelessDeviceType": "Sidewalk",
    "event": "Registration",
    "logLevel": "INFO",
    "message": "Successfully completed device registration. Amazon SidewalkId =
 2000000002"
}
```

- Uplink_Data 및 Downlink_Data

  Sidewalk 디바이스의 이벤트 유형 Uplink_Data 및 Downlink_Data는 LoRaWAN 디바이스의 해당 이벤트 유형과 유사합니다. 자세한 내용은 LoRaWAN 디바이스 로그 항목에 대해 앞서 설명한 Uplink_Data 및 Downlink_Data 단원을 참조하세요.

다음 단계

리소스에 대한 로그 항목을 보는 방법과, AWS IoT 무선에 대한 로깅을 활성화한 후 CloudWatch 콘솔에서 볼 수 있는 다양한 로그 항목을 보는 방법을 배웠습니다. 로그 그룹을 사용하여 필터 스트림을 생성할 수 있지만 CloudWatch Insights를 사용하여 필터 스트림을 생성 및 사용하는 것이 좋습니다. 자세한 내용은 CloudWatch 인사이트를 사용하여 AWS IoT 무선의 로그 필터링 단원을 참조하십시오.

CloudWatch 인사이트를 사용하여 AWS IoT 무선의 로그 필터링

CloudWatch Logs를 사용하여 필터 표현식을 생성할 수 있지만 CloudWatch 인사이트를 사용하여 애플리케이션에 따라 필터 표현식을 보다 효과적으로 생성하고 사용하는 것이 좋습니다.

먼저 CloudWatch 로그 그룹을 사용하여 콘솔에서 로그 항목을 보는 데 사용할 수 있는 다양한 리소스 유형, 해당 이벤트 유형 및 로그 수준에 대해 알아보는 것이 좋습니다. 그런 다음 이 페이지에 있는 일부 필터 표현식의 예를 참조로 사용하여 AWS IoT 무선 리소스에 대한 고유한 필터를 생성할 수 있습니다.

CloudWatch Logs 인사이트 콘솔에서 AWS IoT 로그 보기

CloudWatch 콘솔에서 CloudWatch Logs는 /aws/iotwireless라는 로그 그룹에 나타납니다. CloudWatch Logs에 대한 자세한 내용은 CloudWatch Logs를 참조하세요.

CloudWatch 콘솔에서 AWS IoT 로그를 보려면

CloudWatch 콘솔로 이동하여 탐색 창에서 로그 인사이트를 선택합니다.

1. 필터 텍스트 상자에 **/aws/iotwireless**를 입력한 다음 /aws/iotwireless 로그 인사이트를 선택합니다.
2. 로그 그룹의 전체 목록을 보려면 로그 그룹 선택을 선택합니다. AWS IoT 무선에 대한 로그 그룹을 보려면 /aws/iotwireless를 선택합니다.

이제 쿼리 입력을 시작하여 로그 그룹을 필터링할 수 있습니다. 다음 섹션에는 리소스 지표에 대한 인사이트를 얻는 데 도움이 되는 몇 가지 유용한 쿼리가 포함되어 있습니다.

필터링할 유용한 쿼리를 만들고 AWS IoT 무선을 위한 인사이트 확보

필터 표현식을 사용하여 CloudWatch 인사이트를 통해 유용한 로그 정보를 추가로 표시할 수 있습니다. 다음은 몇 가지 샘플 쿼리를 보여줍니다

특정 리소스 유형에 대한 로그만 표시

LoRaWAN 게이트웨이 또는 Sidewalk 디바이스와 같은 특정 리소스 유형에 대한 로그만 표시하는 데 도움이 되는 쿼리를 만들 수 있습니다. 예를 들어 Sidewalk 디바이스에 대한 메시지만 표시하도록 로그를 필터링하려면 다음 쿼리를 입력하고 쿼리 실행을 선택합니다. 이 쿼리를 저장하려면 저장을 선택합니다.

```
fields @message
| filter @message like /Sidewalk/
```

쿼리가 실행된 후, 결과는 로그 탭에서 볼 수 있고, 여기서 계정의 Sidewalk 디바이스와 관련된 로그의 타임스탬프가 표시됩니다. 이전에 Sidewalk 디바이스와 관련된 이벤트가 발생한 경우 이벤트가 발생한 시간을 표시하는 막대 그래프도 표시됩니다. 다음은 로그 탭의 결과 중 하나를 확장하는 경우의 예를 보여줍니다. 또는 Sidewalk 디바이스와 관련된 오류를 해결하려는 경우 로그 수준을 ERROR로 설정하는 다른 필터를 추가하여 오류 정보만 표시할 수 있습니다.

```
Field           Value
@ingestionTime      1623894967640
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-715adccfb34170214ec2f6667ddfa13cb5af2c3ddfc52fbeee0e554a2e780bed
@message      {
```

```
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "Sidewalk",
                    "devEui": "fefff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Successfully sent downlink message. Amazon SidewalkId =
 2000000006, Sequence number = 0"
                    }
@timestamp          1623894967640
devEui             fefff000000011a
event         Downlink_Data
logLevel            INFO
message           Successfully sent downlink message. Amazon SidewalkId = 2000000006,
 Sequence number = 0
messageId     7e752a10-28f5-45a5-923f-6fa7133fedda
resource      WirelessDevice
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  Sidewalk
```

특정 메시지 또는 이벤트 표시

특정 메시지를 표시하고 이벤트가 발생한 시기를 관찰하는 데 도움이 되는 쿼리를 만들 수 있습니다. 예를 들어 LoRaWAN 무선 디바이스에서 다운링크 메시지가 전송된 시기를 확인하려면 다음 쿼리를 입력하고 쿼리 실행을 선탣합니다. 이 쿼리를 저장하려면 저장을 선택합니다.

```
filter @message like /Downlink message sent/
```

쿼리가 실행되면 다운링크 메시지가 무선 디바이스에 성공적으로 전송되었을 때의 타임스탬프를 보여주는 결과가 로그 탭에 표시됩니다. 이전에 무선 디바이스에 전송된 다운링크 메시지가 있는 경우 다운링크 메시지가 전송된 시간을 표시하는 막대 그래프도 표시됩니다. 다음은 로그 탭의 결과 중 하나를 확장하는 경우의 예를 보여줍니다. 또는 다운링크 메시지가 전송되지 않은 경우 문제를 디버깅할 수 있도록 메시지가 전송되지 않은 경우에 대한 결과만 표시하도록 쿼리를 수정할 수 있습니다.

```
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
```

다음 단계

CloudWatch Insights를 사용하여 로그 메시지를 필터링하는 쿼리를 생성하여 유용한 정보를 얻는 방법을 배웠습니다. 앞서 설명한 필터 중 일부를 결합하고 모니터링 중인 리소스에 따라 고유한 필터를 디자인할 수 있습니다. CloudWatch Insights 사용에 대한 자세한 내용은 CloudWatch Insights로 로그 데이터 분석을 참조하세요.

CloudWatch Insights를 사용하여 쿼리를 생성한 후 저장한 경우 필요에 따라 저장된 쿼리를 로드하고 실행할 수 있습니다. 또는 CloudWatch Logs Insights 콘솔에서 기록(History) 버튼을 클릭하면 이전에 실행한 쿼리를 보고 필요에 따라 다시 실행하거나 추가 쿼리를 생성하여 추가로 수정할 수 있습니다.

AWS IoT 무선에 대한 이벤트 알림

AWS IoT 무선는 AWS IoT Core에 온보딩하는 LoRaWAN 및 Sidewalk 디바이스에 대한 이벤트를 알리는 메시지를 게시할 수 있습니다. 예를 들어, 계정의 Sidewalk 디바이스가 프로비저닝되거나 등록되었을 때와 같은 이벤트에 대한 알림을 받을 수 있습니다.

리소스가 이벤트에 대해 알림을 받는 방법

특정 이벤트가 발생하면 이벤트 알림이 게시됩니다. 예를 들어 Sidewalk 디바이스가 프로비저닝될 때 이벤트가 생성됩니다. 이러한 이벤트가 발생할 때마다 단일 이벤트 알림이 전송됩니다. 이벤트 알림은 MQTT를 통해 JSON 페이로드와 함께 게시됩니다. 페이로드 내용은 이벤트 유형에 따라 달라집니다.

> **Note**
>
> 이벤트 알림은 한 번 이상 게시됩니다. 하지만 1회 이상 게시하는 것도 가능합니다. 이벤트 알림의 순서는 보장되지 않습니다.

이벤트 및 리소스 유형

다음 표에서는 알림을 수신할 다양한 유형의 이벤트를 보여줍니다. 이벤트 유형은 리소스 유형이 무선 디바이스인지, 무선 게이트웨이인지 또는 Sidewalk 계정인지에 따라 달라집니다. 리소스 수준에서 리소스에 대한 이벤트를 활성화할 수도 있습니다. 이 경우 이벤트가 다음 섹션에 설명된 대로 특정 유형의 모든 리소스 또는 일부 리소스에 적용됩니다. 다양한 이벤트 유형에 대한 자세한 내용은 LoRaWAN 리소스에 대한 이벤트 알림 및 Sidewalk 리소스에 대한 이벤트 알림 섹션을 참조하세요.

리소스를 기반 이벤트 유형

Resource	리소스 유형	이벤트 유형
무선 디바이스	LoRaWAN	조인
	Sidewalk	• 디바이스 등록 상태 • 근방
무선 게이트웨이	LoRaWAN	연결 상태
Sidewalk 계정	Sidewalk	• 디바이스 등록 상태

Resource	리소스 유형	이벤트 유형	
		• 근방	

무선 이벤트 알림 수신 정책

이벤트 알림을 수신하려면 먼저 디바이스가 AWS IoT 디바이스 게이트웨이에 연결하여 MQTT 이벤트 주제를 구독할 수 있는 정책을 사용해야 합니다. 또한 적합한 주제 필터도 구독해야 합니다.

다음은 다양한 무선 이벤트에 대한 알림을 수신하는 데 필요한 정책의 예입니다.

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

무선 이벤트에 대한 MQTT 주제 형식

무선 리소스에 대한 이벤트 알림을 보내기 위해 AWS IoT는 달러($) 기호로 시작하는 MQTT 예약 주제를 사용합니다. 이러한 예약된 주제를 게시하고 구독할 수 있습니다. 그러나 달러 기호로 시작하는 새 주제를 생성할 수는 없습니다.

> **Note**
>
> MQTT 주제는 AWS 계정에 특정하며 `arn:aws:iotwireless:aws-region:AWS-account-ID:topic/Topic` 형식을 사용합니다. 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 주제를 참조하세요.

무선 디바이스용으로 예약된 MQTT 주제는 다음 형식을 사용합니다.

- 리소스 수준 주제

  이러한 주제는 AWS IoT 무선에 온보딩한 AWS 계정에 있는 특정 유형의 리소스에 모두 적용됩니다.

  `$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/resources`
- 식별자 수준 주제

  이러한 주제는 리소스 식별자에 의해 지정된 대로 AWS IoT 무선에 온보딩한 AWS 계정에 있는 특정 유형의 일부 리소스에 적용됩니다.

  `$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/{resourceIdentifierType}/{resourceID}/{id}`

리소스 및 식별자 수준의 주제에 대한 자세한 내용은 이벤트 구성 섹션을 참조하세요.

다음 표에서는 다양한 이벤트에 대한 MQTT 주제의 예를 보여줍니다.

이벤트 및 MQTT 주제

Event	MQTT 주제	참고
Sidewalk 디바이스 등록 상태	• 리소스 수준 주제 `$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/wireless_devices`	• `{eventType}` 은(는) `registered` 또는 `provisioned` 일 수 있습니다. • `{resourceType}` 은(는) `sidewalk_accounts` 또는 `wireless_devices` 일 수 있습니다. • `{resourceID}` 는 `sidewalk_accounts` 의 경우 `amazon_id` , `wireless_devices` 의 경우 `wireless_device_id` 입니다.

Event	MQTT 주제	참고
	• 식별자 수준 주제 `$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}`	

Sidewalk 근접	• 리소스 수준 주제
	
	`$aws/iotwireless/events/proximity/{eventType}/sidewalk/wireless_devices`
	• 식별자 수준 주제
	
	`$aws/iotwireless/events/proximity/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}`
		• `{eventType}` 은(는) `beacon_discovered` 또는 `beacon_lost` 일 수 있습니다.
		• `{resourceType}` 은(는) `sidewalk_accounts` 또는 `wireless_devices` 일 수 있습니다.
		• `{resourceID}` 는 `sidewalk_accounts` 의 경우 `amazon_id` , `wireless_devices` 의 경우 `wireless_device_id` 입니다.

Event	MQTT 주제	참고

LoRaWAN 조인	• 리소스 수준 주제
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices`
	• 식별자 수준 주제
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices/{resourceID}/{id}`
		• `{eventType}` 은 `join_req_0_received` , `join_req_2_received` 또는 `join_accepted` 일수 있습니다.
		• `{resourceID}` 은(는) `wireless_device_id` 또는 `dev_eui`일 수 있습니다.

LoRaWAN 게이트웨이 연결 상태	• 리소스 수준 주제
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways`
	• 식별자 수준 주제
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways/{resourceID}/{id}`
		• `{eventType}` 은(는) `connected` 또는 `disconnected` 일 수 있습니다.
		• `{resourceID}` 은(는) `wireless_gateway_id` 또는 `gateway_eui` 일 수 있습니다.

여러 이벤트에 대한 자세한 내용은 LoRaWAN 리소스에 대한 이벤트 알림 및 Sidewalk 리소스에 대한 이벤트 알림 섹션을 참조하세요.

이러한 주제를 구독한 경우 이벤트 알림 주제 중 하나에 메시지가 게시되면 알림을 받게 됩니다. 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 예약 주제를 참조하세요.

무선 이벤트에 대한 요금

이벤트 구독 및 알림 수신 요금에 대한 자세한 내용은 AWS IoT Core 요금을 참조하세요.

무선 리소스에 이벤트 사용

예약된 주제의 구독자가 메시지를 수신하려면 먼저 이벤트 알림을 사용하도록 설정해야 합니다. 이를 위해 AWS Management Console이나 AWS IoT 무선 API 또는 AWS CLI를 사용할 수 있습니다.

이벤트 구성

특정 유형에 속하는 모든 리소스 또는 개별 무선 리소스에 대한 알림을 보내도록 이벤트를 구성할 수 있습니다. 리소스 유형은 무선 게이트웨이, Sidewalk 파트너 계정 또는 LoRaWan 또는 Sidewalk 디바이스일 수 있는 무선 디바이스일 수 있습니다. 무선 디바이스에 대해 활성화할 수 있는 이벤트 유형에 대한 자세한 내용은 LoRaWAN 리소스 이벤트 유형 및 Sidewalk 리소스의 이벤트 유형 섹션을 참조하세요.

모든 리소스

특정 리소스 유형에 속하는 AWS 계정 내의 모든 리소스와 같은 이벤트의 경우 알림이 수신되도록 활성화할 수 있습니다. 예를 들어, AWS IoT Core for LoRaWAN으로 온보딩한 모든 LoRaWAN 게이트웨이의 연결 상태 변경 사항을 알려주는 이벤트를 활성화할 수 있습니다. 이러한 이벤트를 모니터링하면 리소스 플릿의 특정 LoRaWAN 게이트웨이의 연결이 끊어지거나 AWS 계정 내의 여러 Sidewalk 디바이스에 대해 비콘이 손실되는 경우 알림을 받는 데 도움이 됩니다.

개별 리소스

이벤트 구성에 개별 LoRaWAN 및 Sidewalk 리소스를 추가하고 이에 대해 알림을 활성화할 수도 있습니다. 이렇게 하면 특정 유형의 개별 리소스를 모니터링할 수 있습니다. 예를 들어, 일부 LoRaWAN 및 Sidewalk 디바이스를 구성에 추가하고 이러한 리소스에 대한 조인 또는 디바이스 등록 상태 이벤트에 대한 알림을 받을 수 있습니다.

필수 조건

LoRaWAN 또는 Sidewalk 리소스에는 이벤트 알림을 수신할 수 있는 적절한 정책이 있어야 합니다. 자세한 내용은 무선 이벤트 알림 수신 정책 단원을 참조하십시오.

AWS Management Console을 사용하여 알림 활성화

콘솔에서 이벤트 메시지를 활성화하려면 AWS IoT 콘솔의 설정(Settings) 탭에서 LoRaWAN 및 Sidewalk 이벤트 알림(LoRaWAN and Sidewalk event notification) 섹션으로 이동합니다.

특정 리소스 유형에 속하는 AWS 계정 내의 모든 리소스에 대한 알림이 수신되도록 활성화하고 모니터링할 수 있습니다.

모든 리소스에 대한 알림을 활성화하는 방법

1. LoRaWAN 및 Sidewalk 이벤트 알림(LoRaWAN and Sidewalk event notification) 섹션에서 모든 리소스(All resources) 탭으로 이동하여 작업(Action)을 선택한 다음 이벤트 관리(Manage events)를 선택합니다.
2. 모니터링할 이벤트를 활성화한 다음 이벤트 업데이트(Update events)를 선택합니다. 특정 이벤트를 더 이상 모니터링하지 않으려면 작업(Actions)을 선택하고 이벤트 관리(Manage events)를 선택한 후 해당 이벤트를 비활성화합니다.

특정 리소스 유형에 속하는 AWS 계정 내의 개별 리소스에 대한 알림이 수신되도록 활성화하고 모니터링할 수도 있습니다.

개별 리소스에 대한 알림을 활성화하는 방법

1. LoRaWAN 및 Sidewalk 이벤트 알림(LoRaWAN and Sidewalk event notification) 섹션에서 작업(Action)을 선택한 다음 리소스 추가(Add resources)를 선택합니다.
2. 알림을 받을 리소스와 이벤트를 선택합니다.

    a. LoRaWAN 리소스(LoRaWAN resources)와 Sidewalk 리소스(Sidewalk resources) 중 어떤 리소스에 대한 이벤트를 모니터링할지 선택합니다.

    b. 리소스 유형에 따라 리소스에 대해 활성화하려는 이벤트를 선택할 수 있습니다. 그런 다음 이러한 이벤트를 구독하고 알림을 받을 수 있습니다. 선택 항목:

    - LoRaWAN 리소스(LoRaWAN resources): LoRaWAN 디바이스에 대한 조인(join) 이벤트 또는 LoRaWAN 게이트웨이에 대한 연결 상태(connection status) 이벤트를 활성화할 수 있습니다.

- Sidewalk 리소스: Sidewalk 파트너 계정 및 Sidewalk 디바이스에 대해 디바이스 등록 상태, 근접 이벤트 또는 둘 다를 활성화할 수 있습니다.

3. 선택한 리소스 유형 및 이벤트에 따라 모니터링할 무선 디바이스 또는 게이트웨이를 선택합니다. 모든 리소스에 대해 최대 250개의 리소스를 선택할 수 있습니다.
4. 제출(Submit)을 선택하여 리소스를 추가합니다.

추가하는 리소스는 콘솔의 LoRaWAN 및 Sidewalk 이벤트 알림(LoRaWAN and Sidewalk event notification) 섹션에서 MQTT 주제와 함께 리소스 유형 탭에 표시됩니다.

- LoRaWAN 조인(LoRaWAN join) 이벤트 및 Sidewalk 디바이스에 대한 이벤트는 콘솔의 무선 디바이스(Wireless devices) 섹션에 표시됩니다.
- LoRaWAN 게이트웨이에 대한 연결 상태(Connection status) 이벤트는 무선 게이트웨이(Wireless gateways) 섹션에 표시됩니다.
- Sidewalk 계정에 대한 디바이스 등록 상태(Device registration state) 및 근접(proximity) 이벤트는 Sidewalk 계정(Sidewalk accounts) 탭에 표시됩니다.

MQTT 클라이언트를 사용하여 주제 구독

이벤트를 모든 리소스와 개별 리소스 중 어느 유형에 대해 활성화했는지에 따라 활성화한 이벤트는 MQTT 주제와 함께 콘솔의 모든 리소스(All resources) 탭 또는 지정된 리소스 유형의 탭에 표시됩니다.

- MQTT 주제 중 하나를 선택하는 경우 MQTT 클라이언트로 이동하여 이러한 주제를 구독하고 메시지를 받을 수 있습니다.
- 여러 이벤트를 추가한 경우 여러 이벤트 주제를 구독하고 해당 항목에 대한 알림을 받을 수 있습니다. 여러 주제를 구독하려면 주제를 선택하고 작업(Action)을 선택한 다음 구독(Subscribe)을 선택합니다.

AWS CLI를 사용하여 알림 활성화

AWS IoT 무선 API 또는 AWS CLI를 사용하여 이벤트를 구성하고 구성에 리소스를 추가할 수 있습니다.

모든 리소스에 대한 알림을 활성화하는 방법

특정 리소스 유형에 속하는 AWS 계정의 모든 리소스에 대해 알림을 활성화하고 UpdateEventConfigurationByResourceTypes API 또는 update-event-configuration-by-resource-types CLI 명령을 사용하여 모니터링할 수 있습니다. 예:

```
aws iotwireless update-event-configuration-by-resource-types \
   --cli-input-json input.json
```

input.json 내용

```
{
   "DeviceRegistrationState": {
      "Sidewalk": {
         "AmazonIdEventTopic": "Enabled"
      }
   },
   "ConnectionStatus": {
      "LoRaWAN": {
         "WirelessGatewayEventTopic": "Enabled"
      }
   }
}
```

> **Note**
>
> 모든 큰따옴표(")는 백슬래시(\)로 이스케이프됩니다.

GetEventConfigurationByResourceTypes API를 호출하거나 get-event-configuration-by-resource-types CLI 명령을 사용하여 현재 이벤트 구성을 가져올 수 있습니다. 예:

```
aws iotwireless get-event-configuration-by-resource-types
```

개별 리소스에 대한 알림을 활성화하는 방법

API 또는 CLI를 사용하여 이벤트 구성에 개별 리소스를 추가하고 게시할 이벤트 유형을 제어하려면 UpdateResourceEventConfiguration API를 호출하거나 update-resource-event-configuration CLI 명령을 사용합니다. 예:

```
aws iotwireless update-resource-event-configuration \
   --identifer 1ffd32c8-8130-4194-96df-622f072a315f \
```

```
    --identifier-type WirelessDeviceId \
    --cli-input-json input.json
```

input.json 내용

```
{
   "Join": {
      "LoRaWAN": {
         "DevEuiEventTopic": "Disabled"
      },
      "WirelessDeviceIdEventTopic": "Enabled"
   }
}
```

> **Note**
>
> 모든 큰따옴표(")는 백슬래시(\)로 이스케이프됩니다.

GetResourceEventConfiguration API를 호출하거나 `get-resource-event-configuration` CLI 명령을 사용하여 현재 이벤트 구성을 가져올 수 있습니다. 예:

```
aws iotwireless get-resource-event-configuration \
    --identifier-type WirelessDeviceId \
    --identifier 1ffd32c8-8130-4194-96df-622f072a315f
```

이벤트 구성 나열

AWS IoT 무선 API 또는 AWS CLI를 사용하여 하나 이상의 이벤트 주제가 활성화된 이벤트 구성을 나열할 수도 있습니다. 구성을 나열하려면 ListEventConfigurations API 작업 또는 `list-event-configurations` CLI 명령을 사용합니다. 예:

```
aws iotwireless list-event-configurations --resource-type WirelessDevice
```

LoRaWAN 리소스에 대한 이벤트 알림

AWS Management Console 또는 AWS IoT 무선 API 작업을 통해 LoRaWAN 디바이스 및 게이트웨이에 대한 이벤트를 알릴 수 있습니다. 이벤트 알림과 알림을 활성화하는 방법에 대한 자세한 내용은 AWS IoT 무선에 대한 이벤트 알림 및 무선 리소스에 이벤트 사용 단원을 참조하세요.

LoRaWAN 리소스 이벤트 유형

LoRaWAN 리소스에 사용할 수 있는 이벤트는 다음과 같습니다.

- LoRaWAN 디바이스의 조인 이벤트에 대해 알려주는 조인 이벤트입니다. 디바이스가 AWS IoT Core for LoRaWAN에 조인하거나 유형 0 또는 유형 2의 재조인 요청이 수신될 때 알림을 받습니다.
- LoRaWAN 게이트웨이의 연결 상태가 연결 또는 연결 해제로 변경되면 알려주는 연결 상태 이벤트입니다.

다음 섹션에는 LoRaWAN 리소스의 이벤트에 대한 자세한 정보가 포함되어 있습니다.

주제

LoRaWAN 조인 이벤트

AWS IoT Core for LoRaWAN은 AWS IoT에 온보딩하는 LoRaWAN 디바이스에 대한 조인 이벤트를 알리는 메시지를 게시할 수 있습니다. 조인 이벤트는 유형 0 또는 유형 2의 조인 또는 재조인 요청이 수신되고 디바이스가 AWS IoT Core for LoRaWAN에 조인되면 알려줍니다.

조인 이벤트 작동 방식

AWS IoT Core for LoRaWAN으로 LoRaWAN 디바이스를 온보딩할 때 AWS IoT Core for LoRaWAN은 AWS IoT Core for LoRaWAN을 사용하여 디바이스의 조인 절차를 수행합니다. 그러면 디바이스가 사용할 수 있도록 활성화되고 업링크 메시지를 전송하여 사용 가능함을 나타낼 수 있습니다. 디바이스가 조인되면 디바이스와 AWS IoT Core for LoRaWAN 간에 업링크 및 다운링크 메시지를 교환할 수 있습니다. 디바이스 온보딩에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 디바이스 온보딩 섹션을 참조하세요.

디바이스가 AWS IoT Core for LoRaWAN에 조인했을 때 알림을 받도록 이벤트를 활성화할 수 있습니다. 또한 조인 이벤트가 실패하고 유형 0 또는 유형 2의 재조인 요청이 수신될 때와 요청이 수락될 때 알림을 받게 됩니다.

LoRaWAN 조인 이벤트 활성화

LoRaWAN 조인 예약 주제의 구독자가 메시지를 수신하려면 AWS Management Console에서 또는 API나 CLI를 사용하여 해당 주제에 대한 이벤트 알림을 사용하도록 설정해야 합니다. AWS 계정에 있

는 모든 LoRaWAN 리소스 또는 일부 리소스에 대해 이러한 이벤트를 활성화 할 수 있습니다. 이러한 이벤트를 활성화하는 방법에 대한 자세한 내용은 무선 리소스에 이벤트 사용 섹션을 참조하세요.

LoRaWAN 이벤트에 대한 MQTT 주제 형식

LoRaWAN 디바이스용으로 예약된 MQTT 주제는 다음 형식을 사용합니다. 이 주제를 구독한 경우 AWS 계정에 등록된 모든 LoRaWAN 디바이스가 알림을 받을 수 있습니다.

- 리소스 수준 주제

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices
  ```

- 식별자 주제

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices/
  {resourceID}/{id}
  ```

위치:

{eventName}

{eventName}은 `join`이어야 합니다.

{eventType}

{eventType}은 다음과 같습니다.

- `join_req_received`
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`

{resourceID}

{resourceID}는 `dev_eui` 또는 `wireless_device_id`일 수 있습니다.

예를 들어, 다음 주제를 구독하여 AWS IoT Core for LoRaWAN에서 디바이스의 조인 요청을 수락할 때 이벤트 알림을 받을 수 있습니다.

```
$aws/iotwireless/events/join/join_accepted/lorawan/wireless_devices/
wireless_device_id/{id}
```

+ 와일드카드 문자를 사용하여 동시에 여러 주제를 구독할 수도 있습니다. + 와일드카드 문자는 다음 주제와 같이 해당 문자를 포함하는 수준의 모든 문자열과 일치합니다.

`$aws/iotwireless/events/join/join_req_received/lorawan/wireless_devices/wireless_device_id/+`

> **Note**
>
> 예약된 주제를 구독하기 위해 와일드카드 문자 #을 사용할 수 없습니다.

주제 구독 시 + 와일드카드 사용에 관한 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 주제 필터를 참조하세요.

LoRaWAN 조인 이벤트에 대한 메시지 페이로드

다음은 LoRaWAN 조인 이벤트에 대한 메시지 페이로드를 보여줍니다.

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

페이로드에는 다음과 같은 속성이 포함됩니다.

eventId

AWS IoT Core for LoRaWAN에 의해 생성된 고유한 이벤트 ID(문자열)입니다.

eventType

발생한 이벤트의 유형입니다. 다음 값 중 하나일 수 있습니다.

- `join_req_received`: 이 필드는 EUI 파라미터 `JoinEui` 또는 `AppEui`를 표시합니다.
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`: 이 필드는 `NetId` 및 `DevAddr`을 표시합니다.

wirelessDeviceId

LoRaWAN 디바이스의 ID입니다.

timestamp

이벤트가 발생한 시점의 UNIX 타임스탬프입니다.

DevEui

디바이스 레이블 또는 디바이스 설명서에 있는 디바이스의 고유 식별자입니다.

DevAddr 및 EUI(선택 사항)

필드는 선택적 디바이스 주소와 EUI 파라미터 `JoinEUI` 또는 `AppEUI`입니다.

연결 상태 이벤트

AWS IoT Core for LoRaWAN은 AWS IoT에 온보딩하는 LoRaWAN 게이트웨이에 대한 연결 상태 이벤트를 알리는 메시지를 게시할 수 있습니다. LoRaWAN 게이트웨이의 연결 상태가 연결 또는 연결 해제로 변경되면 알려주는 연결 상태 이벤트.

연결 상태 이벤트 작동 방식

AWS IoT Core for LoRaWAN에 게이트웨이를 온보딩한 후 게이트웨이를 AWS IoT Core for LoRaWAN에 연결하고 연결 상태를 확인할 수 있습니다. 이 이벤트는 게이트웨이의 연결 상태가 연결 또는 연결 해제로 변경되면 알려줍니다. AWS IoT Core for LoRaWAN에 게이트웨이를 온보딩 및 연결하는 방법에 대한 자세한 내용은 AWS IoT Core for LoRaWAN에 게이트웨이 온보딩 및 LoRaWAN 게이트웨이를 연결하고 연결 상태를 확인합니다. 섹션을 참조하세요.

LoRaWAN 게이트웨이에 대한 MQTT 주제 형식

LoRaWAN 게이트웨이용으로 예약된 MQTT 주제는 다음 형식을 사용합니다. 이 주제를 구독한 경우 AWS 계정에 등록된 모든 LoRaWAN 게이트웨이가 알림을 받을 수 있습니다.

- 리소스 수준 주제:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways
  ```

- 식별자 주제:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/
  wireless_gateways/{resourceID}/{id}
  ```

위치:

{eventName}

{eventName}은 `connection_status`이어야 합니다.

{eventType}

{eventType}은 `connected` 또는 `disconnected`일 수 있습니다.

{resourceID}

{resourceID}는 `gateway_eui` 또는 `wireless_gateway_id`일 수 있습니다.

예를 들어, 다음 주제를 구독하여 모든 게이트웨이가 AWS IoT Core for LoRaWAN에 연결된 경우 이벤트 알림을 받을 수 있습니다.

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/{id}
```

+ 와일드카드 문자를 사용하여 동시에 여러 주제를 구독할 수도 있습니다. + 와일드카드 문자는 다음 주제와 같이 해당 문자를 포함하는 수준의 모든 문자열과 일치합니다.

```
$aws/iotwireless/events/connection_status/connected/lorawan/
wireless_gateways/wireless_gateway_id/+
```

> **Note**
>
> 예약된 주제를 구독하기 위해 와일드카드 문자 #을 사용할 수 없습니다.

주제 구독 시 + 와일드카드 사용에 관한 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 주제 필터를 참조하세요.

연결 상태 이벤트에 대한 메시지 페이로드

다음은 연결 상태 이벤트에 대한 메시지 페이로드를 보여줍니다.

```
{
 // General fields
    "eventId": "string",
    "eventType": "connected|disconnected",
    "WirelessGatewayId": "string",
    "timestamp": "timestamp",

 // Event-specific fields
    "LoRaWAN": {
        "GatewayEui": "string"
    }
}
```

페이로드에는 다음과 같은 속성이 포함됩니다.

eventId

AWS IoT Core for LoRaWAN에 의해 생성된 고유한 이벤트 ID(문자열)입니다.

eventType

발생한 이벤트의 유형입니다. 가능한 값은 `connected` 또는 `disconnected`입니다.

wirelessGatewayId

LoRaWAN 게이트웨이의 ID입니다.

timestamp

이벤트가 발생한 시점의 UNIX 타임스탬프입니다.

GatewayEui

게이트웨이 레이블 또는 게이트웨이 설명서에 있는 게이트웨이의 고유 식별자입니다.

Sidewalk 리소스에 대한 이벤트 알림

AWS Management Console 또는 AWS IoT 무선 API 작업을 통해 Sidewalk 디바이스 및 파트너 계정에 대한 이벤트를 알릴 수 있습니다. 이벤트 알림과 알림을 활성화하는 방법에 대한 자세한 내용은 AWS IoT 무선에 대한 이벤트 알림 및 무선 리소스에 이벤트 사용 섹션을 참조하세요.

Sidewalk 리소스의 이벤트 유형

Sidewalk 리소스에 사용할 수 있는 이벤트는 다음과 같습니다.

- 디바이스가 등록되어 사용할 준비가 된 경우와 같이 Sidewalk 디바이스 상태의 변경 사항을 알리는 디바이스 이벤트입니다.
- AWS IoT 무선가 Amazon Sidewalk로부터 비콘이 발견되거나 손실되었다는 알림을 수신할 때 알려주는 근접 이벤트.

다음 섹션에는 Sidewalk 리소스의 이벤트에 대한 자세한 정보가 포함되어 있습니다.

주제

디바이스 등록 상태 이벤트

디바이스 등록 상태 이벤트는 Sidewalk 디바이스가 프로비저닝되거나 등록된 경우와 같이 디바이스 등록 상태가 변경될 때 이벤트 알림을 게시합니다. 이 이벤트는 프로비저닝될 때부터 등록될 때까지 디바이스가 거치는 다양한 상태에 대한 정보를 제공합니다.

디바이스 등록 상태 이벤트의 작동 방식

Amazon Sidewalk 및 AWS IoT 무선로 Sidewalk 디바이스를 온보딩하면 AWS IoT 무선에서 `create` 작업을 수행하고 Sidewalk 디바이스를 AWS 계정에 추가합니다. 그러면 디바이스가 프로비저닝됨 상태가 되고 `eventType`이 `provisioned`가 됩니다. 디바이스 온보딩에 대한 자세한 내용은 Amazon Sidewalk용 AWS IoT Core 시작하기 섹션을 참조하세요.

디바이스가 `provisioned`이 된 후 Amazon Sidewalk는 Sidewalk 디바이스를 AWS IoT 무선에 등록하는 `register` 작업을 수행합니다. 암호화 및 세션 키가 AWS IoT로 설정되는 등록 프로세스가 시작됩니다. 디바이스가 등록되면 `eventType`이 `registered`가 되고 디바이스를 사용할 수 있습니다.

디바이스가 등록(`registered`)된 후 Sidewalk에서 디바이스를 등록 취소(`deregister`)하도록 요청을 보낼 수 있습니다. 그런 다음 AWS IoT 무선가 요청을 이행하고 디바이스 상태를 `provisioned`로 되돌립니다. 디바이스 상태에 대한 자세한 내용은 DeviceState를 참조하세요.

디바이스 등록 상태 이벤트에 대한 알림 사용

디바이스 등록 상태 예약 주제의 구독자가 메시지를 수신하려면 AWS Management Console에서 또는 API 또는 CLI를 사용하여 해당 주제에 대한 이벤트 알림을 사용하도록 설정해야 합니다. AWS 계정에 있는 모든 Sidewalk 리소스 또는 일부 리소스에 대해 이러한 이벤트를 활성화 할 수 있습니다. 이러한 이벤트를 활성화하는 방법에 대한 자세한 내용은 무선 리소스에 이벤트 사용 섹션을 참조하세요.

디바이스 등록 상태 이벤트에 대한 MQTT 주제의 형식

디바이스 등록 상태 이벤트를 알리기 위해 달러($) 기호로 시작하는 MQTT 예약 주제를 구독할 수 있습니다. 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 주제를 참조하세요.

Sidewalk 디바이스 등록 상태 이벤트용으로 예약된 MQTT 주제는 다음 형식을 사용합니다.

- 리소스 수준 주제:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 식별자 주제:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

위치:

{eventName}

{eventName}은 `device_registation_state`이어야 합니다.

{eventType}

{eventType}은 `provisioned` 또는 `registered`일 수 있습니다.

{resourceType}

{resourceType}은 `sidewalk_accounts` 또는 `wireless_devices`일 수 있습니다.

{resourceID}

{resourceID}는 {resourceType}이 `sidewalk_accounts`인 경우 `amazon_id`, {resourceType}이 `wireless_devices`인 경우 `wireless_device_id`입니다.

+ 와일드카드 문자를 사용하여 동시에 여러 주제를 구독할 수도 있습니다. + 와일드카드 문자는 해당 문자를 포함하는 수준의 모든 문자열과 일치합니다. 예를 들어, 가능한 모든 이벤트 유형

(provisioned 및 registered)과 특정 Amazon ID에 등록된 모든 디바이스에 대해 알림을 받으려는 경우 다음 주제 필터를 사용할 수 있습니다.

```
$aws/iotwireless/events/device_registration_state/+/sidewalk/
sidewalk_accounts/amazon_id/+
```

> **Note**
>
> 예약된 주제를 구독하기 위해 와일드카드 문자 #을 사용할 수 없습니다. 주제 필터에 대한 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 주제 필터를 참조하세요.

디바이스 등록 상태 이벤트에 대한 메시지 페이로드

디바이스 등록 상태 이벤트에 대한 알림을 사용하도록 설정하면 이벤트 알림이 JSON 페이로드와 함께 MQTT를 통해 게시됩니다. 이러한 이벤트에는 아래와 같은 페이로드 예제가 포함됩니다.

```
{
    "eventId": "string",
    "eventType": "provisioned|registered",
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",

    // Event-specific fields
    "operation": "create|deregister|register",
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

페이로드에는 다음과 같은 속성이 포함됩니다.

eventId

고유한 이벤트 ID(문자열)입니다.

eventType

발생한 이벤트의 유형입니다. 가능한 값은 provisioned 또는 registered입니다.

wirelessDeviceId

무선 디바이스의 식별자입니다.

timestamp

이벤트가 발생한 시점의 UNIX 타임스탬프입니다.

작업을 통해 처리 속도를 높일 수 있습니다

이벤트를 트리거한 작업입니다. 유효한 값은 `create`, `register`, `deregister`입니다.

sidewalk

Sidewalk Amazon ID 또는 이벤트 알림을 받을 `SidewalkManufacturingSn`입니다.

근접 이벤트

근접 이벤트는 AWS IoT가 Sidewalk 디바이스에서 비콘을 수신할 때 이벤트 알림을 게시합니다. Sidewalk 디바이스가 Amazon Sidewalk에 접근하면 디바이스에서 보낸 비콘이 Amazon Sidewalk에 의해 정기적으로 필터링되고 AWS IoT 무선에서 수신합니다. 그런 다음 비콘이 수신될 때 AWS IoT 무선가 이 이벤트에 대해 사용자에게 알립니다.

근접 이벤트의 작동 방식

근접 이벤트는 AWS IoT가 비콘을 수신할 때 알려줍니다. Sidewalk 디바이스는 언제든지 비콘을 방출할 수 있습니다. 디바이스가 Amazon Sidewalk 근처에 있으면 Sidewalk가 비콘을 수신하여 일정한 시간 간격으로 AWS IoT 무선에 전달합니다. Amazon Sidewalk는 이 시간 간격을 10분으로 구성했습니다. AWS IoT 무선가 Sidewalk에서 비콘을 수신하면 이벤트에 대한 알림을 받게 됩니다.

비콘이 발견되거나 비콘이 손실되면 근접 이벤트가 알려줍니다. 근접 이벤트에 대한 알림을 받는 간격을 구성할 수 있습니다.

근접 이벤트에 대한 알림 사용

Sidewalk 근접 예약 주제의 구독자가 메시지를 수신하려면 AWS Management Console에서 또는 API나 CLI를 사용하여 해당 주제에 대한 이벤트 알림을 사용하도록 설정해야 합니다. AWS 계정에 있는 모든 Sidewalk 리소스 또는 일부 리소스에 대해 이러한 이벤트를 활성화 할 수 있습니다. 이러한 이벤트를 활성화하는 방법에 대한 자세한 내용은 무선 리소스에 이벤트 사용 섹션을 참조하세요.

근접 이벤트에 대한 MQTT 주제 형식

근접 이벤트를 알리기 위해 달러($) 기호로 시작하는 MQTT 예약 주제를 구독할 수 있습니다. 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 주제를 참조하세요.

Sidewalk 근접 이벤트용으로 예약된 MQTT 주제는 다음 형식을 사용합니다.

- 리소스 수준 주제:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- 식별자 주제:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

위치:

{eventName}

{eventName}은 `proximity`이어야 합니다.

{eventType}

{eventType}은 `beacon_discovered` 또는 `beacon_lost`일 수 있습니다.

{resourceType}

{resourceType}은 `sidewalk_accounts` 또는 `wireless_devices`일 수 있습니다.

{resourceID}

{resourceID}는 {resourceType}이 `sidewalk_accounts`인 경우 `amazon_id`, {resourceType}이 `wireless_devices`인 경우 `wireless_device_id`입니다.

+ 와일드카드 문자를 사용하여 동시에 여러 주제를 구독할 수도 있습니다. + 와일드카드 문자는 해당 문자를 포함하는 수준의 모든 문자열과 일치합니다. 예를 들어, 가능한 모든 이벤트 유형(`beacon_discovered` 및 `beacon_lost`)과 특정 Amazon ID에 등록된 모든 디바이스에 대해 알림을 받으려는 경우 다음 주제 필터를 사용할 수 있습니다.

```
$aws/iotwireless/events/proximity/+/sidewalk/sidewalk_accounts/amazon_id/+
```

> **Note**
>
> 예약된 주제를 구독하기 위해 와일드카드 문자 #을 사용할 수 없습니다. 주제 필터에 대한 자세한 내용은 AWS IoT 개발자 안내서의 MQTT 주제 필터를 참조하세요.

근접 이벤트에 대한 메시지 페이로드

근접 이벤트에 대한 알림을 사용하면 이벤트 메시지가 JSON 페이로드와 함께 MQTT를 통해 게시됩니다. 이러한 이벤트에는 아래와 같은 페이로드 예제가 포함됩니다.

```
{
    "eventId": "string",
    "eventType": "beacon_discovered|beacon_lost",
    "WirelessDeviceId": "string",
    "timestamp": "1234567890123",

    // Event-specific fields
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

페이로드에는 다음과 같은 속성이 포함됩니다.

eventId

고유한 이벤트 ID로서 문자열입니다.

eventType

발생한 이벤트의 유형입니다. 가능한 값은 `beacon_discovered` 또는 `beacon_lost`입니다.

WirelessDeviceId

무선 디바이스의 식별자입니다.

timestamp

이벤트가 발생한 시점의 UNIX 타임스탬프입니다.

sidewalk

Sidewalk Amazon ID 또는 이벤트 알림을 받을 `SidewalkManufacturingSn`입니다.

AWS IoT 무선 API 작업

LoRaWAN 또는 Sidewalk 엔드 디바이스를 온보딩하거나 Sidewalk 엔드 디바이스를 대량으로 프로비저닝하기 위한 가져오기 태스크를 생성할 때 다음과 같은 추가 API 태스크를 수행할 수 있습니다.

다음 섹션에는 이러한 API 작업에 대한 추가 정보가 포함되어 있습니다.

주제

디바이스 프로필에 대한 AWS IoT 무선 API 작업

LoRaWAN 및 Sidewalk 디바이스 프로필에 대해 다음과 같은 API 작업을 수행할 수 있습니다.

- `CreateDeviceProfile` API 또는 `create-device-profile` CLI
- `GetDeviceProfile` API 또는 `get-device-profile` CLI
- `ListDeviceProfiles` API 또는 `list-device-profiles` CLI
- `DeleteDeviceProfile` API 또는 `delete-device-profile` CLI

다음 섹션에서는 프로필을 나열하고 삭제하는 방법을 보여줍니다. 디바이스 프로필 생성 및 검색에 대한 자세한 내용은 다음을 참조하세요.

- 디바이스 프로파일 추가
- 1단계: 디바이스 프로필 생성

AWS 계정의 디바이스 프로필 나열

`ListDeviceProfiles` API 작업을 사용하여 AWS IoT 무선에 추가한 AWS 계정의 디바이스 프로필을 나열할 수 있습니다. 이 정보를 사용하여 이 프로필을 연결할 디바이스를 식별할 수 있습니다.

LoRaWAN 또는 Sidewalk 디바이스 프로필만 표시하도록 목록을 필터링하려면 API를 실행할 때 `Type`을 설정합니다. 다음은 CLI 명령의 예시입니다.

```
aws iotwireless list-device-profiles --wireless-device-type "Sidewalk"
```

이 명령을 실행하면 프로필 식별자와 Amazon 리소스 이름(ARN)을 포함하여 추가한 디바이스 프로필 목록이 반환됩니다. 특정 프로필에 대한 추가 세부 정보를 검색하려면 `GetDeviceProfile` API를 사용하세요.

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

AWS 계정에서 디바이스 프로필 삭제

`DeleteDeviceProfile` API 작업을 사용하여 디바이스 프로필을 삭제할 수 있습니다. 다음은 CLI 명령의 예시입니다.

> **Warning**
>
> 삭제 작업은 취소할 수 없습니다. 디바이스 프로필이 AWS 계정에서 영구적으로 제거됩니다.

```
aws iotwireless delete-device-profile --name "SidewalkProfile"
```

이 명령은 출력을 생성하지 않습니다. `GetDeviceProfile` API 또는 `ListDeviceProfiles` API 작업을 사용하여 계정에서 프로필이 제거되었는지 확인할 수 있습니다.

LoRaWAN 및 Sidewalk 디바이스에 대한 AWS IoT 무선 API 작업

LoRaWAN 및 Sidewalk 디바이스에 대해 다음과 같은 API 작업을 수행할 수 있습니다.

- `CreateWirelessDevice` API 또는 `create-wireless-device` CLI
- `GetWirelessDevice` API 또는 `get-wireless-device` CLI
- `ListWirelessDevices` API 또는 `list-wireless-devices` CLI
- `DeleteWirelessDevice` API 또는 `delete-wireless-device` CLI
- `UpdateWirelessDevice` API 또는 `update-wireless-device` CLI
- `AssociateWirelessDeviceWithThing` API 또는 `associate-wireless-device-with-thing` CLI
- `DisassociateWirelessDeviceFromThing` API 또는 `disassociate-wireless-device-from-thing` CLI

다음 섹션에서는 디바이스를 나열하고 삭제하는 방법을 보여줍니다. 무선 디바이스 생성 및 디바이스 정보 검색에 대한 자세한 내용은 다음을 참조하세요.

- AWS IoT Core for LoRaWAN에 무선 디바이스 추가
- 2단계: Sidewalk 디바이스 추가

AWS 계정의 무선 디바이스를 IoT 사물에 연결

LoRaWAN 및 Sidewalk 디바이스를 AWS IoT 사물과 연결하려면 `AssociateWirelessDeviceWithThing` API 작업을 사용하세요.

AWS IoT의 사물을 사용하면 디바이스를 더 쉽게 검색하고 관리할 수 있습니다. 사물을 디바이스에 연결하면 디바이스에서 다른 AWS IoT Core 기능에 액세스할 수 있습니다. 이 API에 대한 자세한 내용은 `AssociateWirelessDeviceWithThing` 단원을 참조하세요.

다음은 이 명령을 실행하는 예시를 보여줍니다. 이 명령을 실행하면 출력을 생성하지 않습니다.

```
aws iotwireless associate-wireless-device-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MySidewalkThing"
```

무선 디바이스를 AWS IoT 사물과 분리하려면 다음 예와 같이 `DisassociateWirelessDeviceFromThing` API 작업을 사용하세요.

```
aws iotwireless disassociate-wireless-device-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

AWS 계정의 무선 디바이스 나열

AWS IoT 무선에 추가한 AWS 계정의 무선 디바이스를 나열하려면 ListWirelessDevices API 작업을 사용하세요. LoRaWAN 또는 Sidewalk 디바이스만 반환하도록 목록을 필터링하려면 WirelessDeviceType을 설정합니다.

다음은 이 명령을 실행하는 예시를 보여줍니다.

```
aws iotwireless list-wireless-devices --wireless-device-type Sidewalk
```

이 명령을 실행하면 프로필 식별자와 Amazon 리소스 이름(ARN)을 포함하여 추가한 디바이스 목록이 반환됩니다. 특정 디바이스에 대한 추가 세부 정보를 검색하려면 GetWirelessDevice API 작업을 사용하세요.

```
{
    "WirelessDeviceList": [
        {
            "Name": "mySidewalkDevice",
            "DestinationName": "SidewalkDestination",
            "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
            "Type": "Sidewalk",
            "Sidewalk": {
                "SidewalkId": "1234567890123456"
            },
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f"
        }
    ]
}
```

AWS 계정에서 무선 디바이스 삭제

무선 디바이스를 삭제하려면 삭제하려는 디바이스의 WirelessDeviceID를 DeleteWirelessDevice API 작업에 전달하세요.

다음은 명령의 예시입니다.

```
aws iotwireless delete-wireless-device --id "23456789-abcd-0123-bcde-fabc012345678"
```

이 명령은 출력을 생성하지 않습니다. GetWirelessDevice API 또는 ListWirelessDevices API 작업을 사용하여 계정에서 디바이스가 제거되었는지 확인할 수 있습니다.

무선 디바이스 대상에 대한 AWS IoT 무선 API 작업

LoRaWAN 및 Sidewalk 디바이스 대상에 대해 다음과 같은 API 작업을 수행할 수 있습니다.

- CreateDestination API 또는 create-destination CLI
- GetDestination API 또는 get-destination CLI
- UpdateDestination API 또는 update-destination CLI
- ListDestinations API 또는 list-destinations CLI
- DeleteDestination API 또는 delete-destination CLI

다음 섹션에서는 대상을 가져오고, 나열하고, 업데이트하고, 삭제하는 방법을 보여줍니다. 대상 생성에 대한 자세한 내용은 Sidewalk 엔드 디바이스의 대상 추가 섹션을 참조하세요.

대상에 대한 정보 가져오기

GetDestination API 작업을 사용하여 계정에서 AWS IoT 무선에 추가한 Sidewalk 디바이스에 대한 정보를 가져올 수 있습니다. API에 대상 이름을 입력으로 제공합니다. API가 지정된 식별자와 일치하는 대상에 대한 정보를 반환합니다.

다음은 CLI 명령의 예시입니다.

```
aws iotwireless get-destination --name SidewalkDestination
```

이 명령을 실행하면 대상의 파라미터가 반환됩니다.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "SidewalkDestination",
    "Expression": "IoTWirelessRule",
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

대상의 속성 업데이트

UpdateDestination API 작업을 사용하여 계정에서 AWS IoT 무선에 추가한 목적지의 속성을 업데이트합니다. 다음은 설명 속성을 업데이트하는 CLI 명령 예시입니다.

```
aws iotwireless update-destination --name SidewalkDestination \
    --description "Destination for messages processed using IoTWirelessRule"
```

AWS 계정에서 대상 나열

ListDestinations API 작업을 사용하여 AWS 계정에서 AWS IoT 무선에 추가한 대상을 나열합니다. LoRaWAN 및 Sidewalk 엔드 디바이스의 대상만 반환하도록 목록을 필터링하려면 `WirelessDeviceType` 파라미터를 사용하세요.

다음은 CLI 명령의 예시입니다.

```
aws iotwireless list-destinations --wireless-device-type "Sidewalk"
```

이 명령을 실행하면 Amazon 리소스 이름(ARN)을 포함하여 추가한 대상 목록이 반환됩니다. 특정 대상에 대한 추가 세부 정보를 검색하려면 `GetDestination` API를 사용하세요.

```
{
    "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 계정에서 대상 삭제

대상을 삭제하려면 삭제할 대상의 이름을 DeleteDestination API 작업에 입력으로 전달하세요. 다음은 CLI 명령의 예시입니다.

> **Warning**
>
> 삭제 작업은 취소할 수 없습니다. 대상이 AWS 계정에서 영구적으로 제거됩니다.

```
aws iotwireless delete-destination --name "SidewalkDestination"
```

이 명령은 출력을 생성하지 않습니다. GetDestination API 또는 ListDestinations API 작업을 사용하여 계정에서 대상이 제거되었는지 확인할 수 있습니다.

대량 프로비저닝을 위한 AWS IoT Core for Amazon Sidewalk API 작업

Sidewalk 엔드 디바이스 대량 프로비저닝을 위해 다음과 같은 API 작업을 수행할 수 있습니다.

- StartWirelessDeviceImportTask API 또는 start-wireless-device-import-task CLI
- StartSingleWirelessDeviceImportTask API 또는 start-single-wireless-device-import-task CLI
- ListWirelessDeviceImportTasks API 또는 list-wireless-device-import-tasks CLI
- ListDevicesForWirelessDeviceImportTask API 또는 list-devices-for-wireless-device-import-task CLI
- GetWirelessDeviceImportTask API 또는 get-wireless-device-import-task CLI
- UpdateWirelessDeviceImportTask API 또는 update-wireless-device-import-task CLI
- DeleteWirelessDeviceImportTask API 또는 delete-wireless-device-import-task CLI

다음 섹션에서는 가져오기 작업을 가져오고, 나열하고, 업데이트하고, 삭제하는 방법을 보여줍니다. 가져오기 작업 생성에 대한 자세한 내용은 대량 프로비저닝을 위한 AWS IoT Core for Amazon Sidewalk API 작업 섹션을 참조하세요.

가져오기 작업에 대한 정보 가져오기

ListDevicesForWirelessDeviceImportTask API 작업을 사용하여 특정 가져오기 작업과 해당 작업에 포함된 디바이스의 온보딩 상태에 대한 정보를 검색할 수 있습니다. StartWirelessDeviceImportTask 또는 StartSingleWirelessDeviceImportTask API 작업에서 얻은 가져오기 작업 ID를 API 작업에 대한 입력으로 지정하세요. 그러면 API가 지정된 식별자와 일치하는 가져오기 작업에 대한 정보를 반환합니다.

다음은 CLI 명령의 예시입니다.

```
aws iotwireless list-devices-for-wireless-device-import-task --id e2a5995e-743b-41f2-a1e4-3ca6a5c5249f
```

이 명령을 실행하면 가져오기 작업 정보와 디바이스 온보딩 상태가 반환됩니다.

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

가져오기 작업 디바이스 요약 가져오기

특정 가져오기 작업에 추가한 디바이스의 온보딩 상태에 대한 요약 정보의 수를 가져오려면 GetWirelessDeviceImportTask API 작업을 사용하세요. 다음은 CLI 명령의 예시입니다.

```
aws iotwireless get-wireless-device-import-task --Id "e2a5995e-743b-41f2-
a1e4-3ca6a5c5249f"
```

다음 코드는 명령의 샘플 응답을 보여줍니다.

```
{
   "NumberOfFailedImportedDevices": 2,
   "NumberOfOnboardedImportedDevices": 4,
   "NumberOfPendingImportedDevices": 1
}
```

가져오기 작업에 디바이스 추가

UpdateWirelessDeviceImportTask API 작업을 사용하여 추가한 기존 가져오기 작업에 디바이스를 추가합니다. StartWirelessDeviceImportTask API 작업을 사용하여 생성한 작업에 이전에 포함되지 않았던 디바이스의 일련번호(SMSN)를 추가하는 데 이 API 작업을 사용할 수 있습니다.

디바이스를 가져오기 작업에 추가하려면 API 요청의 일부로 추가할 디바이스의 일련번호가 포함된 Amazon S3 버킷에 새 CSV 파일을 지정하세요. 요청은 현재 가져오기 작업 중인 디바이스에 대한 온보딩 프로세스가 아직 시작되지 않은 경우에만 수락됩니다. 온보딩 프로세스가 이미 시작된 경우 UpdateWirelessDeviceImportTask API 요청은 실패합니다.

여전히 가져오기 작업에 디바이스를 추가하려는 경우 UpdateWirelessDeviceImportTask API 작업을 다시 수행할 수 있습니다. 이 API 작업을 수행하기 전에 첫 번째 UpdateWirelessDeviceImportTask API 요청에서 S3 버킷의 CSV 파일 처리가 완료된 상태여야 합니다.

> **Note**
>
> ListImportedWirelessDeviceTasks API 요청을 수행할 때 UpdateWirelessDeviceImportTask API 작업을 사용하여 지정된 새 CSV 파일의 S3 URL은 현재 반환되지 않습니다. 대신 API 작업은 원래 StartWirelessDeviceImportTask API 요청을 사용하여 전송된 요청의 S3 URL을 반환합니다.

다음은 CLI 명령의 예시입니다.

```
aws iotwireless update-wireless-device-import task \
```

```
    --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f" \
    --sidewalk '{"FileForCreateDevices": "s3://import_task_bucket/import_file3"}'
```

AWS 계정에서 가져오기 작업 나열

`ListWirelessDeviceImportTasks` API 또는 `list-imported-wireless-device-tasks` CLI 명령을 사용하여 AWS 계정에서 가져오기 작업을 나열합니다. 다음은 CLI 명령의 예시입니다.

```
aws iotwireless list-wireless-device-import-tasks
```

이 명령을 실행하면 생성한 가져오기 작업 목록이 반환됩니다. 목록에는 Amazon S3 CSV 파일과 지정된 IAM 역할, 가져오기 작업 ID, 디바이스 온보딩 상태의 요약 정보가 포함됩니다.

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

AWS 계정에서 가져오기 작업 삭제

가져오기 작업을 삭제하려면 가져오기 작업 ID를 `DeleteWirelessDeviceImportTask` API 작업 또는 `delete-wireless-device-import-task` CLI 명령에 전달하세요.

> **Warning**
>
> 삭제 작업은 취소할 수 없습니다. 가져오기 작업이 AWS 계정에서 영구적으로 제거됩니다.

`DeleteWirelessDeviceImportTask` API 요청을 수행할 때 백그라운드 프로세스가 가져오기 작업을 삭제하기 시작합니다. 요청이 진행 중이면 가져오기 작업에 있는 디바이스의 일련번호(SMSN)가 삭제되는 중입니다. 삭제가 완료된 후에야 `ListImportedWirelessDeviceTasks` 또는 `GetImportedWirelessDeviceTasks` API 작업을 사용하여 이 정보를 볼 수 있습니다.

가져오기 작업에 여전히 온보딩 대기 중인 디바이스가 포함되어 있는 경우, 가져오기 작업에 포함된 모든 디바이스가 온보딩되거나 온보딩에 실패한 후에만 `DeleteWirelessDeviceImportTask` API 요청이 처리됩니다. 가져오기 작업은 90일 후에 만료되며 작업이 만료되면 계정에서 삭제할 수 있습니다. 하지만 가져오기 작업을 사용하여 성공적으로 온보딩된 디바이스는 삭제되지 않습니다.

> **Note**
>
> `DeleteWirelessDeviceImportTask` API 요청을 사용하여 삭제 보류 중인 디바이스의 일련번호가 포함된 다른 가져오기 작업을 생성하려고 하면 `StartWirelessDeviceImportTask` API 작업에서 오류가 반환됩니다.

다음은 CLI 명령의 예시입니다.

```
aws iotwireless delete-import-task --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

이 명령은 출력을 생성하지 않습니다. 작업이 삭제된 후 계정에서 가져오기 작업이 제거되었는지 확인하려면 `GetWirelessDeviceImportTask` API 작업 또는 `ListWirelessDeviceImportTasks` API 작업을 사용하면 됩니다.

AWS CloudFormation을 사용하여 AWS IoT Wireless 리소스 생성

AWS IoT Wireless는 리소스 및 인프라를 생성하고 관리하는 데 소요되는 시간을 줄일 수 있도록 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스인 AWS CloudFormation과 통합됩니다. 필요한 모든 AWS 리소스를 설명하는 템플릿을 생성하면 AWS CloudFormation이 해당 리소스의 프로비저닝과 구성을 담당합니다.

AWS CloudFormation을 사용할 때 템플릿을 재사용하여 AWS IoT Wireless 리소스를 일관되고 반복적으로 설정할 수 있습니다. 리소스를 한 번 설명한 다음 여러 AWS 계정 및 리전에서 동일한 리소스를 반복적으로 프로비저닝할 수 있습니다.

AWS IoT Wireless 및 AWS CloudFormation 템플릿

AWS IoT Wireless 및 관련 서비스에 대한 리소스를 프로비저닝하고 구성하려면 AWS CloudFormation 템플릿을 이해해야 합니다. 템플릿은 JSON 또는 YAML로 서식 지정된 텍스트 파일입니다. 이 템플릿은 AWS CloudFormation 스택에서 프로비저닝할 리소스에 대해 설명합니다. JSON 또는 YAML에 익숙하지 않은 경우 AWS CloudFormation Designer를 사용하면 AWS CloudFormation 템플릿을 시작하는 데 도움이 됩니다. 자세한 내용은 AWS CloudFormation 사용 설명서에서 AWS CloudFormation Designer이란 무엇입니까?를 참조하세요.

AWS IoT Wireless는 AWS CloudFormation에서 무선 리소스를 생성할 수 있도록 지원합니다. AWS IoT 무선 리소스에 대한 JSON 및 YAML 템플릿의 예를 비롯한 자세한 내용은 AWS CloudFormation 사용 설명서의 AWS IoT 무선 리소스 유형 참조를 참조하세요.

AWS CloudFormation에 대해 자세히 알아보기

AWS CloudFormation에 대한 자세한 내용은 다음 리소스를 참조하십시오.

- AWS CloudFormation
- AWS CloudFormation 사용 설명서
- AWS CloudFormation 명령줄 인터페이스 사용 설명서

AWS IoT Wireless의 할당량

AWS 계정에는 각 AWS 서비스에 대한 기본 할당량(이전에는 한도라고 함)이 있습니다. 다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다. 일부 할당량에 대한 증가를 요청할 수 있으며 다른 할당량은 늘릴 수 없습니다.

AWS IoT Wireless의 할당량을 보려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스s를 선택하고 AWS IoT Wireless를 선택합니다.

할당량 증가를 요청하려면 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하십시오. Service Quotas에서 아직 할당량을 사용할 수 없는 경우 한도 증가 양식을 사용합니다.

AWS IoT 무선의 할당량은 다음과 같습니다.

- 디바이스 간에 전송되는 디바이스 데이터에 적용되는 AWS IoT Core for LoRaWAN 할당량
- LoRaWAN 및 Sidewalk 디바이스 모두에 적용되는 AWS IoT 무선 API 작업.

자세한 정보는 AWS 일반 참조의 AWS IoT Core for LoRaWAN 할당량을 참조하세요.

AWS IoT 무선 리소스에 태그 지정

디바이스, 게이트웨이, 대상 및 프로필을 쉽게 관리 및 구성하기 위해 이러한 각 리소스에 고유한 메타데이터를 태그의 형태로 할당할 수 있습니다(선택 사항). 이 단원에서는 태그를 설명하고 태그를 생성하는 방법을 보여 줍니다. AWS IoT 무선에는 결제 그룹이 없으며 AWS IoT Core와 동일한 결제 그룹을 사용합니다. 자세한 내용은 AWS IoT Core 설명서의 결제 그룹을 참조하세요.

태그 기본 사항

같은 유형의 AWS IoT 무선 리소스가 여러 개 있는 경우 태그를 사용하여 리소스를 용도, 소유자, 환경 등으로 다양하게 분류할 수 있습니다. 이렇게 하면 지정한 태그를 기반으로 리소스를 신속하게 식별할 수 있습니다.

각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다. 예를 들어, 디바이스 펌웨어가 업데이트되는 LoRaWAN 디바이스 그룹에 태그 세트를 정의할 수 있습니다. 리소스를 더 쉽게 관리할 수 있도록 각 리소스 유형의 요구 사항을 충족하는 태그 키 세트를 생성하는 것이 좋습니다.

추가하거나 적용한 태그를 기준으로 리소스를 검색하고 필터링할 수 있습니다. 또한 IAM 정책 및 결제 그룹 태그를 사용하여 비용을 분류하고 추적함으로써 태그를 사용하여 리소스에 대한 액세스를 제어할 수 있습니다.

태그 생성 및 관리

AWS Management Console, AWS IoT 무선 또는 AWS CLI의 태그 편집기를 사용하여 태그를 생성하고 관리할 수 있습니다.

콘솔 사용

태그 편집기는 태그를 생성하고 관리할 수 있는 AWS Management Console 중앙 통합 방식으로, 이 도구를 사용하면 아주 편리합니다. 자세한 내용은 AWS Management Console과 작업에서 태그 편집기 사용하기를 참조하십시오.

API 또는 CLI 사용

또한 API 또는 CLI를 사용하고, 다음 명령의 Tags 필드를 사용하여 태그를 생성할 때 무선 디바이스, 게이트웨이, 프로필 및 대상에 연결할 수 있습니다.

- AssociateAwsAccountWithPartnerAccount

- CreateDestination
- CreateDeviceProfile
- CreateFuotaTask
- CreateMulticastGroup
- CreateServiceProfile
- CreateWirelessGateway
- CreateWirelessGatewayTaskDefinition
- CreateWirelessDevice
- API_StartBulkAssociateWirelessDeviceWithMulticastGroup

리소스에 대한 태그 또는 목록 태그 업데이트

다음 명령을 사용하여 태깅을 지원하는 기존 리소스에 대해 태그를 추가, 수정, 삭제할 수 있습니다.

- TagResource
- ListTagsForResource
- UntagResource

태그 키와 값을 편집할 수 있으며 언제든지 리소스에서 태그를 제거할 수 있습니다. 태그의 값을 빈 문자열로 설정할 수 있지만 태그의 값을 Null로 설정할 수는 없습니다. 해당 리소스에 대해 키가 기존 태그와 동일한 태그를 추가하는 경우 새 값이 이전 값을 덮어씁니다. 리소스를 삭제하면, 리소스에 대한 연결이 완료된 태그 또한 삭제됩니다.

태그 규제 및 제한

태그에 적용되는 기본 제한은 다음과 같습니다.

- 리소스당 최대 태그 수 — 50개.
- 최대 키 길이 — UTF-8의 유니코드 문자 127자
- 최대 값 길이 — UTF-8의 유니코드 문자 255자
- 태그 키와 값은 대소문자를 구분합니다.
- 태그 이름이나 값에서 접두사 `aws:`는 사용하지 마세요. 그것은 AWS 전용입니다. 이 접두사가 지정된 태그 이름이나 값은 편집하거나 삭제할 수 없습니다. 이 접두사가 지정된 태그는 리소스당 태그 수 제한에 포함되지 않습니다.

- 태깅 스키마를 여러 서비스와 리소스에서 사용하는 경우 다른 서비스에서는 허용되는 문자에 제한이 있을 수 있다는 점에 주의하세요. 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 공백 및 숫자와 + - = . _ : / @ 등의 특수 문자입니다.

IAM 정책에 태그 사용

사용자가 생성, 수정 또는 사용할 수 있는 리소스를 지정하려면 AWS IoT 무선 API 작업에 사용하는 IAM 정책에 태그 기반의 리소스 수준 권한을 적용할 수 있습니다. 리소스 태그를 기반으로 사용자 액세스(권한)를 제어하기 위해 IAM 정책에서 다음 조건 컨텍스트 키 및 값과 함께 `Condition` 요소(`Condition` 블록이라고도 함)를 사용합니다.

- `aws:ResourceTag/`*`tag-key`*`: `*`tag-value`*를 사용하여 특정 태그가 지정된 리소스에 대한 사용자 작업을 허용 또는 거부합니다.
- `aws:RequestTag/`*`tag-key`*`: `*`tag-value`*를 사용하여 태그를 허용하는 리소스를 생성하거나 수정하는 API 요청을 작성할 때 특정 태그를 사용하도록(또는 사용하지 않도록) 요구합니다.
- `aws:TagKeys: [`*`tag-key`*`, ...]`를 사용하여 태깅 가능한 리소스를 생성하거나 수정하는 API 요청을 작성할 때 특정 태그 키 집합을 사용하도록(또는 사용하지 않도록) 요구합니다.

> **Note**
>
> IAM 정책의 조건 컨텍스트 키와 값은 태깅 가능한 리소스의 ID가 필수 파라미터인 AWS IoT 작업에만 적용됩니다. 예를 들어, 이 요청에서 태그를 지정할 수 있는 리소스가 언급되지 않기 때문에 조건 컨텍스트 키 및 값에 따라 DescribeEndpoint의 사용이 허용/거부되지 않습니다.

자세한 내용은 AWS Identity and Access Management 사용 설명서의 태그를 사용한 액세스 제어를 참조하세요. 이 설명서의 IAM JSON 정책 참조 단원에서는 IAM에서 JSON 정책의 자세한 구문과 설명, 요소의 예, 변수 및 평가 로직을 설명합니다.

다음은 태그 기반 제한 2개를 적용하는 정책 예제입니다. 이 정책으로 제한되는 IAM 사용자는 다음과 같습니다.

- 리소스에 태그 "env=prod"를 지정할 수 없습니다. 이 예제에서는 `"aws:RequestTag/env" : "prod"` 행을 참조하세요.
- 기존 태그 "env=prod"가 지정된 리소스를 수정 또는 액세스할 수 없습니다. 이 예제에서는 `"aws:ResourceTag/env" : "prod"` 행을 참조하세요.

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

또한 다음과 같이 목록에서 태그를 둘러싸 지정된 태그 키에 대해 여러 태그 값을 지정할 수도 있습니다.

```
"StringEquals" : {
              "aws:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> 태그를 기준으로 리소스에 대한 사용자 액세스를 허용 또는 거부하는 경우 동일한 리소스에서 태그를 추가 또는 제거할 수 있도록 사용자를 명시적으로 거부할 것을 고려해야 합니다. 그렇지 않으면 사용자가 제한을 피해 태그를 수정하여 리소스에 대한 액세스 권한을 얻을 수 있습니다.

AWS IoT Wireless 사용 설명서의 문서 기록

다음 표에서는 AWS IoT Wireless에 대한 문서 릴리스를 설명합니다.

변경 사항	설명	날짜
최초 릴리스	AWS IoT Wireless 사용 설명서의 최초 릴리스입니다.	2020년 12월 31일