



개발자 가이드

AWS IoT 1-Click



AWS IoT 1-Click: 개발자 가이드

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon이 제공하지 않는 제품 또는 서비스와 관련하여 고객에게 혼동을 유발할 수 있는 방식 또는 Amazon을 폄하하거나 평판에 악영향을 주는 방식으로 사용될 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계 여부에 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS IoT 1-Click이란 무엇입니까?	1
AWS IoT 1-Click 구성 요소	1
AWS IoT 1-Click 작동 방식	3
AWS IoT 1-Click 디바이스	3
디바이스 신청	4
프로젝트, 템플릿 및 배치	4
AWS IoT 1-Click 콘솔 시작하기	7
디바이스 할당	7
프로젝트 만들기	7
예: 회의실 만족도 프로젝트	8
AWS IoT 1-Click 모바일 앱	11
AWS IoT 1-Click 프로그래밍 모델	12
AWS IoT 1-Click 이벤트	13
AWS IoT 1-Click 이벤트	14
AWS IoT 1-Click Health 이벤트	15
디바이스 메소드	15
CloudWatch 지표를 사용한 모니터링	17
AWS AWS CloudTrail 을 사용하여 AWS IoT 1-Click API 호출 로깅	19
CloudTrail 에서 AWS IoT 1-Click 정보	19
예: AWS IoT 1-Click 로그 파일 항목	21
AWS CloudFormation 통합	23
AWS IoT 1-Click ck에 대한 인증 및 액세스 제어	24
AWS IoT 1-Click 리소스 및 운영	24
AWS IoT 1-Click ck에 대한 자격 증명 기반 정책 (IAM 정책) 사용	24
AWS IoT 1-Click에 대한 AWS 관리형 (미리 정의된) 정책	25
AWS IoT 1-Click 리소스 태그 지정	29
태그 기본 사항	29
태그 제한	30
AWS IoT Enterprise Button 사용자 안내서	31
AWS IoT 1-Click에서 사용	33
AWS IoT 1-Click 부록	51
AWS IoT 1-Click 지원되는 디바이스	51
AWS IoT 1-Click 서비스 제한	53
문서 기록	54

AWS 용어집	55
.....	lvi

AWS IoT 1-Click이란 무엇입니까?

AWS IoT 1-Click 사용하면 기업 고객이 디바이스를 제조하거나, 펌웨어를 작성하거나, 안전한 연결을 위해 구성할 필요 없이 간단한 IoT 디바이스를 해당 워크플로우에 쉽게 통합할 수 있습니다. AWS의 제조 파트너는 바로 AWS IoT에 안전하게 연결할 수 있는 디바이스를 생성합니다. 이러한 장치는 트리거할 수 있습니다. [AWS Lambda](#) Java, Python 및 C #와 같은 언어로 작성되는 함수 Lambda 함수는 비즈니스 로직을 자체적으로 구현하거나 AWS 클라우드 또는 온프레미스의 다른 위치에서 작업을 트리거할 수 있습니다.

AWS IoT 1-Click은 디바이스 하드웨어 및 펌웨어와 관련된 세부 정보를 최대한 추상화하여 고객을 위해 사물 인터넷을 단순화하는 것을 목표로 합니다. 이렇게 하면 AWS IoT 1-Click 디바이스를 AWS 클라우드에서 호스팅된 소프트웨어 구성 요소로 볼 수 있습니다. 다른 소프트웨어 구성 요소와 마찬가지로 이러한 디바이스도 명확하게 정의된 인터페이스를 준수합니다. AWS IoT 1-Click에는 디바이스 유형별로 인터페이스가 정의됩니다. 이러한 인터페이스를 사용하여 애플리케이션을 빌드하고 기준을 설정할 수 있습니다.

AWS IoT 1-Click을 사용하면 함수, 위치 또는 다른 기준으로 디바이스를 그룹화할 수 있습니다. 이러한 논리적 디바이스 그룹을 "프로젝트" AWS IoT 1-Click에서 프로젝트를 사용하여 원하는 작업에 대해 Lambda 함수에 디바이스 그룹을 연결할 수 있습니다.

프로젝트에는 사용되는 디바이스 유형, 호출되는 Lambda 함수 및 위치나 함수에 대한 컨텍스트 데이터와 같이 이러한 디바이스에 대해 정의되는 선택적 속성을 지정하는 템플릿이 포함되어 있습니다.

프로젝트가 작성되고 템플릿이 정의되면 프로젝트 내에 배치를 추가할 수 있습니다. 각 배치는 템플릿을 따르고 해당 배치의 특정 위치 또는 기능에 적합한 일련 번호와 속성 값 (키/값 쌍)으로 실제 장치를 지정합니다.

AWS IoT 1-Click 구성 요소

Claim

AWS IoT 1-Click 콘솔, AWS IoT 원클릭 모바일 앱 또는 AWS IoT 1-Click API를 사용하여 AWS 계정에 AWS IoT 1-Click 디바이스를 연결하는 프로세스를 나타냅니다.

클레임 코드

여러 개의 AT&T LTE-M 버튼을 한 번에(즉, 대량) 신청하는 데 사용되는 값입니다. 디바이스 ID를 사용하여 디바이스를 신청할 수도 있습니다. 디바이스 ID 항목을 참조하십시오.

Device

AWS IoT Enterprise Button 또는 AT&T LTE-M 버튼과 같은 물리적 디바이스입니다.

디바이스 속성

키값 페어의 형식으로 특정 디바이스에 연결된 사용자 지정 데이터 또는 기본값입니다. 기본 속성은 배치에서 파생됩니다. 배치 항목을 참조하십시오.

디바이스 ID

모든 장치에는 장치 일련 번호 (DSN) 와 같은 장치 ID가 있습니다. 디바이스 ID는 AWS IoT 1-Click 에서 AWS IoT 1-Click에 디바이스를 등록하는 데 사용할 수 있습니다. 신청 코드는 디바이스 ID와 동일하지 않습니다. 신청 코드 항목을 참조하십시오.

Placement

디바이스를 나타내는 하나 이상의 템플릿 그룹입니다(예: 회의실에 두 개의 템플릿 기반 버튼이 있음). 배치를 채우려면 AWS IoT 1-Click 콘솔 또는 AWS IoT 1-Click 모바일 앱을 사용하여 템플릿화된 디바이스를 선택합니다.

배치 이름

배치 이름입니다. 지리적 위치 또는 객체 ID(예: Room 217, North Dumpster 또는 Container 314)가 포함되는 경우가 많습니다.

프로젝트

0개 이상의 배치로 구성된 명명된 그룹입니다(템플릿 기반 디바이스 포함).

프로젝트 이름

"Meeting Room Satisfaction" 또는 "Charter Container Pickup"과 같이 배치의 그룹에 대한 설명이 포함된 이름입니다.

Template

장치 그룹에 대한 기본 동작 및 기본 특성을 제공하는 데 사용됩니다. 물리적 디바이스에서는 특정 템플릿을 사용하여 해당 템플릿의 속성 (해당 Lambda 함수 및 기본 디바이스 속성) 을 상속합니다. 템플릿은 배치에서 디바이스의 클래스에 대한 동작 및 기본 속성을 정의합니다. 프로젝트에는 둘 이상의 템플릿이 있을 수 있습니다.

클레임 취소

AWS IoT 1-Click 디바이스의 연결을 해제하는 프로세스입니다. 예를 들어 AWS IoT 1-Click 디바이스를 대여하려는 사용자는 새 사용자가 본인의 AWS 계정에 디바이스를 연결할 수 있도록 AWS 계정에서 디바이스의 연결을 해제해야 합니다.

AWS IoT 1-Click 작동 방식

AWS IoT 1-Click 워크플로는 다음과 같습니다.

1. 지원되는 장치 세트 중에서 선택합니다.
2. AWS Lambda 함수를 디바이스와 연결하여 작업을 트리거합니다. 자체 Lambda 함수 중 하나를 사용하거나 서비스에서 제공하는 사전 정의된 함수 중 하나를 사용할 수 있습니다.
3. 디바이스를 물리적으로 배포하고 AWS IoT 1-Click 콘솔, AWS IoT 1-Click 모바일 앱 또는 AWS IoT 1-Click API를 사용하여 디바이스를 활성화합니다.
4. AWS IoT 1-Click 보고서를 사용하거나 자체 보고서를 작성하여 디바이스 상태 및 사용량에 대한 정보를 얻습니다.

AWS IoT 1-Click 디바이스

AWS IoT 1-Click 지원 디바이스

- 미리 만들어져 있습니다. 고객은 이를 설계하거나 제조할 필요가 없습니다.
- AWS IoT 1-Click 계정에 추가할 수 있습니다. [클레임 기능](#).
- 제조 시 인증서로 사전에 프로비저닝되고 AWS IoT (IoT) 에 안전하게 연결되도록 구성됩니다. AWS IoT 1-Click 디바이스용 인증서를 설치하는 데 시간을 들일 필요가 없습니다.
- AWS IoT 1-Click 디바이스 유형 AWS IoT 1-Click Click에서 정의한 표준 형식으로 이벤트를 방출합니다. 예를 들어, AWS IoT 1-Click 디바이스 button 유형의 이벤트 형식은 제조업체와 관계없이 동일한 이벤트 형식을 보유합니다.
- 디바이스 유형과 제품 유형이 있습니다. 디바이스 유형은 디바이스가 방출하는 이벤트 형식과 디바이스가 지원하는 디바이스 메서드를 나타냅니다. 자세한 내용은 [AWS IoT 1-Click 프로그래밍 모델](#) 섹션을 참조하세요. 제품 유형은 제조업체 및 브랜드 사용에 대한 세부 정보를 제공합니다. 예를 들어 디바이스 유형이 button인 경우 제품 유형은 AT&T LTE-M 버튼일 수 있습니다.

Important

AWS IoT 1-Click 지원 디바이스는 공장에서 특정 디바이스에 연결되도록 구성됩니다. [AWS 리전](#). 이를 디바이스 리전이라고 합니다. 이 디바이스 연결 디바이스 리전은 추가 사용자 입력 없이 디바이스가 AWS IoT 에 안전하게 연결되도록 하는 데 필요합니다. 이러한 이유로 디바이스 리전을 변경할 수 없습니다.

AWS IoT 1-Click 디바이스에서 방출된 이벤트는 항상 사전 구성된 디바이스 리전을 통해 라우팅되므로 동일한 AWS 리전에서 디바이스 관련 Amazon CloudWatch Logs 및 AWS CloudTrail 측정치에 액세스할 수 있습니다. 디바이스 리전은 활성화된 디바이스가 청구되는 위치이기도 합니다.

배치, 템플릿 및 프로젝트 데이터는 계정과 연결된 AWS 리전에 저장됩니다. 이 리전은 디바이스 리전과 다를 수 있습니다.

구매 방법 및 구매 방법을 포함하여 AWS IoT 1-Click 지원 디바이스에 대한 자세한 내용은 [클레임](#) 해당 항목에 대한 자세한 내용은 [AWS IoT 1-Click 부록](#).

디바이스 신청

AWS IoT 1-Click 디바이스가 공장에서 출시될 때 해당 디바이스는 AWS IoT 1-Click 디바이스와 연결되어 있지 않습니다. 고객은 본인의 계정에서 디바이스를 사용하기 위해 신청 프로세스를 진행해야 합니다. 디바이스를 신청하는 방법에는 다음의 두 가지 방법이 있습니다.

- **클레임 코드 사용:** 당사는 클레임 코드를 받는 경우 (형식 C에서-xxxxxx) 을 구매한 후 AWS IoT 1-Click 콘솔 또는 AWS IoT 1-Click 모바일 앱에 입력하여 단일 주문과 관련된 디바이스를 신청할 수 있습니다. AWS IoT Enterprise Button 포함한 모든 디바이스에서 신청 코드를 사용하여 신청할 수 있는 것은 아닙니다.
- **장치 ID 사용:** AWS IoT 1-Click 콘솔 또는 AWS IoT 1-Click 모바일 앱을 통해 디바이스 ID (DSN이라고도 하는 디바이스 일련 번호) 를 사용하여 디바이스를 신청할 수 있습니다. 디바이스 ID를 사용하여 AWS IoT 1-Click 디바이스를 신청할 수 있습니다.

디바이스 신청 방법에 대한 자세한 내용은 [AWS IoT 1-Click 부록](#) 및 [디바이스 할당](#) 단원을 참조하십시오.

프로젝트, 템플릿 및 배치

기능, 위치별로 또는 다른 기준으로 디바이스를 구성할 수 있습니다. 이 논리적 디바이스 그룹을 프로젝트라고 합니다. 프로젝트를 사용하여 Lambda 함수와 디바이스 그룹을 연결할 수 있습니다.

프로젝트에는 사용되는 디바이스 유형, 호출되는 Lambda 함수 및 위치나 함수와 같이 컨텍스트 데이터를 보유하는 속성 이름을 지정하는 템플릿이 포함되어 있습니다.

프로젝트가 생성되고 템플릿이 정의되면 프로젝트에 배치를 추가할 수 있습니다. 배치는 템플릿을 따르며 해당 배치의 위치 또는 함수에 적합한 일련 번호 및 속성 값으로 디바이스를 지정합니다.

다음은 프로젝트 및 배치의 사용을 보여주는 몇 가지 예입니다.

예 1:

SalesPersonNotification 프로젝트에서는 10명의 고객이 버튼을 눌러 영업 담당자에게 연락할 수 있도록 버튼을 받습니다. 고객마다 하나씩 10개의 배치가 있습니다. 각 배치에는 CustomerName(예: Mr. Jones), SalesPersonPhoneNumber(예: 1-555-555-1234) 및 버튼 일련 번호(예: G030PM12345678) 값이 있고 디바이스 템플릿 NotificationButton이 포함되어 있습니다. 배치별로 CustomerName 및 SalesPersonPhoneNumber 속성이 정의됩니다. 고객이 버튼을 클릭하면 AWS IoT 1-Click Click에서 SendSMSLambda를 사용하여 CustomerName 및 SalesPersonPhoneNumber 해당 버튼과 관련된 값. 해당 값에 따라 SMS가 전송됩니다.

- 배치 템플릿:
 - 각 고객은 하나의 버튼을 사용하여 영업 담당자에게 알리기 때문에 NotificationButton가 생성됩니다.
 - 디바이스 템플릿 (배치에 포함됨) 은 NotificationButton를 클릭하면 SendSMSLambdaLambda 함수
 - 호출된 속성 CustomerName 및 SalesPersonPhoneNumber 각 배치에 대해 정의됩니다.
- 배치: 고객마다 1개씩, 10개의 배치가 생성됩니다. 각 배치에는 고유한 CustomerName(예: "Mr. Jones"), SalesPersonPhoneNumber(예: 1-555-555-1234) 및 버튼 일련 번호(예: G030PM12345678) 값이 있습니다.
- 작업: 고객이 버튼을 클릭하면 AWS IoT 1-Click Click에서 SendSMSLambda를 사용하여 CustomerName 및 SalesPersonPhoneNumber 값을 입력하면 해당 버튼에 연결된 SMS가 전송됩니다.

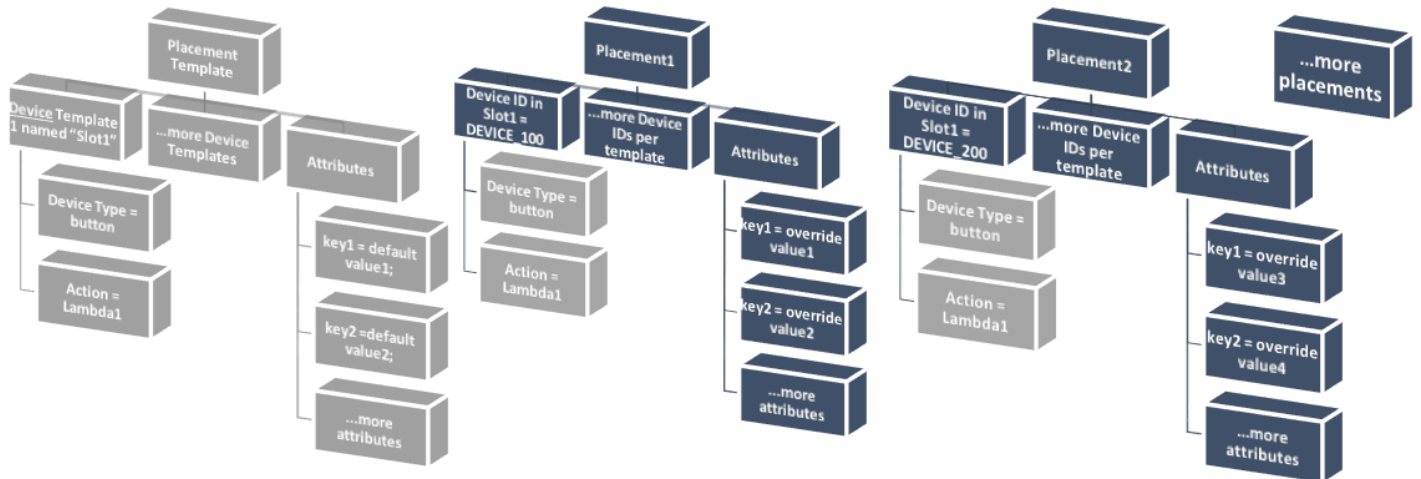
예 2:

MeetingRoomFeedback 프로젝트에서는 50개의 회의실에서 만족을 나타내는 엄지 손가락 위로 버튼과 불만족을 나타내는 엄지 손가락 아래로 버튼을 통해 사용자 만족도를 추적합니다. 두 개의 디바이스 템플릿 ThumbsUp 및 ThumbsDown이 있으며, 엄지 손가락 위로 버튼을 클릭하면 PostiveFeedbackLambda 함수가 호출되고 엄지 손가락 아래로 버튼을 클릭하면 NegativeFeedbackLambda가 호출됩니다. AMeetingRoomNumber 각 배치의 회의실 번호를 보유하도록 속성이 정의됩니다. 회의실당 하나씩 50개의 디바이스 배치가 생성됩니다. 각 배치에는 회의실 번호(예: 1001)로 설정된 MeetingRoomNumber 키와 고유한 일련 번호(예: G030PM12345678 및 G030PM23456789)로 식별되는 두 개의 버튼이 있습니다. 회의실에서 버튼을 클릭하면 AWS IoT 1-

Click Click에서PositiveFeedbackLambda함수 또는NegativeFeedbackLambda함수를 사용하여MeetingRoomNumberUSD 상당. 그런 다음 피드백이 처리되고 표로 작성될 수 있습니다.

- 프로젝트 이름: MeetingRoomFeedback
- 배치 템플릿:
 - 각 룸에 두 개의 버튼이 있으므로 두 개의 장치 템플릿이 만들어지며 각 각ThumbsUp및ThumbsDown.
 - 장치 템플릿은 다음을 지정합니다.ThumbsUp버튼을 호출합니다.PostiveFeedbackLambda를 클릭하면 해당ThumbsDown버튼을 호출합니다.NegativeFeedbackLambda를 클릭합니다.
 - 라는 속성MeetingRoomNumber각 배치의 회의실 번호를 보유하도록 정의됩니다.
- 배치: 방마다 배치 1개씩, 50개의 디바이스 배치가 생성됩니다. 각 배치에 는MeetingRoomNumber키를 특정 회의실 번호 쌍 (예: 1001) 로 설정하고 고유한 일련 번호 (예: G030PM12345678 및 G030PM23456789) 로 식별되는 두 개의 버튼으로 구성됩니다.
- 작업: 회의실에서 버튼을 클릭하면 AWS IoT 1-Click Click에서PositiveFeedbackLambda함수 또는NegativeFeedbackLambda함수를 사용하여MeetingRoomNumber값을 사용하여 피드백이 처리되고 표로 작성될 수 있습니다.

다음 다이어그램에서는 이러한 개념을 보여 줍니다.



자세한 내용은 [AWS IoT 1-Click 콘솔 시작하기](#) 섹션을 참조하세요.

AWS IoT 1-Click 콘솔 시작하기

다음 주제에서는 일반적인 AWS IoT 1-Click 작업을 수행하는 방법을 설명합니다.

주제

- [디바이스 할당](#)
- [프로젝트 만들기](#)

디바이스 할당

다음 절차에서는 하나 이상의 AWS IoT 1-Click 지원 디바이스를 신청하는 방법을 보여줍니다.

1. AWS 계정에 로그인. AWS 계정이 없는 경우 <https://aws.amazon.com/>를 선택하고 AWS 계정 생성을 방문하여 온라인 지시 사항을 따릅니다.
2. AWS 관리 콘솔에서 “1-Click”을 검색한 다음 AWS IoT 1-Click 선택합니다.
3. 하나 이상의 AWS IoT 1-Click 모바일 앱을 iOS 또는 Android용 AWS IoT 1-Click 모바일 앱을 설치하고 해당 버튼을 로컬 Wi-Fi 네트워크에 연결합니다. AWS IoT 1-Click 모바일 앱은 온보딩 AWS IoT 1-Click 콘솔의 페이지를 참조하십시오. LTE-M Button의 경우 셀룰러 네트워크를 사용하므로 이 단계가 필요하지 않습니다.
4. 선택 온보딩을 선택한 다음 를 선택합니다. 디바이스.
5. 하나 이상의 를 입력합니다. [디바이스 ID](#)(예: 장치 일련 번호) 또는 [클레임 코드](#)을 쉼표로 구분한 다음 Claim. 경우 Claim 버튼을 사용할 수 없는 경우 입력한 모든 값을 다시 확인합니다.
6. 디바이스에서 해당 버튼을 누른 다음 Done. 알려진 모든 디바이스 목록이 표시되어야 합니다.

프로젝트 만들기

다음 절차에서는 AWS IoT 1-Click 지원 디바이스에 대한 AWS IoT 1-Click 프로젝트를 생성하는 방법을 보여줍니다.

1. AWS 계정에 로그인하고 AWS IoT 1-Click 콘솔을 엽니다.
2. 선택 온보딩을 선택한 다음 를 선택합니다. 프로젝트 생성.
3. 프로젝트의 이름과 설명 (선택 사항) 을 입력한 다음 다음.
4. 배치에 대해 하나 이상의 템플릿을 정의하려면 장치 템플릿 프로그래밍을 선택하고 Start.

5. 모든 버튼 디바이스에 대한 템플릿을 정의하려면 모든 버튼 유형.
6. 디바이스 템플릿 이름에 템플릿을 설명하는 이름을 입력합니다. 작업에서 SMS 보내기 또는 이메일 보내기를 선택합니다. 다음을 수행할 수 있습니다. Lambda 함수를 사용하여 사용자 지정 작업 옵션을 선택하고 자신의 Lambda 함수 중 하나를 선택하십시오. 선택한 항목에 따라 전화 번호, 이메일 주소 또는 Lambda 함수 이름을 입력합니다. Lambda 함수를 만드는 방법에 대한 자세한 내용은 단원을 참조하십시오. [AWS Lambda 개발자 안내서](#).
7. 언더다른 디바이스 템플릿 추가 (배치당 여러 디바이스가 필요한 경우) 선택할 Add.
8. 속성 키/값 페어를 입력합니다. 필요한 경우 추가 키/값 페어를 입력할 수 있습니다.
9. 프로젝트 만들기를 선택합니다.

다음 섹션은 [예: 회의실 만족도 프로젝트](#)에서는 AWS IoT 1-Click 콘솔을 사용하여 프로젝트를 생성하는 방법에 대한 실제 예를 보여줍니다.

예: 회의실 만족도 프로젝트

다음 예는 AWS IoT 1-Click 개념을 이해하는 데 도움이 될 수 있습니다.

- 50개의 회의실 (및 관련 AV 장비) 의 만족도를 추적하는 프로젝트가 만들어지고 MeetingRoomSat.
- 각 회의실에는 두 개의 장치 (버튼) 가 표시됩니다. 하나는 물리적으로 “만족”으로 표시되고 다른 하나는 “불만족”으로 표시됩니다. 룸당 두 개의 버튼이 있기 때문에 두 개의 템플릿이 만들어집니다. Satisfied라는 이름을 지정하며 다른 하나는 Unsatisfied.
- 이 Satisfied 템플릿은 Lambda 함수를 호출하도록 구성됩니다. SatLambda.
- 이 Unsatisfied 템플릿은 Lambda 함수를 호출하도록 구성됩니다. UnsatLambda.
- 이 두 템플릿 모두에서 속성 (키/값 쌍) MeetingRoomNum(키) 가 만들어지며 그 값은 TBD(TBD값은 두 버튼이 실제로 룸에 배치될 때 룸 번호로 변경됩니다).
- 각 방마다 하나씩 50개의 배치가 생성됩니다. 각 배치에는 연관된 두 개의 템플릿 (예: Satisfied 및 Unsatisfied).
- 두 개의 버튼은 물리적 라벨이 부착되어 방에 배치됩니다. 그런 다음 AWS IoT 1-Click 모바일 앱 또는 AWS IoT 1-Click 콘솔과 버튼의 일련 번호를 통해 “Satisfied” 및 “Unsatisfied”로 표시된 버튼이 50개의 배치 중 하나와 연결됩니다. 이 프로세스는 나머지 배치가 모두 배치될 때까지 계속됩니다.
- 회의실에서 방 버튼을 클릭하면 AWS IoT 1-Click 콘솔에서 SatLambda 또는 UnsatLambda 함수를 사용하여 MeetingRoomNum 값을 사용하여 피드백이 처리되고 클라우드에 저장됩니다.
- 나중에 다른 템플릿을 프로젝트에 추가하여 50개의 기존 배치에 각 욕실에 더 많은 수건이나 다른 세면 도구가 필요하다는 것을 나타내는 새 버튼의 슬롯이 포함될 수 있습니다.

다음은 AWS IoT 1-Click 콘솔을 사용하여 사무실용 건물 (사무실용 건물 그룹의 일부) 에서 회의실 만족도를 모니터링하기 위한 프로젝트를 생성하는 예를 제공합니다.

오디오/비디오 장비를 비롯한 회의실 만족도를 모니터링하기 위해 각 회의실에 두 개의 AWS IoT Enterprise 버튼을 배치했습니다. 하나는 "Satisfied"로 레이블이 지정되며 다른 하나는 "Unsatisfied"로 레이블이 지정되며 다른 하나는 "Unsatisfied"로 레이블이 지정되며 다른 하나는 "Un 이 는 파일럿 프로젝트로, 캠퍼스의 다른 건물에서 회의실의 고객 만족도를 높이는 데 프로젝트 결과를 사용할 수 있습니다.

회의가 완료된 후, 참가자는 회의실 및 장비에 대한 전반적인 만족도를 기록하기 위해 "Satisfied" 또는 "Unsatisfied" 버튼을 누르도록 요청을 받습니다. 이 데이터는 회의실에서 제대로 작동하지 않는 A/V 장비나 기타 문제를 식별하는 데 사용됩니다.

AWS IoT 1-Click 콘솔을 사용하여 이 프로젝트를 설정할 수 있습니다.

1. AWS IoT 1-Click 콘솔에서 프로젝트 생성.
2. 프로젝트 이름에 **MeetingRoomSatisfaction**을 입력합니다. 프로젝트 설명에 를 입력하고 **Project used to track customer meeting room satisfaction, including A/V equipment.** 선택다음.
3. 언더장치 템플릿 프로그래밍 선택할 Start를 선택한 다음 를 선택합니다. 모든 버튼 유형.
4. 디바이스 템플릿 이름에 **Satisfied**를 입력합니다. 이는 "Satisfied"라는 레이블이 지정된 모든 버튼에 사용되는 템플릿입니다. 작업에서 이메일 보내기를 선택합니다.

Note

회의실 만족 파일럿이 성공하면 작업에서 Lambda 함수를 사용한 사용자 지정 작업을 선택할 수 있습니다. 이 사용자 지정 Lambda 함수는 이메일을 보내거나 "Satisfaction" 버튼 데이터를 나중에 분석하도록 Amazon DynamoDB 테이블에 저장할 수 있습니다. Lambda 함수를 만드는 방법에 대한 자세한 내용은 단원을 참조하십시오. [AWS Lambda 개발자 안내서](#).

5. 언더다른 디바이스 템플릿 추가 (배치당 여러 디바이스가 필요한 경우) 선택할 Add를 선택한 다음 를 선택합니다. 모든 버튼 유형. 디바이스 템플릿 이름에 **Unsatisfied**를 입력합니다. 이는 "Unsatisfied"라는 레이블이 지정된 모든 버튼에 사용되는 템플릿입니다. 작업에서 이메일 보내기를 선택합니다.
6. 필요한 이메일 기본값에 이메일 주소를 입력합니다. 필요한 제목 기본값에 **Meeting Room Feedback**을 입력합니다. 필요한 본문 기본값에 **Either positive or negative meeting room feedback has been provided.**를 입력합니다.

7. 속성 키에 **Building**을 입력합니다. 기본값에 **Headquarters**를 입력합니다. 회의실 만족도 파일럿이 회사의 본사 건물에서 진행되고 있습니다. 파일럿이 성공하면 회사의 다른 건물에 배치됩니다. 따라서 회의실 장치가 어떤 건물에서 정보를 제공하는지 아는 것이 중요합니다.
8. 두 번째 키/값 페어 행에서 속성 키에 **Room**을 입력합니다. 기본값에 **TBD**를 입력합니다. 이 **TBD**값이 회의실 번호로 변경됩니다 (AWS IoT 1-Click 모바일 앱 또는 AWS IoT 1-Click 콘솔 사용).
9. 프로젝트 만들기를 선택합니다.

AWS IoT 1-Click 모바일 앱을 사용하여 “Satisfied” 버튼이 회의실에 배치되면만족템플릿이 해당 템플릿과 연관되고 **TBD**값이 회의실 번호로 바뀝니다. “Unsatisfied” 버튼이 회의실에 배치된 경우에도 마찬가지입니다.

AWS IoT 1-Click 모바일 앱

AWS IoT 1-Click 모바일 앱을 사용하면 다음을 수행할 수 있습니다.

- AWS IoT 1-Click 콘솔과 유사한 사용자 인터페이스를 사용하여 현장에서 AWS IoT 1-Click 디바이스를 편리하게 구성하고 모니터링.
- AWS IoT 1-Click 디바이스 (예: AWS IoT Enterprise Button) 에 사용되는 Wi-Fi 자격 증명을 구성합니다.

AWS IoT 1-Click 모바일 앱은 iPhone 및 Android 모바일 디바이스에서 사용할 수 있습니다. 앱을 다운로드하려면 [앱 스토어](#) 또는 [Google Play](#)을 검색하여 AWS IoT 1-Click 검색합니다.

AWS IoT 1-Click 프로그래밍 모델

AWS IoT 1-Click 디바이스를 사용하여 애플리케이션을 빌드하기 위해 프로그래머는 [AWS IoT 1-Click 디바이스 API](#) 및 [AWS IoT 1-Click 프로젝트 API](#). 디바이스 API는 AWS IoT 1-Click 디바이스 구성 요소와 상호 작용하며, 디바이스에서 발생하는 이벤트를 처리합니다. 이러한 이벤트에는 장치 활성화 및 비활성화, 이벤트 형식 및 트리거되는 동작 (Lambda 함수) 정의 등이 포함됩니다. 디바이스 API는 제조업체가 디바이스를 등록한 리전에 있는 AWS 구성 요소와 긴밀하게 결합됩니다. 이것이 바로 그 이유입니다. [AWS 디바이스 리전](#)은 고객이 디바이스를 사용하고 있는 리전과 다를 수 있습니다. 프로젝트 API는 AWS IoT 1-Click 프로젝트 서비스와 상호 작용하며, AWS IoT 1-Click 디바이스를 함께 관리하는 데 사용되며, 이를 통해 다음을 수행할 수 있습니다.

- 디바이스를 프로젝트로 그룹화.
- 프로젝트의 모든 디바이스에 대한 작업을 설정하는 데 사용되는 템플릿 생성.
- 프로젝트와 관련된 컨텍스트 데이터를 저장하는 속성 정의.

AWS IoT 1-Click 프로그래밍 모델을 사용하여 디바이스 API를 사용하여 개별 디바이스를 프로그래밍할 수 있습니다. 이 경우 AWS IoT 1-Click 디바이스 유형을 사용합니다. API는 표준 이벤트 형식과 해당 유형의 모든 장치에 대한 프로그래밍 인터페이스를 형성하는 메서드 목록을 정의합니다. 지정된 데이터 유형에 관련된 메서드를 호출하기 위해 프로그래머는 [InvokeDeviceMethod API](#)를 사용하고 디바이스 메서드를 파라미터로 지정할 수 있습니다.

예를 들어, 디바이스 유형 “버튼”이 있는 모든 AWS IoT 1-Click 디바이스는 클릭과 연관된 이벤트를 방출하며, 디바이스를 클릭할 때 호출되는 콜백 함수를 설정하는 메서드를 포함하고 있습니다. 버튼 인터페이스에 대한 자세한 내용은 [디바이스 유형별 인터페이스](#)를 참조하십시오. 다음은 이 콜백 함수를 설정하는 코드입니다.

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



```

        .withMethodName("setOnClickCallback"))
        .withDeviceMethodParameters(methodParameters);

```

프로젝트 API를 사용하여 여러 장치를 프로그래밍할 수 있습니다. API를 사용하여 먼저 각 배치의 장치 템플릿 및 속성을 포함하여 각 배치의 모양을 정의합니다. 이렇게 한 후에 특정 디바이스 ID로 배치를 생성합니다. 각 배치는 동일한 템플릿을 따릅니다. 이렇게 하기 위한 샘플 코드는 다음과 같습니다.

```

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

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

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

```

AWS IoT 1-Click 이벤트

AWS IoT 1-Click에서 콜백을 등록하여 디바이스 이벤트를 구독할 수 있습니다. AWS IoT 1-Click 고객이 소유하고 구현한 AWS Lambda 함수를 예로 들 수 있습니다. 이 콜백은 사용할 수 있는 이벤트가 있을 때마다 호출됩니다. 이벤트 및 해당 페이로드에 대한 자세한 내용은 [AWS IoT 1-Click 이벤트](#) 및 [AWS IoT 1-Click Health 이벤트](#) 단원을 참조하십시오.

AWS IoT 1-Click 이벤트

button 유형의 디바이스는 클릭할 때마다 클릭 이벤트를 게시합니다. 다음 방법으로 이 이벤트를 구독할 수 있습니다.

- 디바이스에서 디바이스 SetOnClickCallback 메서드 호출.
- 이전 프로젝트 코드 생성기 예제와 같이 관련 프로젝트를 적절하게 구성.

다음 예제에서는 디바이스에 관련 배치가 있는 경우에만 placementInfo 섹션이 있습니다. 자세한 내용은 [프로젝트, 템플릿 및 배치](#) 섹션을 참조하세요.

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

AWS IoT 1-Click Health 이벤트

디바이스는 AWS IoT 1-Click 서비스에서 계산한 상태 파라미터를 기반으로 상태 이벤트를 게시하지만 해당 임계값은 사용자가 설정합니다. 다음 예제는 남은 수명이 10%인 G030PMXXXXXXXXXXXX 디바이스에 대한 상태 이벤트의 JSON 페이로드를 나타냅니다("remainingLifeLowerThan":10 키/값 페어 참고).

```
{
  "deviceEvent": {
    "deviceHealthMonitor": {
      "condition": {
        "remainingLifeLowerThan": 10
      }
    }
  },
  "deviceInfo": {
    "attributes": {
      "key2": "value2",
      "key1": "value1",
      "projectRegion": "us-west-2"
    },
    "type": "button",
    "deviceId": "G030PMXXXXXXXXXXXX",
    "remainingLife": 5.4
  }
}
```

디바이스 메소드

AWS IoT 1-Click 디바이스 메서드는 다음 표에 나와 있는 것처럼 특정 디바이스 유형의 디바이스에서 지원하는 API입니다. 모든 디바이스에서 지원하는 디바이스 메서드의 전체 목록은 [GetDeviceMethods](#) 호출을 통해 검색할 수 있습니다.

디바이스 유형	메소드 이름	설명
device	getDeviceHealthParameters	다음과 같은 장치의 상태 매개 변수를 가져옵니다.remainingLife .

디바이스 유형	메소드 이름	설명
device	setDeviceHealthMonitorCallback	장치 상태 매개 변수가 임계값 미만일 때 호출될 콜백을 설정합니다.
device	getDeviceHealthMonitorCallback	상태 매개 변수가 임계 값 미만일 때 호출되는 구성된 콜백을 가져옵니다.
button	setOnClickCallback	버튼을 클릭했을 때 호출되는 콜백을 설정합니다.
button	getOnClickCallback	버튼을 클릭할 때 호출되는 구성된 콜백을 가져옵니다.

Amazon CloudWatch AWS IoT 1-Click 사용한 모니터링

AWS IoT 1-Click 클에서 사용자를 대신하여 디바이스를 자동으로 모니터링하고 [Amazon CloudWatch](#). 이러한 메트릭은 제조업체가 장치를 등록한 장치 지역에서 보고됩니다. 디바이스 리전에 대한 자세한 내용은 단원을 참조하십시오. [AWS IoT 1-Click 클에서 작동하는 방식](#). Amazon CloudWatch 대시보드에서 측정치를 확인할 수 있습니다. [iot1클릭네임스페이스](#)

Amazon CloudWatch Events AWS 서비스를 자동화하고 애플리케이션 가용성 문제나 리소스 변경 같은 시스템 이벤트에 자동으로 응답할 수 있습니다. AWS 서비스의 이벤트는 거의 실시간으로 CloudWatch 이벤트로 전송됩니다. 원하는 이벤트만 표시하도록 간단한 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 트리거할 수 있는 작업은 다음과 같습니다.

- AWS Lambda 함수 호출
- Amazon EC2 실행 명령 호출
- Amazon Kinesis Data Streams 릴레이
- AWS Step Functions 상태 시스템 활성화
- Amazon SNS 주제 또는 AWS SMS 대기열에 알림

AWS IoT 1-Click 클에서 다음 측정치를 추적하고 보고합니다.

- TotalEvents는 디바이스에 의해 게시된 이벤트 수를 추적합니다. 디바이스 이벤트, 프로젝트, 디바이스 유형 또는 제품 유형에 따라 이 측정치를 보고 그래프로 표시할 수 있습니다.
- RemainingLife는 장치에 대한 남은 수명의 대략적인 비율을 나타냅니다. AWS IoT 1-Click 클에서 디바이스 제조업체의 등급에 따라 이 수치를 보고합니다. 예를 들어, 버튼이 약 2000번의 클릭에 대해 지속되도록 설계되고 500번의 클릭이 기록된 경우 RemainingLife 값은 75%로 보고됩니다. RemainingLife 측정치는 프로젝트, 디바이스 유형 또는 제품 유형에 따라 보고 그래프로 표시할 수 있습니다. 고객은 RemainingLife 측정치를 사용하여 디바이스가 특정 임계값 밑으로 떨어질 때 경보를 트리거하도록 설정할 수 있습니다. 그런 다음 고객은 RemainingLife를 사용하여 디바이스 GetDeviceHealthParameters 낮은 장치를 식별 하는 방법 RemainingLife 값
- CallbackInvocationErrors는 디바이스가 이벤트를 방출할 때 콜백(Lambda 함수) 호출의 실패를 추적합니다. CallbackInvocationErrors 측정치는 호출된 콜백(콜백으로 설정된 Lambda 함수 ARN) 또는 프로젝트별로 보고 그래프로 표시할 수 있습니다. 고객은 경보를 설정할 수 있습니다. CallbackInvocationErrors 메트릭을 사용하여 AWS IoT 1-Click 클이 디바이스에서 구성된 Lambda 함수로 이벤트를 라우팅할 수 없을 때 알림을 받을 수 있습니다.

자세한 내용은 [Amazon CloudWatch Events 사용 설명서](#)를 참조하십시오.

AWS CloudTrail 을 사용하여 AWS IoT 1-Click API 호출 로깅

AWS IoT 1-Click은 AWS 1-Click에서 사용자, 역할, AWS 서비스가 수행한 작업의 레코드를 제공하는 AWS CloudTrail 서비스와 통합됩니다. CloudTrail은 AWS IoT 1-Click에 사용되는 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 AWS IoT 1-Click 콘솔에서 수행된 호출과 AWS IoT 1-Click API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 AWS IoT 1-Click에 대한 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS IoT 1-Click에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

구성 및 활성화 방법 등 CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용 설명서](#)를 참조하세요.

CloudTrail 에서 AWS IoT 1-Click 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. AWS IoT 1-Click에서 지원되는 이벤트 CloudTrail 발생하면, 해당 활동이 이벤트 기록. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 [CloudTrail 이벤트 기록에서 이벤트 보기](#)를 참조하십시오.

AWS IoT 1-Click에 사용되는 이벤트를 비롯하여 AWS 계정의 이벤트 기록을 보유하려면 추적을 생성하십시오. 추적은 CloudTrail이 Amazon S3 버킷으로 로그 파일을 전송할 수 있게 해줍니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 섹션을 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)

AWS IoT 1-Click [디바이스 API](#)에서는 다음과 같은 작업을 CloudTrail 로그 파일에 이벤트로 로깅할 수 있습니다.

- [ListDevices](#)

- [설명장치](#)
- [장치 메서드 가져](#)
- [업데이트장치상태](#)
- [호출 장치메소드](#)

AWS IoT 1-Click [프로젝트 API](#)에서는 다음과 같은 작업을 CloudTrail 로그 파일에 이벤트로 로깅할 수 있습니다.

- [CreateProject](#)
- [업데이트프로젝트](#)
- [DescribeProject](#)
- [ListProjects](#)
- [프로젝트 삭제](#)
- [CreatePlacement](#)
- [업데이트배치](#)
- [설명 배치](#)
- [목록 배치](#)
- [삭제 배치](#)
- [연관장치배치와 연관됨](#)
- [DisassociateDeviceFromPlacement](#)
- [장치 가져 오기삽입](#)

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 AWS Identity and Access Management(IAM) 사용자 자격 증명으로 했는지 여부
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하십시오.

예: AWS IoT 1-Click 로그 파일 항목

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 어떤 소스로부터의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예에 DescribeDevice 작업을 보여 주는 CloudTrail 로그 항목이 나와 있습니다.

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

다음 예에 CreateProject 작업을 보여 주는 CloudTrail 로그 항목이 나와 있습니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
```


AWS CloudFormation 통합

AWS IoT 1-Click mation 은 AWS CloudFormation 과 통합됩니다. AWS CloudFormation 은 클라우드 환경 내 모든 인프라 리소스 (예: Amazon EC2, Auto Scaling, Amazon SNS 등) 를 설명하고 프로비저닝할 수 있는 공통 언어인 AWS CloudFormation 과 통합됩니다. AWS CloudFormation 을 사용하면 간단한 텍스트 파일을 사용하여 자동화되고 안전한 방식으로 모든 리전과 계정에 걸쳐 애플리케이션에 필요한 모든 리소스를 모델링하고 프로비저닝할 수 있습니다. 이 파일은 클라우드 환경에서 신뢰할 수 있는 단일 소스로 사용됩니다. 자세한 내용은 단원을 참조하십시오. [AWS CloudFormation 사용 설명서](#)뿐만 아니라 AWS IoT 1-Click 주제 (예: [AWS::IoT1Click::Project](#)) 의 AWS CloudFormation 사용 설명서.

AWS IoT 1-Click ck에 대한 인증 및 액세스 제어

AWS IoT 1-Click API에 액세스하려면 자격 증명이 필요합니다. 이러한 자격 증명에는 AWS IoT 1-Click 프로젝트 또는 디바이스와 같은 AWS 리소스에 액세스할 수 있는 권한이 있어야 합니다. 다음 단원에서는 AWS Identity of Access Management (IAM) 및 AWS IoT 1-Click 사용하여 리소스에 대한 액세스를 보호할 수 있는 방법에 대한 세부 정보를 제공합니다.

모든 AWS 리소스는 AWS 계정의 소유이고, 리소스 생성 또는 리소스 액세스 권한은 권한 정책에 따라 결정됩니다. 계정 관리자는 IAM 자격 증명(즉 사용자, 그룹 및 역할)에 권한 정책을 연결할 수 있고, 일부 서비스(예: AWS Lambda)에서는 리소스에 대한 권한 정책 연결도 지원합니다. 권한을 부여하려면 관리자는 권한을 부여 받을 사용자, 권한 대상인 리소스, 해당 리소스에 허용되는 특정 작업을 결정합니다.

AWS IoT 1-Click 리소스 및 운영

AWS IoT 1-Click ck에서 기본 리소스는 프로젝트와 디바이스입니다. 정책에서 Amazon 리소스 이름 (ARN)을 사용하여 정책이 적용되는 리소스를 식별합니다. 다음 표에서처럼 이러한 리소스에는 고유한 Amazon Resource Name(ARN)이 연결됩니다.

리소스 유형	ARN 형식
Device	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
프로젝트	arn:aws:iot1클릭:영역:계정 ID:프로젝트/프로젝트 이름

AWS IoT 원클릭은 AWS IoT 1-Click 리소스와 함께 사용할 수 있는 API를 구현합니다. 이러한 작업을 IAM에서 작업이라고 합니다. 사용 가능한 작업 목록은 이 항목의 끝 부분에 있는 표를 참조하십시오.

AWS IoT 1-Click ck에 대한 자격 증명 기반 정책 (IAM 정책) 사용

이 주제에서는 계정 관리자가 권한 정책을 IAM 자격 증명 (즉, 사용자, 그룹 및 역할)에 연결하고 이 과정을 통해 AWS IoT 1-Click 리소스에서 작업을 수행할 권한을 부여할 수 있는 방법을 보여 주는 자격 증명 기반 정책의 예를 제공합니다.

다음은 권한 정책의 예입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot1click:CreateProject"
      ],
      "Resource": "*"
    }
  ]
}
```

정책에는 하나의 문이 있습니다. 이 문은 하나의 AWS IoT 1-Click 작업에 대한 권한을 부여합니다 (iot1click:CreateProject) 을 리소스에 사용할 수 있는 Amazon 리소스 이름 (ARN) 을 사용합니다. 이 경우 ARN은 와일드카드 문자(*)를 사용하여 임의의 리소스에 대해 권한이 부여되었음을 나타냅니다.

모든 AWS IoT 1-Click API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 를 참조하십시오. [AWS IoT 1-Click API 권한: 작업, 권한 및 리소스 참조.](#)

AWS IoT 1-Click에 대한 AWS 관리형 (미리 정의된) 정책

Amazon Web Services 는 AWS 에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반 사용 사례를 처리합니다. 이러한 AWS 관리형 정책은 사용자가 필요한 권한을 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 내용은 단원을 참조하십시오. [AWS 관리형 정책의 IAM 사용 설명서.](#)

계정의 사용자에게 연결할 수 있는 다음 AWS 관리형 정책은 AWS IoT 1-Click ck에 고유하며, 사용 사례 시나리오를 기준으로 그룹화되어 있습니다.

- **AWSIoT1ClickFullAccess:** AWS IoT 1-Click 리소스에 대한 전체 액세스를 부여합니다. 부여된 권한에는 디바이스와 프로젝트를 관리하기 위한 모든 AWS IoT 1-Click 작업이 포함됩니다.
- **AWSIoT1ClickReadOnlyAccess:** AWS IoT 1-Click 리소스에 대한 읽기 전용 액세스를 부여합니다. 이 액세스를 통해 사용자는 AWS IoT 1-Click 디바이스와 프로젝트를 나열하고 프로젝트 구성을 검토할 수 있습니다.

Note

IAM Console에 로그인하여 이러한 권한 정책을 검토할 수 있습니다 (<https://console.aws.amazon.com/iam/>) 에서 특정 정책 이름을 검색할 수 있습니다.

AWS IoT 1-Click 작업 및 리소스에 대한 권한을 허용하는 고유의 사용자 지정 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

AWS IoT 1-Click API 권한: 작업, 권한 및 리소스 참조

AWS 클라우드에서 액세스 제어를 설정하고 IAM 자격 증명에 연결할 수 있는 권한 정책 (자격 증명 기반 정책) 을 설정할 때 다음 표를 참조로 사용할 수 있습니다. 표에는 AWS IoT 1-Click API 작업, 수행할 권한을 부여할 수 있는 해당 작업, 권한을 부여할 수 있는 AWS 리소스가 나열되어 있습니다. 정책의 Action 필드에서 작업을 지정하고, 정책의 Resource 필드에서 리소스 값을 지정합니다.

AWS IoT 1-Click 정책에서 AWS 차원 조건 키를 사용하여 조건을 표시할 수 있습니다. AWS 차원 키의 전체 목록은 단원을 참조하십시오. [사용 가능한 키](#)의 IAM 사용 설명서.

Note

작업을 지정하려면 `iot1click:` 접두사 다음에 API 작업 이름을 사용합니다(예: `iot1click:ListProjects`).

IoT 1-Click 작업	필요한 권한(API 작업)	리소스
ListDevices	IOT1클릭:장치 목록	*
설명장치	IOT1클릭: 장치 설명	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
장치 메서드 가져	IOT1클릭: 장치 메서드 가져 오기:	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
업데이트장치상태	IOT1클릭:업데이트장치상태	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID

IoT 1-Click 작업	필요한 권한(API 작업)	리소스
호출 장치메소드	IOT1클릭: 호출된 장치메소드	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
목록장치이벤트	IOT1클릭:장치 이벤트 목록	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
초기화 장치Claim	IOT1클릭: 초기화장치 초기화	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
파이널리즈디바이스클레임	IOT1클릭: 종료장치클레임	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
책임지지 않은 장치	IOT1클릭:책임지지 않은 장치	ARN:AWS:iot1클릭:지역:계정 ID:장치/장치 ID
클레임 장치클레임 코드	IOT1클릭: 클레임 장치클레임 코드	*
CreateProject	IOT1클릭: 프로젝트 만들기	arn:aws:iot1클릭:영역:계정 ID: 프로젝트/프로젝트 이름
업데이트프로젝트	IOT1클릭:업데이트 프로젝트	arn:aws:iot1클릭:영역:계정 ID: 프로젝트/프로젝트 이름
DescribeProjects	IOT1클릭: 프로젝트 설명	arn:aws:iot1클릭:영역:계정 ID: 프로젝트/프로젝트 이름
ListProjects	IOT1클릭:목록 프로젝트	*
프로젝트 삭제	IOT1클릭:프로젝트 삭제	arn:aws:iot1클릭:영역:계정 ID: 프로젝트/프로젝트 이름
CreatePlacement	IOT1클릭:배치 만들기	arn:aws:iot1클릭:영역:계정 ID: 프로젝트/프로젝트 이름
업데이트배치	IOT1클릭:업데이트배치	arn:aws:iot1클릭:영역:계정 ID: 프로젝트/프로젝트 이름

IoT 1-Click 작업	필요한 권한(API 작업)	리소스
설명 배치	IOT1클릭:배치 설명	arn:aws:iot1클릭:영역:계정 ID:프로젝트/프로젝트 이름
목록 배치	IOT1클릭:목록 배치	arn:aws:iot1클릭:영역:계정 ID:프로젝트/프로젝트 이름
삭제 배치	IOT1클릭:배치 삭제	arn:aws:iot1클릭:영역:계정 ID:프로젝트/프로젝트 이름
연관장치배치와 연관됨	IOT1클릭:연결장치배치와 연관됨	arn:aws:iot1클릭:영역:계정 ID:프로젝트/프로젝트 이름
Dissacociate장치개시	iot1클릭: 해제된 장치에서분리됨	arn:aws:iot1클릭:영역:계정 ID:프로젝트/프로젝트 이름
장치 가져 오기삽입	IOT1클릭: 장치 가져 오기배치	arn:aws:iot1클릭:영역:계정 ID:프로젝트/프로젝트 이름

AWS IoT 1-Click 리소스 태그 지정

AWS IoT 1-Click 리소스를 쉽게 관리할 수 있도록 필요에 따라 태그를 사용하여 ARN 기반 리소스에 고유한 메타데이터를 할당할 수 있습니다. 이 챕터는 태그에 대해 설명하고, 태그를 생성하는 방법을 보여줍니다.

태그 기본 사항

태그를 사용하면 용도, 소유자 또는 환경을 기준으로 하는 등 AWS IoT 1-Click 리소스를 다양한 방식으로 분류할 수 있습니다. 이 기능은 지정한 태그에 따라 특정 리소스를 빠르게 검색하고 식별할 수 있기 때문에 동일한 유형의 리소스가 많을 때 유용합니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다. 예를 들어 특정 관리자 또는 계정이 소유한 여러 버튼에 대한 태그 세트를 정의할 수 있습니다. 추가하는 태그에 따라 리소스를 검색하고 필터링할 수 있습니다. 각 리소스 유형에 대한 요건을 충족하는 태그 키 세트를 고안하는 것이 좋습니다. 일관된 태그 키 세트를 사용하면 리소스를 보다 쉽게 관리할 수 있습니다. 자세한 내용은 [AWS 태그 지정 전략](#)을 참조하십시오.

또한 태그를 사용하여 비용을 분류 및 추적할 수 있습니다. 리소스에 태그를 적용하면, AWS 에서 사용 내역 및 비용이 태그별로 집계된 CSV 파일로 비용 할당 보고서를 만듭니다. 비즈니스 범주를 나타내는 태그(예: 비용 센터, 애플리케이션 이름 또는 소유자)를 적용하여 여러 서비스에 대한 비용을 정리할 수 있습니다. 비용 할당에 태그를 사용하는 방법에 대한 자세한 내용은 [비용 할당 태그 사용의 AWS Billing and Cost Management 사용 설명서](#).

AWS Management Console의 태그 편집기는 태그를 생성하고 관리할 수 있는 중앙 통합 방식으로, 이 도구를 사용하면 아주 편리합니다. 자세한 내용은 단원을 참조하십시오. [Tag Editor 작업](#) in [AWS 관리 콘솔 시작하기](#).

AWS CLI 및 AWS IoT 1-Click 디바이스 및 프로젝트 API를 사용하여 태그 관련 작업을 수행할 수 있습니다. 태그를 생성하면 AWS IoT 1-Click 프로젝트 및 디바이스와 연결할 수 있습니다. tags 필드를 다음 명령에서 사용할 수 있습니다.

- [CreateProject](#)(프로젝트 API)
- [Finalize Claim](#)(디바이스 API)

다음 명령을 사용하여 기존 리소스의 태그를 추가, 수정, 삭제할 수 있습니다.

AWS IoT 1-Click 프로젝트 API 프로젝트 ARN 사용	AWS IoT 1-Click 디바이스 API (디바이스 ARN 사용)
TagResource	Tag
ListTagsForResource	TagResource (POST), TagListTagsForResource (GET) 및 UntagResource (DELETE) 참조
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 제한

태그에 적용되는 기본 제한은 다음과 같습니다.

- 리소스당 최대 태그 수 - 50개
- 최대 키 길이 - UTF-8 유니코드 문자 127자
- 최대 값 길이 - UTF-8 유니코드 문자 255자
- 태그 키와 값은 대/소문자를 구분합니다.
- AWS용으로 예약되어 있기 때문에, 태그 이름이나 값에는 aws 접두사를 사용할 수 없습니다. 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 서비스 또한 허용되는 문자에 대한 제한이 있을 수 있음을 유의하십시오. 일반적으로 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 공백 및 숫자와 특수 문자 + - = . _ : / @

AWS IoT Enterprise Button 사용자 안내서

AWS IoT Enterprise Button 간단하고 쉽게 구성할 수 있는 Wi-Fi 기반 버튼으로, 기업과 개발자가 AWS IoT 1-Click Click을 사용하여 기존 비즈니스 워크플로우와 쉽게 통합할 수 있도록 설계되었습니다.

AWS IoT Enterprise Button 다음 3가지 유형의 클릭을 지원합니다.

- 단일
- Double
- 긴 클릭

제대로 작동하려면 AWS IoT 1-Click 모바일 앱 (iOS 또는 Android) 을 사용하여 버튼의 Wi-Fi 연결을 구성해야 합니다. 앱에서 AWS 계정에 로그인하여 버튼의 Wi-Fi 연결을 구성하거나 앱 오른쪽 위에 있는 Wi-Fi 아이콘을 탭하여 로그인을 건너뛸 수 있습니다.

모바일 앱 또는 콘솔을 통해 연결이 구성되고 디바이스가 신청되면 단일, 이중 또는 긴 클릭이 발생할 때 버튼에 녹색 불이 들어옵니다.

버튼을 구성한 후 버튼에 문제가 있다고 의심되는 경우 이 표를 참고하여 문제를 해결할 수 있습니다.

색상	상태	권장 사항
흰색 깜박임	Wi-Fi에 연결, IP 주소 가져오기 또는 AWS IoT 에 연결 중.	해당 사항 없음
녹색	Wi-Fi에 성공적으로 연결되었으며 AWS IoT 에 메시지를 게시함.	해당 사항 없음
파란색 깜박임	버튼이 구성 모드에 있습니다.	구성 프로세스가 완료될 때까지 기다립니다.
주황색	Wi-Fi가 구성되지 않았습니다.	AWS IoT 1-Click 모바일 앱을 사용하여 Wi-Fi를 구성합니다.
빨간색: 짧은, 짧은, 짧은	구성된 무선 네트워크에 연결하는 동안 오류가 발생했습니다.	네트워크 설정이 변경되었는지 또는 버튼이 Wi-Fi 라우터에서

색상	상태	권장 사항
		너무 멀리 떨어져 있는지 확인하십시오.
빨간색: 짧은, 짧은, 긴	무선 네트워크에서 IP 주소를 가져오는 중 오류가 발생했습니다.	무선 네트워크 문제를 확인합니다.
빨간색: 짧은, 긴, 짧은	호스트 이름 조회를 수행하는 동안 오류가 발생했습니다.	무선 네트워크 문제를 확인합니다.
빨간색: 짧은, 긴, 긴	AWS IoT 에 연결할 수 없습니다.	무선 네트워크 문제를 확인합니다. 네트워크 문제가 있고 문제가 지속되는 경우 AWS 지원 센터 을 클릭하고 디바이스 일련 번호 (DSN) 를 알려 주십시오. 이 번호는 버튼 뒷면에 있습니다.
빨간색: 긴, 짧은, 짧은	서버와의 보안 연결을 설정할 수 없습니다.	AWS IoT 1-Click iOS 또는 Android 모바일 앱을 사용하여 최신 펌웨어가 설치되어 있는지 확인합니다.
빨간색: 긴, 짧은, 긴	HTTP 403 금지 오류를 받았습니다.	에 문의하십시오. AWS 지원 센터 을 클릭하고 DSN를 알려 주십시오. 이 번호는 버튼 뒷면에 있습니다.
빨간색: 15초간 버튼을 누른 후	버튼 재설정.	버튼을 15초 동안 눌러 AWS IoT Enterprise Button Wi-Fi 구성을 재설정할 수 있습니다.

AWS IoT 1-Click에서 사용

AWS Command Line Interface (AWS CLI) 를 사용하기 위해 AWS IoT 1-Click 클라이언트를 사용하여 쓰레기 수거 서비스를 효율화하려는 쓰레기 수거 업체의 시나리오를 살펴보겠습니다.

이 시나리오에서는 각 쓰레기 수거통이 AWS IoT Enterprise Button 과 연결됩니다. 쓰레기 수거통이 가득 차면 고객이 쓰레기 수거통을 교체하도록 요청하기 위해 연결된 버튼을 누르기만 하면 됩니다.

Note

모든 AWS IoT Enterprise Button 디바이스 ID는 "G030PM"으로 시작합니다.

다음은 쓰레기 수거 업체에서 고객이 사용할 AWS IoT Enterprise Button 준비하는 단계입니다.

고객이 사용할 AWS IoT Enterprise Button 준비하려면

1. AWS IoT 1-Click 모바일 앱을 사용하는 유일한 방법은 AWS IoT (IoT 1-Click) 을 사용하는 것입니다. 앱을 설치하려면 [AWS IoT 1-Click 모바일 앱](#) 단원을 참조하십시오. 앱을 설치한 후 보통 때처럼 AWS 계정에 로그인을 누르지 마십시오. 이 연습은 AWS CLI를 사용하는 방법을 알아보기 위한 것입니다. AWS 계정에 로그인을 누르면 `initiate-device-claim` 및 `finalize-device-claim` 명령이 호출되는데, 다음 단계와 같이 CLI를 사용하여 "수동으로" 이 작업을 수행하려고 합니다.
2. AWS CLI 데모를 위해 AWS 계정에 로그인의 오른쪽 위 모서리에 있는 작은 원 모양의 Wi-Fi 아이콘을 선택합니다. 그런 다음 Wi-Fi 구성을 선택합니다. 디바이스 ID를 스캔하거나 입력하고 모바일 앱의 나머지 지침을 따릅니다.
3. AWS CLI가 설치되지 않은 경우 [AWS CLI 설치](#). 사용 가능한 AWS IoT 1-Click AWS CLI 명령을 보려면 다음 두 명령을 실행합니다.

```
aws iot1click-projects help
```

```
aws iot1click-devices help
```

4. 이제 Wi-Fi에 연결된 AWS IoT Enterprise Button 쓰레기 수거 업체의 AWS 계정에 연결하려면 디바이스의 디바이스 ID를 사용하여 다음 명령을 실행합니다.

```
aws iot1click-devices initiate-device-claim --device-id G030PM0123456789
{
  "State": "CLAIM_INITIATED"
```

```
}

```

디바이스에서 해당 버튼을 누릅니다. 흰색 불이 간헐적으로 깜박인 후 약 1초 동안 녹색 불이 켜집니다. 그렇지 않은 경우 이전 Wi-Fi 연결 절차를 다시 수행합니다.

5. 앞 단계에서 녹색 불이 켜진 후에 디바이스의 ID 값을 사용하여 다음 명령을 실행합니다.

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

이 "State": "CLAIMED" 응답은 디바이스가 AWS IoT 1-Click 서비스에 성공적으로 등록되었음을 나타냅니다.

Note

디바이스 제조업체에서 "C-"로 시작하는 신청 코드를 제공한 경우 다음 예와 같이 `aws iot1click-devices claim-devices-by-claim-code` 명령으로 단일 신청 코드를 사용하여 하나 이상의 디바이스를 신청할 수 있습니다.

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

이 예에서는 "Total": 9은 9개의 디바이스가 클레임 코드와 연결되어 있음을 나타냅니다. C-123EXAMPLE가 AWS IoT 1-Click 서비스에서 성공적으로 신청되었습니다.

6. 다음으로 이라는 JSON 텍스트 파일을 만들어 쓰레기 수거 업체에 적합한 AWS IoT 1-Click 프로젝트를 생성할 준비를 합니다. `create-project.json`. 이 파일에는 다음이 포함되어 있습니다.

```
{
  "projectName": "SeattleDumpsters",
  "description": "All dumpsters in the Seattle region.",
  "placementTemplate": {
    "defaultAttributes": {
      "City" : "Seattle"
    },

```

```

    "deviceTemplates": {
      "empty-dumpster-request" : {
        "deviceType": "button"
      }
    }
  }
}

```

placementTemplate 및 deviceTemplates 키-값 페어는 SeattleDumpsters 프로젝트의 일부인 모든 버튼에 적용되는 속성입니다. 이 프로젝트를 만들려면 다음 명령을 실행합니다 (create-project.json에 있는 [현재 작업 디렉터리](#)에서 AWS CLI 명령 프롬프트) 입니다.

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

새로 생성된 프로젝트를 보려면 다음 명령을 실행합니다.

```

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

```

자세한 내용을 보려면 다음과 같이 describe-project 명령을 실행합니다.

```

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

```

```

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

```

7. 시애틀 리전에 대한 프로젝트가 생성되었으니 다음과 같이 특정 쓰레기 수거통(고객 217용)에 대한 배치를 생성합니다. 이스케이프된 따옴표는 Windows에 필요합니다.

```

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

```

새로 생성된 배치를 보려면 다음 명령을 실행합니다.

```

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

```

자세한 내용을 보려면 다음과 같이 describe-placement 명령을 실행합니다.

```

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

```



```

    "projectName": "SeattleDumpsters",
    "placementName": "customer217",
    "attributes": {
      "phone": "206-123-4567",
      "location": "1800 9th Ave Seattle, WA 98101"
    },
    "createdDate": 1563488454,
    "updatedAt": 1563488454
  }
}

```

8. 이제 디바이스는 쓰레기 수거 업체의 AWS IoT 1-Click 계정과 연결되어 있지만 배치와 연결되어 있지 않습니다. 다음 명령을 실행하면 이를 확인할 수 있습니다.

```

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

```

디바이스를 배치에 연결하려면 다음 명령을 실행합니다.

```

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

```

이전 명령을 확인하려면 `get-devices-in-placement`를 다시 실행합니다.

```

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

```

자세한 내용을 보려면 다음과 같이 `describe-device` 명령을 실행합니다. `iot1click-projects`에서 `iot1click-devices`로 전환된 것에 주목하십시오.

```

aws iot1click-devices describe-device --device-id G030PM0123456789
{

```

```

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

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

```

현재 하나의 디바이스만 있기 때문에 다음 명령을 실행하면 비슷한 결과가 나타납니다.

```

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

```

9. 디바이스가 제대로 작동하는지 확인하려면 다음 명령을 실행합니다. [ISO 8061 형식](#)인 타임스탬프를 적절하게 조정합니다.

```
aws iot1click-devices list-device-events --device-id G030PM0123456789 --from-time-stamp 2019-07-17T15:45:12.880Z --to-time-stamp 2019-07-19T15:45:12.880Z
{
  "Events": [
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\"clickType\": \"SINGLE\",
        \"reportedTime\": \"2019-07-18T23:47:55.015Z\", \"certificateId\":
        \"fe8798a6c97c62ef8756b80eeefdcd2280f3352f82faa8080c74cc4f4a4d1811\",
        \"remainingLife\": 99.85000000000001, \"testMode\": false}"
    }
  ]
}
```

여기서 한 번 클릭 이벤트(`"clickType": "SINGLE"`)가 2019-07-18T23:47:55.015Z에 발생한 것을 볼 수 있습니다. 이제 디바이스를 두 번 클릭하고(버튼을 두 번 연속 빠르게 누름) 명령을 다시 실행합니다. 다음과 유사한 두 번 클릭 이벤트(`"clickType": "DOUBLE"`)를 확인할 수 있습니다.

```
aws iot1click-devices list-device-events --device-id G030PM0123456789 --from-time-stamp 2019-07-17T15:45:12.880Z --to-time-stamp 2019-07-19T15:45:12.880Z
{
  "Events": [
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\"clickType\": \"SINGLE\",
        \"reportedTime\": \"2019-07-18T23:47:55.015Z\", \"certificateId\":
        \"fe8798a6c97c62ef8756b80eeefdcd2280f3352f82faa8080c74cc4f4a4d1811\",
        \"remainingLife\": 99.85000000000001, \"testMode\": false}"
    },
    {
      "Device": {
        "Attributes": {},

```

```

        "DeviceId": "G030PM0123456789",
        "Type": "button"
    },
    "StdEvent": "{\"clickType\": \"DOUBLE\",
    \\\"reportedTime\\\": \\\"2019-07-19T00:14:41.353Z\\\", \\\"certificateId\\\":
    \\\"fe8798a6c97c62ef8756b80eeefdcd2280f3352f82faa8080c74cc4f4a4d1811\\\",
    \\\"remainingLife\\\": 99.8, \\\"testMode\\\": false}"
    }
  ]
}

```

10. 각 디바이스 유형에는 호출 가능한 디바이스 메서드 세트가 있습니다. 해당 디바이스 유형에서 사용할 수 있는 메서드 목록을 보려면 다음과 같이 `get-device-methods` 명령을 실행합니다.

```

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

```

사용 가능한 메서드 중 하나를 호출하려면 다음과 같이 `invoke-device-method` 명령을 사용합니다.

```

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

```

여기서 `invoke-device-method.json`에는 다음이 포함되어 있습니다.

```
{
  "DeviceId": "G030PM0123456789",
  "DeviceMethod": {
    "DeviceType": "device",
    "MethodName": "getDeviceHealthParameters"
  }
}
```

Note

get 메서드(예: `getDeviceHealthParameters`)에는 파라미터가 사용되지 않습니다. 따라서 JSON 파일 내의 `"DeviceMethodParameters": ""` 줄을 사용할 수 없습니다. 이렇게 하면 다음과 같은 오류가 발생합니다. `An error occurred (InvalidRequestException) when calling the InvokeDeviceMethod operation: A request parameter was invalid.`

11. `aws iot1click-devices list-devices --device-type button`을 실행하면 `Enabled`의 기본값이 `false`인 것을 볼 수 있습니다. 다음 명령은 이 키를 `true`로 설정합니다.

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

다시 `false`로 설정하려면 `--no-enabled` 인수를 사용하여 앞의 명령을 다시 실행합니다.

12. 고객 정보가 변경되면 다음과 같이 디바이스의 배치 정보를 업데이트할 수 있습니다. `iot1click-devices`에서 `iot1click-projects`로 전환된 것에 주목하십시오. 다음 명령을 실행하여 `customer217`의 현재 정보를 봅니다(`attributes`에 있음).

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

```

    "updatedAt": 1563488454
  }
}

```

그리고 나서 다음 명령을 실행하여 고객의 전화 및 위치 속성을 업데이트합니다.

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

여기서 update-placement.json에는 다음이 포함되어 있습니다.

```

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

```

이 업데이트를 검토하려면 다음과 같이 describe-placement를 다시 실행합니다.

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

```

- 프로젝트 정보를 업데이트하려면 update-project 명령을 사용합니다. 프로젝트에는 일반적으로 여러 고객 배치가 포함되어 있습니다. 다음은 기존 SeattleDumpster 프로젝트 정보입니다.

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

```
{
  "project": {
    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
    "projectName": "SeattleDumpsters",
    "description": "All dumpsters in the Seattle region.",
    "createdDate": 1563483100,
    "updatedAt": 1563483100,
    "placementTemplate": {
      "defaultAttributes": {
        "City": "Seattle"
      },
      "deviceTemplates": {
        "empty-dumpster-request": {
          "deviceType": "button",
          "callbackOverrides": {}
        }
      }
    },
    "tags": {}
  }
}
```

“All dumpsters”을 “All dumpsters (yard바스터, 재활용, 쓰레기 수거) 를 사용하려면 다음 명령을 실행합니다.

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

모든 SeattleDumpsters 배치에 대해 "description" 키의 값이 업데이트된 것을 볼 수 있습니다.

```
aws iot1click-projects describe-project --project-name SeattleDumpsters
{
  "project": {
    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
    "projectName": "SeattleDumpsters",
    "description": "All dumpsters (yard waste, recycling, garbage) in the Seattle region.",
    "createdDate": 1563483100,
    "updatedAt": 1563819039,
```

```

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

```

14. 다음과 같이 태그를 사용하여 프로젝트 리소스(iot1click-projects) 및 배치 리소스(iot1click-devices)에 메타 정보를 적용할 수 있습니다.

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

여기서 projects-tag-resource.json에는 다음이 포함되어 있습니다.

```

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

```

프로젝트 리소스의 태그 목록을 보려면 다음을 실행합니다.

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

```



```
}
```

컨텍스트에서 프로젝트 태그를 보려면 다음을 실행합니다.

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

디바이스 Amazon 리소스 이름(ARN)을 검색하려면 다음을 실행합니다.

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

```

```

        "projectName": "SeattleDumpsters",
        "placementName": "customer217",
        "deviceTemplateName": "empty-dumpster-request"
    },
    "DeviceId": "G030PM0123456789",
    "Enabled": true,
    "RemainingLife": 99.7,
    "Type": "button",
    "Tags": {}
}
]
}

```

이전 디바이스에 태그를 추가하려면 다음을 실행합니다.

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

여기서 `devices-tag-resources.json`에는 다음이 포함되어 있습니다. `ResourceArn` 및 `Tags`는 대소문자를 구분합니다.

```

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

```

디바이스 리소스의 태그 목록을 보려면 다음을 실행합니다.

```

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

```

컨텍스트에서 디바이스 태그를 보려면 `list-devices` 명령을 실행합니다.

```
aws iot1click-devices list-devices
{
  "Devices": [
    {
      "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789",
      "Attributes": {
        "projectRegion": "us-west-2",
        "projectName": "SeattleDumpsters",
        "placementName": "customer217",
        "deviceTemplateName": "empty-dumpster-request"
      },
      "DeviceId": "G030PM0123456789",
      "Enabled": true,
      "RemainingLife": 99.7,
      "Type": "button",
      "Tags": {
        "Driver Phone": "206-123-4567",
        "Driver": "John Smith"
      }
    }
  ]
}
```

15. 이제 AWS Lambda 함수를 트리거하거나 Amazon SNS 메시지를 보내는 것과 같은 작업을 디바이스 버튼 누르기에 연결할 수 있습니다. AWS IoT 1-Click 콘솔 ([AWS IoT 1-Click 프로그래밍 모델도 옵션 임](#)). 적절한 작업을 디바이스에 연결한 후에는 1단계 및 2단계에서 설명한 것과 동일한 절차를 사용하여 디바이스를 고객의 위치로 가져 와서 Wi-Fi 네트워크에 연결할 수 있습니다.

AWS IoT 1-Click 디바이스 해체

다음 단계에서는 앞의 단계를 되돌리는(실행 취소) 방법에 대해 설명합니다.

1. 프로젝트 리소스에서 태그를 제거하려면 다음 명령을 실행합니다.

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

그러면 다음과 같이 프로젝트의 Manager 태그가 제거됩니다.

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

2. 디바이스 리소스에서 태그를 제거하려면 다음 명령을 실행합니다.

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

그러면 다음과 같이 디바이스의 태그가 제거됩니다. "Tags": {}로 목록이 비어 있는 것을 볼 수 있습니다.

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

```

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

```

3. 배치에서 디바이스의 연결을 해제하려면 다음 명령을 실행합니다.

```

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

```

다음에서 볼 수 있듯이 customer217 배치에 연결된 디바이스가 더 이상 없습니다.

```

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

```

4. 프로젝트에서 배치를 삭제하려면 다음 명령을 실행합니다.

```

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

```

다음에서 볼 수 있듯이 SeattleDumpsters 프로젝트에는 배치가 없습니다. customer217 배치가 SeattleDumpsters 내의 유일한 배치이기 때문입니다.

```

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

```

5. 프로젝트를 삭제하려면 다음 명령을 실행합니다.

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

다음에서 볼 수 있듯이 모든 프로젝트가 제거됩니다. SeattleDumpsters는 AWS IoT 1-Click 계정과 연결된 유일한 프로젝트입니다.

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

예를 들어 친구가 자신의 AWS 계정을 사용하여 디바이스를 시험해 볼 수 있도록 하려면 먼저 다음과 같이 AWS IoT 1-Click 계정에서 디바이스의 신청을 취소해야 합니다.

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

이제 모든 AWS IoT 1-Click 계정에서 디바이스를 사용할 수 있습니다.

AWS IoT 1-Click 부록


다음 단원에서는 AWS IoT 1-Click 정보를 추가로 제공합니다.

AWS IoT 1-Click 지원되는 디바이스

제품	디바이스 유형	디바이스 ID 접두사	디바이스를 신청하려면	구입 링크	디바이스 리전†
AWS IoT (미국, EU 및 일본)에 사용되는 IoT 버튼	Button	P5SJVQ(디바이스 ID의 처음 6자리)	AWS IoT 1-Click 모바일 앱에서 Wi-Fi를 구성하고 디바이스를 요청할 디바이스 ID를 입력합니다.	Seeed Studio Bazaar	US West (Oregon)
서콤 IoT 버튼 (미국만 해당)	Button	7VT4EQ (디바이스 ID의 처음 6자리)	AWS IoT 1-Click 모바일 앱 또는 AWS IoT 1-Click 콘솔에서 디바이스를 구입할 때 획득한 클레임 코드를 입력합니다. AWS IoT 1-Click 모바일 앱에 디바이스 ID를 입력하여 디바이스를 요청할 수도 있습니다.	세르콤 모바일 제품 온라인 스토어	US West (Oregon)

제품	디바이스 유형	디바이스 ID 접두사	디바이스를 신청하려면	구입 링크	디바이스 리전 [†]
SOLACOM LTE-M Button (일본에서만)	Button	7MF6JK(디바이스 ID의 처음 6자리)	AWS IoT 1-Click 모바일 앱에서 디바이스를 신청할 디바이스 ID를 입력합니다.	SORACOM	US West (Oregon)
AWS IoT Enterprise Button (미국, EU 및 일본)	Button	G030PM(디바이스 ID의 처음 6자리)	AWS IoT 1-Click 모바일 앱에서 디바이스 ID를 입력하여 Wi-Fi를 구성하고 디바이스를 신청합니다.	공급 중단	US West (Oregon)
AT&T LTE M 버튼 (미국만 해당)	Button	B9GHXT (디바이스 ID의 처음 6자리)	AWS IoT 1-Click 모바일 앱 또는 AWS IoT 1-Click 콘솔에서 디바이스를 구입할 때 획득한 클레임 코드를 입력합니다. AWS IoT 1-Click 모바일 앱에 디바이스 ID를 입력하여 디바이스를 요청할 수도 있습니다.	공급 중단	US West (Oregon)

†디바이스 리전에 대한 자세한 내용은 [AWS IoT 1-Click 디바이스](#).

 Note

은 DSN (디바이스 일련 번호) 이 G030JF, G030MD 및 G030PT 로 시작하는 AWS IoT 1-Click 에서 시작하는 AWS IoT 버튼을 지원하지 않습니다. AWS IoT 1-Click을 사용하지 않고 AWS IoT 클라우드에 이러한 버튼을 연결하는 방법을 알아보려면 단원을 참조하십시오. [클라우드 프로그래밍 가능 대시 버튼](#).

AWS IoT 1-Click 서비스 제한

- 배치 템플릿당 최대 5개의 디바이스 템플릿을 사용할 수 있습니다. 이는 배치당 5개의 디바이스에 해당합니다.
- AWS IoT 1-Click 프로젝트당 최대 512개의 프로젝트가 있습니다. [AWS 리전](#)계정당.
- AWS IoT 1-Click 리소스당 최대 50개의 태그가 있습니다. 태그는 리소스를 관리하는 데 사용 가능한 키-값 페어(메타데이터)입니다. 자세한 내용은 [AWS 태그 지정 전략](#)을 참조하십시오.

개발자 안내서의 문서 이력

다음 표에서는 본 AWS IoT 1-Click의 릴리스 문서에 대해 설명합니다.

- API 버전: 최신
- 최신 설명서 업데이트: 2018년 10월 22일

변경 사항	설명	날짜
릴리스	설명서의 최초 릴리스	2018년 5월 14일
수정	편집 개선	2018년 5월 31일
수정	지원되는 디바이스 테이블 업데이트	2018년 10월 22일

AWS 용어집

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

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.