aws

開発者ガイド

AWS IoT 1-Click

AWS IoT 1-Click: 開発者ガイド

Table of Contents

AWS IoT 1-Click とは何ですか？

AWS IoT 1-Click を使用すると、エンタープライズのお客様がシンプルな IoT デバイスをワークフローに簡単に組み込めるようになります。デバイスを製造したり、ファームウェアを作成したり、セキュアな接続を設定する必要はありません。当社の製造パートナーは、細かい設定なしに AWS IoT に安全に接続できるデバイスを作成します。これらのデバイスはAWS LambdaJava、Python、C# などの言語で作成された関数です。Lambda 関数は、独自のビジネスロジックを実装することも、AWS クラウドまたはオンプレミスでアクションをトリガーすることもできます。

AWS IoT 1-Click は、デバイスのハードウェアとファームウェアに関する詳細を可能な限り抽象化することにより、顧客の IoT を簡素化することを目指しています。これにより、AWS IoT 1-Click デバイスをソフトウェアコンポーネントとして表示することができます。これらのデバイスは、他のソフトウェアコンポーネントと同様、明確に定義されたインターフェイスに準拠しています。AWS IoT 1-Click には、デバイスのタイプごとに定義されたインターフェイスがあります。これらのインターフェイスを使用して、アプリケーションを構築し、ベースにすることができます。

AWS IoT 1-Click を使用すると、機能、場所、またはその他の基準でデバイスをグループ化できます。デバイスの、この論理的なグループ化は、」プロジェクトAWS IoT 1-Click で。プロジェクトを使用して、デバイスのグループを、希望のアクションの Lambda 関数に関連付けることができます。

プロジェクトには、使用されるデバイスのタイプ、呼び出される Lambda の機能、ロケーションや機能のコンテキストデータなど、オプションの属性が定義されているテンプレートが含まれています。

プロジェクトを作成してテンプレートを定義したら、プロジェクト内にプレイスメントを追加することができます。それぞれがテンプレートに従ってその特定のプレイスメントの特定の場所または関数に合ったシリアル番号と属性値 (キー/値ペア) によって実際のデバイスを指定します。

AWS IoT 1-Click コンポーネント

Claim

> AWS IoT 1-Click コンソール、AWS IoT 1-Click モバイルアプリ、または AWS IoT 1-Click API を使用して、AWS IoT 1-Click デバイスを AWS アカウントに関連付けるプロセスを指します。

請求コード

一度に (つまり、一括で) 多数の AT&T LTE-M ボタンを要求するために使用される値。デバイス ID を使用してデバイスを要求することもできます。デバイス ID のエントリを参照してください。

デバイス

AWS IoT Enterprise Button や AT&T LTE-M ボタンなどの物理デバイス。

デバイスの属性

キーと値のペアの形式で特定のデバイスに関連付けられたデフォルトデータまたはカスタムデータ。デフォルトの属性はプレイスメントから取得されます。プレイスメントのエントリを参照してください。

デバイス ID

すべてのデバイスには、デバイスシリアルナンバー (DSN) などのデバイス ID があります。デバイス ID を使用して、AWS IoT 1-Click デバイスを AWS IoT 1-Click に登録できます。要求コードはデバイス ID と同じではありません。要求コードのエントリを参照してください。

Placement

デバイスを表す 1 つ以上のテンプレートのグループ (例: 2 つのテンプレート化されたボタンを含むルーム)。プレイスメントに入力するには、AWS IoT 1-Click モバイルアプリケーションを使用して、テンプレート化されたデバイスを選択します。

プレイスメント名

多くの場合、地理的位置またはオブジェクト ID (`Room 217`、`North Dumpster`、`Container 314` など) が含まれるプレイスメントの名前。

プロジェクト

ゼロ以上のプレイスメントの名前付きグループ (テンプレート化されたデバイスを含む)。

Project name

プレイスメントのグループのわかりやすい名前 (「会議室の満足度」または「チャーターコンテナピックアップ」など)。

テンプレート

デバイスのグループのデフォルトの動作とデフォルトの属性を提供するために使用されます。物理デバイスは特定のテンプレートを使用してそのテンプレートのプロパティを継承します-

Lambda 関数とデフォルトのデバイスの属性です。テンプレートは、プレイスメント内のデバイスのクラスの動作属性とデフォルト属性を定義します。プロジェクトには複数のテンプレートを使用できます。

要求解除

AWS IoT 1-Click デバイスの関連付けを解除するプロセスです。たとえば、AWS IoT 1-Click デバイスを貸したい人は、新しいユーザーがデバイスを自分の AWS アカウントに関連付けることができうように、まずデバイスを AWS アカウントから関連付けを解除します。

AWS IoT 1-Click のしくみ

AWS IoT 1-Click ワークフローは次のとおりです。

1. サポートされている一連のデバイスから選択します。
2. AWS Lambda 関数を、アクションをトリガーするデバイスに関連付けます。独自の Lambda 関数の 1 つ、またはサービスにより提供されている事前定義された関数の 1 つを使用します。
3. デバイスを物理的にデプロイし、AWS IoT 1-Click コンソール、AWS IoT 1-Click モバイルアプリ、または AWS IoT 1-Click API を使用してデバイスを有効にします。
4. 既製の AWS IoT 1-Click レポートを使用するか、ユーザー自身の AWS IoT 1-Click レポートを作成してすることで、デバイスのステータスと使用方法について説明します。

AWS IoT 1-Click デバイス

AWS IoT 1-Click がサポートされているデバイス:

- 既成のデバイスがあります。顧客がデバイスを設計したり製造する必要はありません。
- AWS IoT 1-Click アカウントに追加するには、要求機能。
- 製造時に証明書を使用して事前プロビジョニングされ、AWS IoT に安全に接続するように設定されます。AWS IoT 1-Click デバイスの証明書のインストールに時間をかける必要はありません。
- 各 AWS IoT 1-Click デバイスtypeAWS IoT 1-Click で定義される標準形式のイベントを出力します。たとえば、すべての AWS IoT 1-Click デバイスは、`button`タイプは、製造元にかかわらず同じイベント形式を持っています。
- デバイスタイプと製品タイプがあります。デバイスタイプは、デバイスによって出力されるイベントの形式と、サポートしているデバイスのメソッドを示します。詳細については、「AWS IoT 1-Click プログラミングモデル」を参照してください。製品タイプは、製造元やブランディングの詳細を提供します。たとえば、デバイスタイプが`button`の場合、製品タイプは AT&T LTE-M Button になります。

> **Important**
>
> AWS IoT 1-Click がサポートされているデバイスは、特定のAWS リージョン。これらはデバイスリージョンと呼ばれます。このデバイスのデバイスリージョンは、デバイスが追加の

ユーザー入力なしに AWS IoT に安全に接続されるようにするために必要です。このため、デバイスリージョンは変更できません。
AWS IoT 1-Click によって出力されたイベントは、常に事前定義されたデバイスリージョンを介してルーティングされ、同じ AWS リージョンのデバイス関連の Amazon CloudWatch Logs および AWS CloudTrail メトリックスにアクセスできます。デバイスリージョンは、有効なデバイスに課金される場所でもあります。
プレイスメント、テンプレート、およびプロジェクトのデータは、アカウントに関連付けられた AWS リージョンに保存されます。このリージョンは、デバイスリージョンとは異なる場合があります。

AWS IoT 1-Click がサポートされているデバイスについては、購入方法およびクレームそれらについては、AWS IoT 1-Click 付録。

デバイスの要求

AWS IoT 1-Click デバイスは工場出荷時に AWS のお客様のアカウントに関連付けられているわけではありません。顧客はアカウント内のデバイスを使用するに要求プロセスを実行する必要があります。デバイスを要求する方法は 2 つあります。

- ギフト券コードを使用する: あなたはクレームコードを受け取った場合 (フォーマットC-*xxxxx*) を購入時点で要求できる場合は、AWS IoT 1-Click コンソールまたは AWS IoT 1-Click モバイルアプリに入力して単一の注文に関連するデバイスを要求できます。AWS IoT Enterprise Button を含む、すべてのデバイスが要求コードを使用して要求できるわけではありません。
- デバイス ID の使用: デバイス ID (DSN として知られるデバイスシリアルナンバー) を使用して、AWS IoT 1-Click コンソールまたは AWS IoT 1-Click モバイルアプリを介してデバイスを要求できます。すべての AWS IoT 1-Click デバイスはデバイス ID を使用して要求できます。

デバイスを要求する方法の詳細については、「AWS IoT 1-Click 付録」および「デバイスの要求」を参照してください。

「プロジェクト、テンプレート、およびプレイスメント」

デバイスは関数、場所、またはその他の基準によって整理できます。デバイスのこの論理的なグループ化は、プロジェクトと呼ばれます。プロジェクトを使用して、デバイスのグループを Lambda 関数に関連付けることができます。

プロジェクトには使用するデバイスのタイプを指定するテンプレート、呼び出す Lambda 関数、場所や関数のようなコンテキスト依存データを保持するための属性名が含まれています。

プロジェクトを作成してテンプレートを定義したら、プロジェクト内にプレイスメントを追加することができます。プレイスメントはテンプレートに従い、そのプレイスメントの特定の場所または関数に合ったシリアル番号と属性値によってデバイスを指定します。

次は、プロジェクトとプレイスメントの使用方法を示す例です。

例 1:

SalesPersonNotification プロジェクトでは、10 人の顧客が、販売担当者に連絡するために押すことができるボタンを受け取ります。顧客ごとに 1 つずつ、10 個のプレイスメントがあります。各プレイスメントには、CustomerName (例: Jones 氏)、SalesPersonPhoneNumber (例: 1-555-555-1234) とボタンのシリアルナンバー (例: G030PM12345678) の値があります。デバイステンプレート、NotficationButton は、プレイスメントに含まれています。CustomerName および SalesPersonPhoneNumber 属性が各プレイスメントごとに定義されます。お客様がボタンをクリックすると、AWS IoT 1-Click はSendSMSLambdaとCustomerNameおよびSalesPersonPhoneNumberそのボタンに関連付けられた値。SMS がこれらの値に基づいて送信されます。

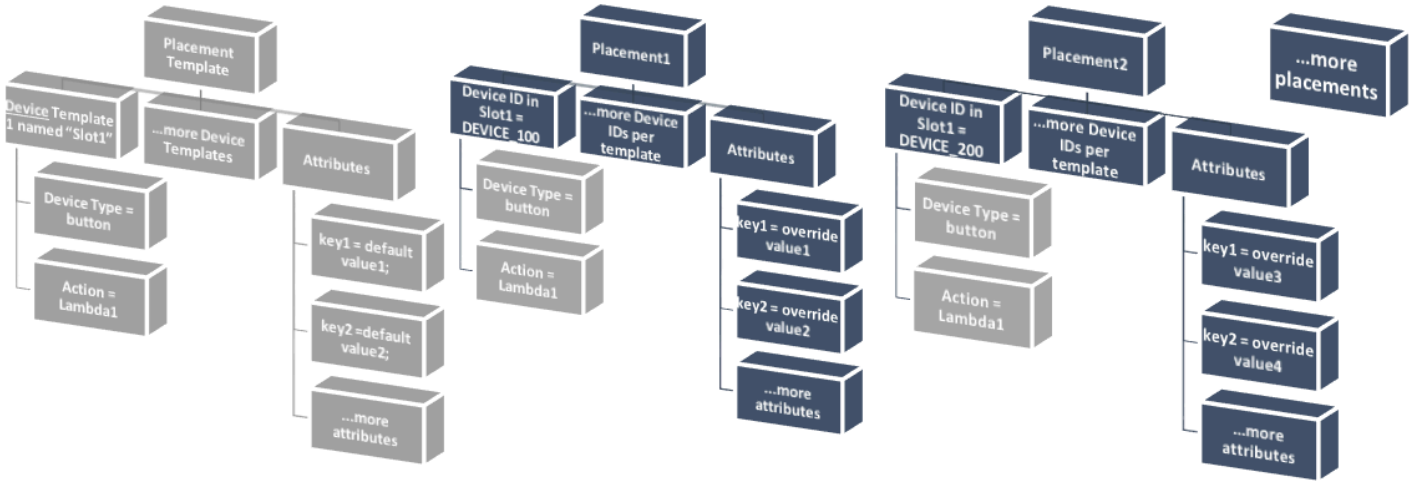
- プレイスメントテンプレート:
  - それぞれの顧客が販売担当者に通知するボタンを取得するため、NotficationButton という名前のデバイステンプレートが作成されます。
  - デバイステンプレート (プレイスメントに含まれる) は、NotificationButtonを呼び出します。クリックすると、SendSMSLambdaLambda 関数。
  - CustomerName および SalesPersonPhoneNumber という名前の属性が各プレイスメントに定義されます。
- プレイスメント: 1 人の顧客につき、10 個のプレイスメントが作成されます。各プレイスメントには、CustomerName (例:「Jones 氏」)、SalesPersonPhoneNumber (例: 1-555-555-1234)、ボタンのシリアル番号 (例: G030PM12345678) の特定の値があります。
- オペレーション: 顧客がボタンをクリックすると、AWS IoT 1-Click はSendSMSLambdaとCustomerNameおよびSalesPersonPhoneNumber値が関連付けられ、SMS がそれらの値に基づいて送信されます。

例 2:

MeetingRoomFeedback プロジェクトでは、50 の会議室ごとに、Thumbs Up と Thumbs Down ボタンを押すことで、ユーザー満足度が追跡されます。ThumbsUp と ThumbsDown の 2 つのデバイステンプレートがあります。Thumbs Up ボタンがクリックされると、PostiveFeedbackLambda 関数が呼び出されます。Thumbs Down ボタンがクリックされると、NegativeFeedbackLambda が呼び出されます。AMeetingRoomNumber属性が定義されて、各プレイスメントの部屋番号を保持します。会議室ごとに 1 つ、50 個のデバイスプレイスメントが作成されます。各プレイスメントには、部屋番号 (例: 1001) に設定された MeetingRoomNumber キーと、固有のシリアル番号 (例: G030PM12345678 および G030PM23456789) によって特定される 2 つのボタンが含まれます。会議室でボタンがクリックされると、AWS IoT 1-Click はPositiveFeedbackLambda関数またはNegativeFeedbackLambda関数を使用してMeetingRoomNumber値. フィードバックが処理され、一覧表示されます。

- プロジェクト名: MeetingRoomFeedback
- プレイスメントテンプレート:
  - 各部屋には 2 つのボタンがあるため、それぞれ ThumbsUp と ThumbsDown という名前の 2 つのデバイステンプレートが作成されます。
  - デバイステンプレートは、ThumbsUp ボタンをクリックすると、PostiveFeedbackLambda を呼び出し、ThumbsDown ボタンでは、NegativeFeedbackLambda を呼び出すように指定します。
  - MeetingRoomNumber という名前の属性が定義されて、各プレイスメントの部屋番号を保持します。
- プレイスメント: 1 つの部屋ごとに 1 つのプレイスメント、50 個のデバイスプレイスメントが作成されます。各プレイスメントには、特定の部屋番号ペア (例: 1001) に設定された MeetingRoomNumber キーと固有のシリアル番号 (例: G030PM12345678 および G030PM23456789) によって特定される 2 つのボタンが含まれます。
- オペレーション: 会議室でボタンがクリックされると、AWS IoT 1-Click はPositiveFeedbackLambda関数またはNegativeFeedbackLambda関数を使用してMeetingRoomNumbervalue — フィードバックが処理され、一覧表示されます。

以下の図に、これらのコンセプトを示します。



詳細については、「AWS IoT 1-Click コンソールのスタートアップ」を参照してください。

AWS IoT 1-Click コンソールのスタートアップ

以下のトピックでは、一般的な AWS IoT 1-Click タスクを実行する方法について説明します。

トピック

デバイスの要求

次の手順では、1 つ以上の AWS IoT 1-Click 対応デバイスを要求する方法について説明します。

1. AWS アカウントにサインインします。AWS アカウントをお持ちでない場合は、を開きます。https://aws.amazon.com/] で、[AWS 無料アカウント作成方法にアクセスし、オンラインの手順に従います。
2. AWS マネジメントコンソールから「1-Click」を検索し、AWS IoT 1-Click を選択します。
3. 1 つ以上の AWS IoT エンタープライズボタンを使用している場合は、iOS 用または Android 用の AWS IoT 1-Click モバイルアプリをインストールして、ボタンをローカル Wi-Fi ネットワークに接続します。AWS IoT 1-Click モバイルアプリケーションは、オンボードAWS IoT 1-Click コンソールのページを選択します。LTE-M ボタンでは、携帯電話ネットワークを使用するため、この手順は不要です。
4. 選択オンボード[] を選択してから、[クレームデバイス。
5. 1 つ以上のデバイス ID (デバイスのシリアル番号など) またはクレームコードをコンマで区切って選択し、Claim。そのファイルにClaimボタンが使用できない場合は、入力したすべての値を再確認します。
6. デバイスのボタンを押し、を選択します。Done。既知のすべてのデバイスのリストが表示されます。

プロジェクトの作成

次の手順では、AWS IoT 1-Click でサポートされているデバイスの AWS IoT 1-Click プロジェクトを作成します。

1. AWS アカウントにサインインし、AWS IoT 1-Click コンソールを開きます。

2. 選択オンボード[] を選択してから、[プロジェクトを作成します。。
3. プロジェクトの名前とオプションの説明を入力し、を選択します。次。
4. プレイスメント用に 1 つまたは複数のテンプレートを定義するには、デバイステンプレートをプログラムする] で、[を起動。
5. 任意のボタンデバイスのテンプレートを定義するには、すべてのボタン型。
6. [デバイステンプレート名] に、テンプレートのわかりやすい名前を入力します。[アクション] で、[SMS を送信] または [E メールを送信] を選択します。「」を使用できます。Lambda 関数を使用したカスタムアクションオプションを選択し、独自の Lambda 関数のいずれかを選択します。選択に応じて、電話番号、E メールアドレス、または Lambda 関数名を入力します。Lambda 関数の作成の詳細については、を参照してください。AWS Lambda デベロッパーガイド。
7. []他のデバイステンプレートの追加 (配置ごとに複数のデバイスがある場合)選択するを追加します。。
8. 属性キーと値のペアを入力します。必要に応じて、追加のキーと値のペアを入力できます。
9. [Create project (プロジェクトの作成)] を選択します。

次のセクション例: Meeting Room Satisfaction Projectでは、AWS IoT 1-Click コンソールを使用してプロジェクトを作成する方法の実際の例を示します。

例: Meeting Room Satisfaction Project

以下の例は、AWS IoT 1-Click の概念を理解するのに役立ちます。

- 50 の 会議室 (および関連する AV 機器) の満足度を追跡するプロジェクトが作成され、`MeetingRoomSat` という名前が付けられます。
- 各会議室には、1 つは物理的に「満足」とマークされ、もう 1 つは「不満」とマークされた 2 つのデバイス (ボタン) が表示されます。部屋ごとに 2 つのボタンがあるため、1 つは `Satisfied`、もう 1 つは `Unsatisfied` という名前の 2 つのテンプレートが作成されます。
- -`Satisfied`テンプレートは、という名前の Lambda 関数を呼び出すように設定されています。`SatLambda`。
- -`Unsatisfied`テンプレートは、という名前の Lambda 関数を呼び出すように設定されています。`UnsatLambda`。
- これらの両方のテンプレートで、値が `MeetingRoomNum` の TBD (キー) という名前の属性 (キー/値ペア) が作成されます (両方のボタンが物理的に部屋に配置されている場合は、TBD 値は部屋番号に変更されます)。

- 50 のプレイスメントが作成され、各部屋に 1 つずつ作成されます。各プレイスメントには、関連付けられた 2 つのテンプレートがあります (例: `Satisfied` および `Unsatisfied`)。
- 2 つのボタンは物理的にラベル付けされ、部屋に配置されます。次に、AWS IoT 1-Click モバイルアプリケーションまたは AWS IoT 1-Click コンソールおよびボタンのシリアルナンバーを使用して、「満足」および「不満」とマークされたボタンが 50 のプレイスメントの 1 つに関連付けられます。このプロセスは、残りのすべてのプレイスメントがデプロイされるまで続きます。
- 会議室で部屋ボタンをクリックすると、AWS IoT 1-Click は、`SatLambda`または`UnsatLambda`関数を`MeetingRoomNum`の値から、フィードバックを処理し、クラウド上に保存することができます。
- 後で別のテンプレートをプロジェクトに追加して、バスルームにもっと多くのタオルやその他のアメニティが必要であることを示す新しいボタンのスロットを 50 個の既存のプレイスメントが持つようにできます。

AWS IoT 1-Click コンソールを使用して、オフィスビル内 (オフィスビル群の一部として) の会議室の満足度をモニタリングするプロジェクトを作成する例を示します。

オーディオ/ビデオ機器を含む会議室の満足度をモニタリングするために、各会議室に 2 つの AWS IoT エンタープライズボタンが配置されます。1 つは「満足」、もう 1 つは「不満」です。これはパイロットプロジェクトであり、その結果を使用して、キャンパス内の他のビルの会議室の顧客満足度を向上させることができます。

会議が終了した後、参加者は、会議室とその関連機器の全体的な満足度を記録するために、「満足」ボタンまたは「不満」ボタンのどちらかを押すよう促されます。その後、このデータは、機能しない A/V 機器やその他の問題のある会議室を特定するために使用されます。

AWS IoT 1-Click コンソールは、次のように、このプロジェクトを設定するために使用できます。

1. AWS IoT 1-Click コンソールから、を選択します。プロジェクトを作成します。。
2. プロジェクト名に、**MeetingRoomSatisfaction** と入力します。プロジェクトの説明に、と入力します。**Project used to track customer meeting room satisfaction, including A/V equipment.**選択次。
3. []デバイステンプレートをプログラムする選択するを起動[] を選択してから、[すべてのボタン型。
4. [デバイステンプレート名] に **Satisfied** と入力します。これは、「満足」というラベルの付いたすべてのボタンに使用されるテンプレートです。[アクション] で、[E メールを送信] を選択します。

> **Note**
>
> 会議室の満足度パイロットが成功した場合は、[アクション] で [Custom action using a Lambda function (Lambda 関数を使用したカスタムアクション)] を選択します。このカスタム Lambda 関数は E メールを送信したり、後で分析するために「満足」ボタンデータを Amazon DynamoDB テーブルに格納することができます。Lambda 関数の作成については、を参照してください。AWS Lambda デベロッパーガイド。

5. []他のデバイステンプレートの追加 (配置ごとに複数のデバイスがある場合)選択するを追加します。[] を選択してから、[すべてのボタン型。[デバイステンプレート名] に **Unsatisfied** と入力します。これは、「不満」というラベルの付いたすべてのボタンに使用されるテンプレートです。[アクション] で、[E メールを送信] を選択します。
6. [Required email default value (必須の E メールデフォルト値)] に E メールアドレスを入力します。[Required subject default value (必須の件名デフォルト値] に **Meeting Room Feedback** と入力します。[Required body default value (必須の本文デフォルト値)] に **Either positive or negative meeting room feedback has been provided.** と入力します。
7. [Attribute key (属性キー)] に **Building** と入力します。[デフォルト値] に、**Headquarters** と入力します。会議室満足度パイロットは、会社の本社ビルで行われています。このパイロットが成功した場合、会社の他のビルにデプロイされます。したがって、情報がどのビルの会議室のデバイスからのものかを認識することが重要です。
8. 2 番目のキーと値ペアの行で、[Attribute key (属性キー)] に **Room**] と入力します。[デフォルト値] に、**TBD** と入力します。-**TBD**ボタンが配置されると、値は、会議室番号に変更されます (AWS IoT 1-Click モバイルアプリまたは AWS IoT 1-Click コンソールを使用)。
9. [Create project (プロジェクトの作成)] を選択します。

AWS IoT 1-Click モバイルアプリを使用して、会議室に「満足」ボタンが配置されると、満足テンプレートが関連付けられ、**TBD**の値は、会議室番号に置き換えられます。会議室に「不満」ボタンが配置される場合も同様です。

AWS IoT 1-Click モバイルアプリ

AWS IoT 1-Click モバイルアプリを使用すると、次のことができます。

- AWS IoT 1-Click コンソールと同様のユーザーインターフェイスを使用して、フィールド内の AWS IoT 1-Click デバイスを便利に設定して監視するには。
- (AWS IoT IoT エンタープライズボタンなどの) Wi-Fi 接続 AWS IoT 1-Click デバイスの Wi-Fi 認証情報を設定するには。

AWS IoT 1-Click モバイルアプリは iPhone と Android のモバイルデバイスで使用できます。アプリをダウンロードするには、App StoreまたはGoogle Play(AWS IoT 1-Click で検索するには。

AWS IoT 1-Click プログラミングモデル

AWS IoT 1-Click デバイスを使用してアプリケーションを構築するには、プログラマーはAWS IoT 1-Click デバイス APIとAWS IoT 1-Click プロジェクトの API。デバイス API は AWS IoT 1-Click デバイスコンポーネントとやり取りし、デバイスからのイベントを処理します。これらのイベントには、デバイスの有効化と無効化、イベントフォーマットの定義、およびそれらがトリガーするアクション (Lambda 関数) が含まれます。デバイス API は、製造元がデバイスを登録したリージョンに存在している AWS コンポーネントと緊密に連携しています。これは、AWS デバイスリージョンのお客様がデバイスを使用しているリージョンと異なる場合があります。プロジェクト API は AWS IoT 1-Click プロジェクトサービスとやり取りし、AWS IoT 1-Click デバイスを合計して管理するために使用されます。これにより、次のことが可能になります。

- デバイスをプロジェクトにグループ化します。
- プロジェクト内のすべてのデバイスのアクションを設定するために使用するテンプレートを作成します。
- プロジェクトに関連するコンテキストデータを格納する属性を定義します。

AWS IoT 1-Click プログラミングモデルを使用して、デバイス API を使用して個々のデバイスをプログラムできます。この場合、AWS IoT 1-Click デバイスタイプを使用します。API は、標準イベント形式とそのタイプのすべてのデバイスのプログラミングインターフェイスを形成するメソッドのリストを定義します。特定のデバイスタイプに関連するメソッドを呼び出すために、プログラマーは、`InvokeDeviceMethod API` を使用し、パラメータとしてデバイスメソッドを指定できます。

たとえば、デバイスタイプ「ボタン」を持つすべての AWS IoT 1-Click デバイスは、クリックに関連するイベントを発行し、デバイスがクリックされたときに呼び出されるコールバック関数を設定するメソッドを持っています。ボタンインターフェイスの詳細については、「デバイスタイプごとのインターフェイス」を参照してください。このコールバック関数を設定するコードは次のとおりです。

```
    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);
```

プロジェクト API を使用して多数のデバイスをプログラムします。API を使用して、最初に、各プレイスメントのデバイステンプレートや属性など、各プレイスメントの外観を定義します。完了したら、特定のデバイス ID でプレイスメントを作成します。各プレイスメントは同じテンプレートに従います。これを行うためのサンプルコードは次のとおりです。

```
    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 コールバックイベント

AWS IoT 1-Click では、コールバックを登録することでデバイスイベントをサブスクライブできます。コールバックの例としては、AWS IoT 1-Click のお客様が所有し実装する AWS Lambda 関数があります。このコールバックは、イベントが発生するたびに呼び出され、使用されます。イベント

とそのペイロードの詳細については、AWS IoT 1-Click イベント セクションおよび AWS IoT 1-Click Health スイベント セクションを参照してください。

AWS IoT 1-Click イベント

button タイプのデバイスは、クリックされるたびにクリックイベントを発行します。このイベントにサブスクライブするには、次の方法があります。

- デバイス上のデバイス SetOnClickCallback メソッドを呼び出します。
- 前述のプロジェクトコード作成例に示すように、関連するプロジェクトを適切に設定します。

以下の例では、placementInfo セクションは、デバイスに関連付けられたプレイスメントがある場合にのみ存在することに注意してください。詳細については、「「プロジェクト、テンプレート、およびプレイスメント」」を参照してください。

```
{
    "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 スイベント

AWS IoT 1-Click サービスによって計算されたヘルスパラメータに基づいてヘルスイベントを発行しますが、対応するしきい値を設定します。次の例では、残りライフが 10% ("remainingLifeLowerThan":10 というキーと値のペアに注目) であるデバイス G030PMXXXXXXXXXX のヘルスイベントの JSON ペイロードを表します。

```
{
    "deviceEvent": {
      "deviceHealthMonitor": {
        "condition": {
          "remainingLifeLowerThan": 10
        }
      }
    },
    "deviceInfo": {
      "attributes": {
        "key2": "value2",
        "key1": "value1",
        "projectRegion": "us-west-2"
      },
      "type": "button",
      "deviceId": "G030PMXXXXXXXXXX",
      "remainingLife": 5.4
    }
  }
```

デバイスのメソッド

AWS IoT 1-Click デバイスメソッドは、次の表に示す、特定のデバイスの種類でサポートされている API です。任意のデバイスでサポートされているデバイスメソッドの詳細な一覧は、GetDeviceMethods を呼び出して取得できます。

デバイスタイプ	メソッド名	説明
device	getDeviceHealthParameters	remainingLife など、デバイスのヘルスパラメータを取得します。
device	setDeviceHealthMonitorCallback	デバイスのヘルスパラメータがしきい値を下回ったときに呼び出されるようにコールバックを設定します。
device	getDeviceHealthMonitorCallback	ヘルスパラメータがしきい値を下回ったときに呼び出される、設定済みのコールバックを取得します。
button	setOnClickCallback	ボタンがクリックされたときに呼び出されるようにコールバックを設定します。
button	getOnClickCallback	ボタンをクリックしたときに呼び出される、設定済みのコールバックを取得します。

Amazon CloudWatch を使用した AWS IoT 1-Click のモニタリング

ユーザーの代わりに AWS IoT 1-Click がデバイスを監視し、Amazon CloudWatch。これらのメトリックスは、製造元が登録したデバイスのあるデバイスリージョンにレポートされます。デバイスリージョンの詳細については、」AWS IoT 1-Click のしくみ。Amazon CloudWatch ダッシュボードのIOT名前空間。

Amazon CloudWatch Events を使用すると、AWS のサービスを自動化し、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。AWS のサービスからのイベントは、ほぼリアルタイムに CloudWatch イベントに配信されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。トリガーできる以下のアクション。

- AWS Lambda 関数の呼び出し。
- Amazon EC2 実行コマンドの呼び出し
- Amazon Kinesis Data Streams へのイベントの中継。
- AWS Step Functions ステートマシンのアクティブ化。
- Amazon SNS トピックまたは AWS SMS キューへの通知。

AWS IoT 1-Click は、次のメトリクスを追跡して報告します。

- [TotalEvents] はデバイスが発行するイベントの数を追跡します。このメトリックスは、デバイスイベント、プロジェクト、デバイスタイプ、製品タイプごとに表示し、グラフ化できます。
- RemainingLifeは、デバイスの残りのライフの概算パーセントを表します。AWS IoT 1-Click は、デバイスの製造元の評価に基づいてこの数を報告します。たとえば、ボタンが約 2000 回のクリックに耐えるように設計されており、500 回 のクリックが記録されている場合は、[RemainingLife] の値は 75% と報告されます。[RemainingLife] メトリクスはプロジェクト、デバイスタイプ、製品タイプごとに表示し、グラフ化できます。顧客は、[RemainingLife] メトリクスを使用して、デバイスが特定のしきい値を下回ったときにトリガーされるアラームを設定できます。その後、顧客はRemainingLifeを使用して、デバイスのGetDeviceHealthParameters低いデバイスを識別する方法RemainingLife値.
- [CallbackInvocationErrors] は、デバイスがイベントを発行したときにコールバック (Lambda 関数) を呼び出す際のエラーを追跡します。[CallbackInvocationErrors] メトリクスは、呼び出されたコールバック (コールバックとして設定された Lambda 関数 ARN) またはプロジェクトごとに表示し、

グラフ化できます。お客様は、CallbackInvocationErrorsAWS IoT 1-Click がデバイスから設定した Lambda 関数にイベントをルーティングできなかった場合に通知を受け取ることができます。

詳細については、Amazon CloudWatch Events ユーザーガイドを参照してください。

AWS CloudTrail による AWS IoT 1-Click API 呼び出しの口グ記録

AWS IoT 1-Click は、AWS CloudTrail と統合されます。AWS CloudTrail は、AWS IoT 1-Click のユーザー、ロール、または AWS のサービスで実行されたアクションのレコードを提供するサービスです。CloudTrail は、AWS IoT 1-Click の API 呼び出しをイベントとしてキャプチャします。キャプチャされたコールには、AWS IoT 1-Click コンソールからのコールと、AWS IoT 1-Click API オペレーションへのコードコールが含まれます。証跡を作成する場合は、AWS IoT 1-Click のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [Event history (イベント履歴)] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、AWS IoT 1-Click に対するリクエスト、リクエスト元の IP アドレス、リクエストの実行者、リクエスト日時などの詳細を確認できます。

CloudTrail を設定して有効にする方法などの詳細については、AWS CloudTrail ユーザーガイドを参照してください。

CloudTrail の AWS IoT 1-Click 情報

CloudTrail は、アカウント作成時に AWS アカウントで有効になります。AWS IoT 1-Click でサポートされているイベントアクティビティが発生すると、そのアクティビティはイベント履歴。最近のイベントは、AWS アカウントで表示、検索、ダウンロードできます。詳細については、「CloudTrail イベント履歴でのイベントの表示」を参照してください。

AWS IoT 1-Click のイベントなど、AWS アカウントのイベントの継続的な記録については、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで作成した証跡がすべての AWS リージョンに適用されます。証跡では、AWS パーティションのすべてのリージョンからのイベントがログに記録され、指定した Amazon S3 バケットにログファイルが配信されます。さらに、その他の AWS サービスを設定して、CloudTrail ログで収集されたデータをより詳細に分析し、それに基づく対応を行うことができます。詳細については、以下を参照してください。

- 証跡を作成するための概要
- CloudTrail のサポート対象サービスと統合
- Amazon SNS の CloudTrail 通知の設定
- 「複数のリージョンから CloudTrail ログファイルを受け取る」および「複数のアカウントから CloudTrail ログファイルを受け取る」

AWS IoT 1-Clickデバイス APICloudTrail ログファイルのイベントとして以下のアクションを記録します。

- ListDevices
- DescribeDevice
- GetDeviceMethods
- UpdateDeviceState
- InvokeDeviceMethod

AWS IoT 1-Clickプロジェクト APICloudTrail ログファイルのイベントとして以下のアクションを記録します。

- CreateProject
- UpdateProject
- DescribeProject
- ListProjects
- DeleteProject
- CreatePlacement
- UpdatePlacement
- DescribePlacement
- ListPlacements
- DeletePlacement
- AssociateDeviceWithPlacement
- DisassociateDeviceFromPlacement
- GetDevicesInPlacement

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この ID 情報は以下のことを確認するのに役立ちます。

- リクエストが、ルートまたは AWS Identity and Access Management (IAM) ユーザーの認証情報のどちらを使用して送信されたかどうか。
- リクエストが、ロールとフェデレーティッドユーザーのどちらの一時的なセキュリティ認証情報を使用して送信されたか.

- リクエストが AWS の別のサービスによって生成されたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

例: AWS IoT 1-Click ログファイルのエントリ

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail ログファイルには、1 つ以上のログエントリがあります。イベントは任意の発生元からの 1 つのリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次は、`DescribeDevice` アクションを示す CloudTrail ログエントリの例です。

```
{
    "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"
}
```

次は、CreateProject アクションを示す CloudTrail ログエントリの例です。

```
{
    "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": "2018-04-12T20:31:02Z",
    "eventSource": "iot1click.amazonaws.com",
    "eventName": "CreateProject",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "console.aws.amazon.com",
    "requestParameters": {
        "description": "",
        "placementTemplate": {
            "defaultAttributes": "***",
            "deviceTemplates": {
                "happyId": {
                    "deviceType": "button",
                    "callbackOverrides": {
                        "onClickCallback": "arn:aws:lambda:us-
west-2:012345678910:function:rating_buttons_happy"
                    }
                },
                "sadId": {
                    "deviceType": "button",
                    "callbackOverrides": {
                        "onClickCallback": "arn:aws:lambda:us-
west-2:012345678910:function:rating_buttons_sad"
                    }
                }
            }
        }
    }
}
```

AWS CloudFormation の統合

AWS IoT 1-Click は AWS CloudFormation と統合されており、AWS CloudFormation はと統合されており、クラウド環境 (Amazon EC2、Auto Scaling、Amazon SNS など) ですべてのインフラストラクチャリソースを記述およびプロビジョニングするための共通言語です。AWS CloudFormation では、シンプルなテキストファイルを使用して、すべてのリージョンとアカウントにわたって、アプリケーションに必要なすべてのリソースを自動的かつ安全な方法でモデル化し、プロビジョニングできます。このファイルは、クラウド環境における信頼できる唯一の情報源として機能します。詳細については、「」を参照してください。AWS CloudFormation ユーザーガイドAWS IoT 1-Click のトピック (AWS::IoT1Click::Project) のAWS CloudFormation ユーザーガイド。

AWS IoT 1-Click の認証とアクセスコントロール

AWS IoT 1-Click API にアクセスするには、認証情報が必要です。これらの認証情報には、AWS IoT 1-Click プロジェクトやデバイスなど、AWS リソースにアクセスするための権限が必要です。以下のセクションでは、AWS アイデンティティとアクセス管理 (IAM) と AWS IoT 1-Click を使用してリソースへのアクセスを保護する方法について詳しく説明します。

すべての AWS リソースは AWS アカウントによって所有され、となり、リソースの作成またはアクセスは、アクセス許可のポリシーによって管理されます。アカウント管理者は、アクセス権限ポリシーを IAM アイデンティティ (ユーザー、グループ、ロール) にアタッチできます。一部のサービス (AWS Lambda など) もリソースにアクセス権限ポリシーをアタッチすることができます。アカウント管理者はアクセス権限を付与する際に、付与先のユーザー、対象のリソース、および対象のリソースに許可されるアクションを決定します。

AWS IoT 1-Click リソースおよびオペレーション

AWS IoT 1-Click では、主なリソースはプロジェクトとデバイスです。ポリシーで Amazon リソースネーム (ARN) を使用して、ポリシーを適用するリソースを識別します。これらのリソースには、次の表に示すとおり、一意の Amazon リソースネーム (ARN) が関連付けられています。

リソースタイプ	ARN 形式
デバイス	arn:aws:iot1click:region:account-id:devices/device-id
プロジェクト	arn:aws:iot1click:region:account-id:projects/project-name

AWS IoT 1-Click は、AWS IoT 1-Click リソースで動作する API を実装します。これらは IAM におけるアクションと呼ばれます。使用可能な操作のリストについては、このトピックの最後の表を参照してください。

AWS IoT 1-Click でアイデンティティベースのポリシー (IAM ポリシー) を使用する

このトピックで取り上げる ID ベースのポリシーの例では、アカウント管理者が IAM ID (ユーザー、グループ、ロール) にアクセス許可ポリシーをアタッチし、AWS IoT 1-Click リソースに対するオペレーションを実行するアクセス許可を付与する方法を示しています。

以下に示しているのは、アクセス権限ポリシーの例です。

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

ポリシーには 1 つのステートメントがあり、AWS IoT 1-Click アクションに対するアクセス権限を付与します (`iot1click:CreateProject`) をアプリケーションの Amazon リソースネーム (ARN) を使用してリソースに割り当てます。この場合の ARN はワイルドカード文字 (*) を指定して、どのリソースにもアクセス権限を付与することを示します。

AWS IoT 1-Click の 1-Click の API オペレーションとそれらが適用されるリストについては、AWS IoT 1-Click の API アクセス権限: アクション、アクセス許可、およびリソースのリファレンス。

AWS IoT 1-Click の AWS 管理 (定義済み) ポリシー

Amazon Web Services、AWS によって作成され管理されるスタンドアロンの IAM ポリシーを提供することで、多くの一般的ユースケースに対応します。これらの AWS 管理ポリシーは、一般的ユースケースに必要なアクセス権限を付与することで、どの権限が必要なのかをユーザーが調査する必要をなくすことができます。詳細については、IAM ユーザーガイドの「AWS 管理ポリシー」を参照してください。

アカウントのユーザーにアタッチ可能な以下の AWS 管理ポリシーは、AWS IoT 1-Click に固有のもので、ユースケースシナリオ別にグループ化されます。

- `AWSIoT1ClickFullAccess`: AWS 管理コンソールを使用して AWS IoT 1-Click リソースへのフルアクセスを許可します。付与されるアクセス権限には、デバイスとプロジェクトを管理するためのすべての AWS IoT 1-Click アクションが含まれます。
- `AWSIoT1ClickReadOnlyAccess`: AWS 管理コンソールを使用して AWS IoT 1-Click リソースへの読み取り専用アクセスを許可します。このアクセス許可により、ユーザーは AWS IoT 1-Click デバイスとプロジェクトをリストしてプロジェクト設定を確認することができます。

> **Note**
>
> これらのアクセス権限ポリシーは、IAM コンソール (https://console.aws.amazon.com/iam/) に特定のポリシー名を検索します。

独自のカスタム IAM ポリシーを作成して、AWS IoT 1-Click アクションとリソースのための権限を許可することもできます。これらのカスタムポリシーは、それらのアクセス権限が必要な IAM ユーザーまたはグループにアタッチできます。

AWS IoT 1-Click の API アクセス権限: アクション、アクセス許可、およびリソースのリファレンス

AWS クラウドでアクセスコントロールをセットアップし、IAM ID (アイデンティティベースのポリシー) にアタッチできるアクセス権限ポリシーを作成する際、以下の表をリファレンスとして使用できます。この表は、各 AWS IoT 1-Click API オペレーション、アクションを実行するためのアクセス権限を付与できる対応するアクション、およびアクセス権限を付与できる AWS リソースを示しています。ポリシーの `Action` フィールドでアクションを指定し、ポリシーの `Resource` フィールドでリソースの値を指定します。

AWS IoT 1-Click ポリシーで AWS IoT 1-Click ポリシーを使用して、条件を表現することができます。AWS 全体を対象とするすべてのキーのリストについては、「」を参照してください。使用できるキー()IAM ユーザーガイド。

> **Note**
>
> アクションを指定するには、API オペレーション名 (`iot1click:` など) の前に `iot1click:ListProjects` プレフィックスを使用します。

IoT 1-Click オペレーション	必要なアクセス権限 (API アクション)	リソース
ListDevices	iot1click:ListDevices	*
DescribeDevice	iot1click:DescribeDevice	arn:aws:iot1click:region:account-id:devices/device-id
GetDeviceMethods	iot1click:GetDeviceMethods	arn:aws:iot1click:region:account-id:devices/device-id
UpdateDeviceState	iot1click:UpdateDeviceState	arn:aws:iot1click:region:account-id:devices/device-id
InvokeDeviceMethod	iot1click:InvokeDeviceMethod	arn:aws:iot1click:region:account-id:devices/device-id
ListDeviceEvents	iot1click:ListDeviceEvents	arn:aws:iot1click:region:account-id:devices/device-id
InitializeDeviceClaim	iot1click:InitializeDeviceClaim	arn:aws:iot1click:region:account-id:devices/device-id
FinalizeDeviceClaim	iot1click:FinalizeDeviceClaim	arn:aws:iot1click:region:account-id:devices/device-id
UnclaimDevice	iot1click:UnclaimDevice	arn:aws:iot1click:region:account-id:devices/device-id
ClaimDeviceByClaimCode	iot1click:ClaimDeviceByClaimCode	*
CreateProject	iot1click:CreateProject	arn:aws:iot1click:region:account-id:projects/project-name
UpdateProject	iot1click:UpdateProject	arn:aws:iot1click:region:account-id:projects/project-name

IoT 1-Click オペレーション	必要なアクセス権限 (API アクション)	リソース
DescribeProject	iot1click:DescribeProject	arn:aws:iot1click:region:account-id:projects/project-name
ListProjects	iot1click:ListProjects	*
DeleteProject	iot1click:DeleteProject	arn:aws:iot1click:region:account-id:projects/project-name
CreatePlacement	iot1click:CreatePlacement	arn:aws:iot1click:region:account-id:projects/project-name
UpdatePlacement	iot1click:UpdatePlacement	arn:aws:iot1click:region:account-id:projects/project-name
DescribePlacement	iot1click:DescribePlacement	arn:aws:iot1click:region:account-id:projects/project-name
ListPlacements	iot1click:ListPlacements	arn:aws:iot1click:region:account-id:projects/project-name
DeletePlacement	iot1click:DeletePlacement	arn:aws:iot1click:region:account-id:projects/project-name
AssociateDeviceWithPlacement	iot1click:AssociateDeviceWithPlacement	arn:aws:iot1click:region:account-id:projects/project-name
DissacociateDeviceFromPlacement	iot1click:DissacociateDeviceFromPlacement	arn:aws:iot1click:region:account-id:projects/project-name
GetDevicesInPlacement	iot1click:GetDevicesInPlacement	arn:aws:iot1click:region:account-id:projects/project-name

AWS IoT 1-Click リソースにタグを付ける

AWS IoT 1-Click リソースを管理しやすくするために、タグを使用して任意の ARNベースのリソースに独自のメタデータを割り当てることができます。この章では、タグとその作成方法について説明します。

タグの基本

タグを使用すると、AWS IoT 1-Click リソースを目的、所有者、環境などさまざまな形で分類できます。これは、同じ型のリソースが多数ある場合に役立ちます。割り当てたタグに基づいて特定のリソースをすばやく検索および識別できます。タグはそれぞれ、1 つのキーとオプションの値で構成され、どちらもユーザーが定義します。たとえば、特定の管理者またはアカウントが所有する複数のボタンに対して、タグのセットを定義できます。追加したタグに基づいてリソースを検索およびフィルタリングできます。ニーズを満たす一連のタグキーをリソースタイプごとに考案されることをお勧めします。一貫性のあるタグキーセットを使用することで、リソースの管理が容易になります。詳細については、「AWS タグ付け戦略」を参照してください。

また、AWS コストを分類して追跡するためにもタグを使用できます。タグをリソースに適用すると、AWS はタグ別に利用量とコストを集計したカンマ区切り値 (CSV) ファイルとしてコスト配分レポートを作成します。自社のカテゴリ (たとえばコストセンター、アプリケーション名、所有者) を表すタグを適用すると、複数のサービスにわたってコストを分類することができます。コスト配分のためのタグの使用の詳細については、コスト配分タグの使用()AWS Billing and Cost Management ユーザーガイド。

使いやすいように、AWS マネジメントコンソールでタグエディタを使用すると、統一された方法で一元的にタグを作成および管理できます。詳細については、「」を参照してください。タグエディターの使用がAWS マネジメントコンソールの開始方法。

AWS CLI および AWS IoT 1-Click デバイスおよびプロジェクト API を使用してタグを操作することもできます。タグを作成するときに AWS IoT 1-Click プロジェクトおよびデバイスをタグに関連付けるには、タグ以下のコマンドで入力します。

- CreateProject (プロジェクト API)
- 要求の確定 (デバイス API)

既存のリソースのタグを追加、変更あるいは削除するには、次のコマンドを使用できます。

AWS IoT 1-Clickプロジェクト API(プロジェクト ARN を使用)	AWS IoT 1-Clickデバイス API(デバイス ARN を使用)
TagResource	Tag
ListTagsForResource	TagResource （POST)、TagListTagsForResource （GET)、UntagResource （DELETE) を参照してください。
UntagResource	

タグのキーと値は編集でき、タグはリソースからいつでも削除できます。タグの値を空の文字列に設定することはできますが、タグの値を null に設定することはできません。特定のリソースについて既存のタグと同じキーを持つタグを追加した場合、古い値は新しい値によって上書きされます。リソースを削除すると、リソースに関連付けられているすべてのタグも削除されます。

タグの制限

タグには以下のような基本制限があります。

- リソースあたりのタグの最大数 – 50
- キーの最大長:127 文字 (Unicode) (UTF-8)
- 値の最大長:255 文字 (Unicode) (UTF-8)
- タグのキーと値は大文字と小文字が区別されます。
- タグの名前または値に aws: プレフィックスは使用しないでください。このプレフィックスは AWS 用に予約されています。このプレフィックスが含まれるタグの名前または値は編集または削除できません。このプレフィックスを持つタグは、リソースあたりのタグ数の制限時には計算されません。
- 複数のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスでも許可される文字に制限が適用されることがあるのでご注意ください。一般的に使用が許可される文字は、UTF-8 で表現できる文字、スペース、数字と、特殊文字 (+ - = . _ : / @) です。

AWS IoT Enterprise Button ユーザーガイド

AWS IoT Enterprise Button は、シンプルで簡単に Wi-Fi ベースのボタンを設定できます。企業や開発者が、AWS IoT 1-Click を使用して既存のビジネスフローやシステムに簡単に統合できるように設計されています。

AWS IoT Enterprise Button では、次の 3 種類のクリックがサポートされます。

- 単一
- Double
- 長押し

正しく機能させるには、AWS IoT 1-Click モバイルアプリ (iOS または Android) を使用してボタンの Wi-Fi 接続を設定する必要があります。アプリでは、AWS アカウントにログインするか、アプリの右上にある Wi-Fi アイコンをタップしてログインをスキップすることで、ボタンの Wi-Fi 接続を設定できます。

接続が設定され、モバイルアプリまたはコンソール経由で要求されると、シングルクリック、ダブルクリック、または長押しされた際に、ボタンが緑色に点灯します。

設定後にボタンに問題があると思われる場合は、この表でトラブルシューティングを行うことができます。

カラー	ステータス	推奨事項
白で点滅	Wi-Fi に接続して IP アドレスを取得中、または AWS IoT に接続中。	該当なし
緑で点灯	正常に Wi-Fi に接続され、AWS IoT にメッセージが発信されました。	該当なし
青で点滅	ボタンは設定モードです。	設定プロセスが完了するまで待機します。

カラー	ステータス	推奨事項
オレンジで点灯	Wi-Fi が設定されていません。	AWS IoT 1-Click モバイルアプリを使用して Wi-Fi を設定します。
赤: 短、短、短	設定されたワイヤレスネットワークへの接続時にエラーが発生しました。	ネットワーク設定が変更されたかどうか、またはボタンが Wi-Fi ルーターから離れすぎていないか確認します。
赤: 短、短、長	ワイヤレスネットワークからの IP アドレスの取得時にエラーが発生しました。	ワイヤレスネットワークの問題を確認します。
赤: 短、長、短	ホスト名ルックアップの実行中にエラーが発生しました。	ワイヤレスネットワークの問題を確認します。
赤: 短、長、長	AWS IoT に接続できません。	ワイヤレスネットワークの問題を確認します。ネットワークの問題がなく、問題が解決しない場合は、にお問い合わせください。AWS サポートセンター[] に、デバイスのシリアル番号 (DSN) を提供します。ボタンの裏側にあります。
赤: 長、短、短	サーバーとセキュアな接続を確立できません。	AWS IoT 1-Click iOS または Android のモバイルアプリを使用して、最新のファームウェアがあるかどうかを確認します。

カラー	ステータス	推奨事項
赤: 長、短、長	HTTP 403 forbidden エラーを受信しました。	にお問い合わせください。AWS サポートセンターに問い合わせるには、DSN を提供します。ボタンの裏側にあります。
赤: ボタンを 15 秒押した後	ボタンはリセットされます。	ボタンを 15 秒間押すと、AWS IoT エンタープライズボタン Wi-Fi 設定をリセットすることができます。

AWS IoT 1-Click を使用して、AWS CLI を使用します。

AWS コマンドラインインターフェイス (AWS CLI) の使用方法を示すため、AWS IoT 1-Click を使用してごみ収集サービスを合理化しようとするごみ処理会社のシナリオを検討します。

このシナリオでは、ごみ箱は AWS IoT Enterprise Button とペアになっています。ごみ箱がいっぱいになった場合は、関連付けられたボタンを押すだけで、ごみ箱の交換をリクエストできます。

> **Note**
>
> すべての AWS IoT Enterprise Button デバイス ID は「G030PM」で始まります。

ごみ処理会社は、AWS IoT Enterprise Button を準備するために、次の手順を使用します。

AWS IoT Enterprise Button 顧客用に準備するには

1. AWS IoT Enterprise Button 用の Wi-Fi を設定する唯一の方法は、AWS IoT 1-Click モバイルアプリケーションを使用することです。アプリをインストールするには、「AWS IoT 1-Click モバイルアプリ」を参照してください。アプリをインストールした後、(通常の場合と同様に) [AWS アカウントにログイン] をクリックしないでください。この演習では、AWS CLI の使用方法を示します。[AWS アカウントにログイン] をクリックすると、initiate-device-claim およびfinalize-device-claim コマンドが呼び出されます。次の手順に示すように、CLI を使用して「手動で」これを実行します。
2. AWS CLI のデモンストレーションの目的で、AWS アカウントへのログイン右上隅にある小さな丸い Wi-Fi アイコンを選択します。次に、[Wi-Fi の設定] を選択します。デバイス ID をスキャンまたは入力し、モバイルアプリの残りの指示に従います。
3. AWS CLI がインストールされていない場合は、の手順に従います。AWS CLI のインストール。使用可能な AWS IoT 1-Click AWS CLI コマンドを一覧表示するには、次の 2 つのコマンドを実行します。

   ```
   aws iot1click-projects help
   ```

   ```
   aws iot1click-devices help
   ```

4. Wi-Fi に接続された AWS IoT Enterprise Button をごみ処理会社の AWS アカウントに関連付けるには、デバイスのデバイス IDを使用して次のコマンドを実行します。

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

デバイスのボタンを押します。白いライトが断続的に点滅した後、約 1 秒間、緑色のライトが点灯します。そうでない場合は、前述の Wi-Fi の接続手順を繰り返します。

5. 前の手順で緑色のライトが点灯したら、次のコマンドを実行します (デバイス ID 値を使用)。

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

-"State": "CLAIMED"レスポンスは、デバイスが AWS IoT 1-Click サービスに正常に登録されたことを示します。

> **Note**
>
> 次の例に示すように、デバイスの製造元が「C-」で始まる要求コードを提供した場合、aws iot1click-devices claim-devices-by-claim-code コマンドだけを使用して、単一の要求コードを使用して 1 つ以上のデバイスを要求することができます。
>
> ```
> aws iot1click-devices claim-devices-by-claim-code --claim-code C-123EXAMPLE
> {
>     "Total": 9
>     "ClaimCode": "C-123EXAMPLE"
> }
> ```
>
> この例では、"Total": 9は、クレームコードに関連付けられた 9 つのデバイスです。C-123EXAMPLEAWS IoT 1-Click サービスによって正常に要求されました。

6. 次に、という名前の JSON テキストファイルを作成して、ごみ処理会社に適切な AWS IoT 1-Click プロジェクトを作成する準備をします。create-project.json。ファイルには、次のものが含まれています。

```
{
   "projectName": "SeattleDumpsters",
```

```
   "description": "All dumpsters in the Seattle region.",
   "placementTemplate": {
      "defaultAttributes": {
         "City" : "Seattle"
      },
      "deviceTemplates": {
         "empty-dumpster-request" : {
            "deviceType": "button"
         }
      }
   }
}
```

placementTemplate と deviceTemplates のキーと値のペアは、SeattleDumpsters プロジェクトの一部であるすべてのボタンに適用される属性です。このプロジェクトを作成するには、次のコマンドを実行します。create-project.jsonにあるカレント作業ディレクトリです。AWS CLI のコマンドプロンプトです。

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

新しく作成したプロジェクトを表示するには、次のコマンドを実行します。

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

詳細については、次のように describe-project コマンドを実行します。

```
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. シアトル地域用に作成したプロジェクトを使用して、次に示すように特定のごみ箱 (お客様 217 用) のプレイスメントを作成します。Windows ではエスケープされた引用符が必要です。

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

新しく作成されたプレイスメントを表示するには、次のコマンドを実行します。

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

詳細については、次のように describe-placement コマンドを実行します。

```
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,
        "updatedDate": 1563488454
    }
}
```

8. これで、デバイスはごみ処理会社の AWS IoT 1-Click アカウントに関連付けられていますが、プレイスメントには関連付けられていません。次のコマンドを実行してこれを確認します。

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

デバイスをプレイスメントに関連付けるには、次のコマンドを実行します。

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

前のコマンドを確認するには、get-devices-in-placement をもう一度実行します。

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

```
}
```

詳細については、次のように describe-device コマンドを実行します (iot1click-projects から iot1click-devices への切り替えに注意してください)。

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

現在デバイスが 1 つしかないため、次のコマンドでも同様の結果が得られます。

```
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. デバイスが正常に機能していることを確認するには、次のコマンドを実行します。ISO 8061 形式のタイムスタンプを適切に調整します。

```
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\":
 \"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",
 \"remainingLife\": 99.85000000000001, \"testMode\": false}"
        }
    ]
}
```

ここでは、シングルクリックイベント (\"clickType\": \"SINGLE\") が 2019-07-18T23:47:55.015Z に発生したことが分かります。次にデバイスをダブルクリックし (連続してすばやく 2 回ボタンを押す)、コマンドを再度実行します。次のようなダブルクリックイベント (\"clickType\": \"DOUBLE\") に注目してください。

```
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\":
 \"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",
 \"remainingLife\": 99.85000000000001, \"testMode\": false}"
        },
        {
            "Device": {
                "Attributes": {},
                "DeviceId": "G030PM0123456789",
                "Type": "button"
            },
            "StdEvent": "{\"clickType\": \"DOUBLE\",
 \"reportedTime\": \"2019-07-19T00:14:41.353Z\", \"certificateId\":
 \"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",
 \"remainingLife\": 99.8, \"testMode\": false}"
        }
    ]
}
```

10. 各デバイスタイプには、呼び出し可能なデバイスメソッドのセットがあります。デバイスタイプで使用可能なメソッドを一覧表示するには、次のように get-device-methods コマンドを実行します。

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

使用可能なメソッドの 1 つを呼び出すには、次に示すように invoke-device-method コマンドを使用します。

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

invoke-device-method.json には以下が含まれています。

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

> **Note**
>
> get メソッド (getDeviceHealthParameters など) はパラメータを使いません。したがって、JSONファイル内の "DeviceMethodParameters": "" 行は使用できません。(使用すると以下のエラーが発生します: An error occurred (InvalidRequestException) when calling the InvokeDeviceMethod operation: A request parameter was invalid.)

11. aws iot1click-devices list-devices --device-type button を実行すると、Enabled のデフォルト値が false であることがわかります。次のコマンドで、このキーを true に設定します。

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

これを false に戻すには、もう一度 --no-enabled 引数を使用して、前のコマンドを実行します。

12. お客様情報が変更された場合、次に示すように、デバイスのプレイスメント情報を更新できます (iot1click-devices から iot1click-projects への切り替えに注意してください)。次のコマンドを実行して、customer217 の現在の情報を表示します (attributes を参照)。

```
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,
        "updatedDate": 1563488454
    }
}
```

次に、次のコマンドを実行して、お客様の電話および口ケーションの属性を更新します。

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

update-placement.json には以下が含まれています。

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

この更新を確認するには、次に示すように、describe-placement をもう一度実行します。

```
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,
        "updatedDate": 1563572842
    }
}
```

13. プロジェクト情報を更新するには、update-project コマンドを使用します。通常、プロジェクトには複数のお客様のプレイスメントが含まれます。既存の SeattleDumpster プロジェクト情報は次のとおりです。

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

「シアトル地域のすべてのごみ箱」を「シアトル地域のすべてのごみ箱 (廃棄物)」に変更するにはウェスター、リサイクル、ごみ)」で、次のコマンドを実行します。

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

"description" キーの値はすべての SeattleDumpsters プレイスメントで更新されたことがわかります。

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

14. 次のように、タグを使用して、メタ情報をプロジェクトリソース (iot1click-projects) およびプレイスメントリソース (iot1click-devices) に適用できます。

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

projects-tag-resource.json には以下が含まれています。

```
{
    "resourceArn": "arn:aws:iot1click:us-west-2:012345678901:projects/
SeattleDumpsters",
    "tags": {
        "Account": "45215",
```

```
        "Manager": "Tom Jones"
    }
}
```

プロジェクトリソースのタグを一覧表示するには、以下を実行します。

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

コンテキスト内でプロジェクトタグを表示するには、以下を実行します。

```
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"
        }
```

```
    }
}
```

デバイスの Amazon リソースネーム (ARN) を検出するには、以下を実行します。

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

前述のデバイスにタグを追加するには、以下を実行します。

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

devices-tag-resources.json には、次の内容が含まれています (ResourceArn と Tags では大文字と小文字を区別します)。

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

```
}
```

デバイスリソースのタグを一覧表示するには、以下を実行します。

```
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"
    }
}
```

コンテキストでデバイスタグを表示するには、list-devices を実行します。

```
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. この時点で、AWS Lambda 関数のトリガーや Amazon SNS メッセージの送信など、アクションをデバイスのボタンを押す操作に関連付けることができます。AWS IoT 1-Click コンソール (AWS IoT 1-Click プログラミングモデルもオプションです) 。適切なアクションがデバイスに関

連付けられたら、手順 1 と 2 で説明したのと同じ手順で、デバイスをお客様の口ケーションに移動して Wi-Fi ネットワークに接続できます。

AWS IoT 1-Click デバイスのティアダウン

次の手順では、前述の手順を元に戻す方法について説明します。

1. プロジェクトリソースのタグを削除するには、次のコマンドを実行します。

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

これにより、次に示すように、プロジェクトの Manager タグが削除されます。

```
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. デバイスリソースのタグを削除するには、次のコマンドを実行します。

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

次に示すように、これによりデバイスのタグが削除されます (空のリスト "Tags": {} に注目してください)。

```
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. プレイスメントからデバイスの関連付けを削除するには、次のコマンドを実行します。

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

次に示すとおり、プレイスメント customer217 にはデバイスが関連付けられていません。

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

4. プロジェクトからプレイスメントを削除するには、次のコマンドを実行します。

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

以下で示すように、プロジェクト SeattleDumpsters にはプレイスメントがありません。これは、customer217 がSeattleDumpsters 内で唯一のプレイスメントであったためです。

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

5. プロジェクトを削除するには、次のコマンドを実行します。

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

以下で示すように、すべてのプロジェクトが削除されました。SeattleDumpstersは、AWS IoT 1-Click アカウントに関連付けられた唯一のプロジェクトでした。

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

たとえば、友人に AWS アカウントを使用してデバイスを試してもらいたい場合は、次のように、まず AWS IoT 1-Click アカウントからそのデバイスの要求を解除する必要があります。

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

これで、デバイスはどの AWS IoT 1-Click アカウントでも使用できるようになりました。

AWS IoT 1-Click 付録

このセクションでは、次に示すような追加の AWS IoT 1-Click 情報について説明します。

AWS IoT 1-Click 対応デバイス

製品	デバイスタイプ	デバイス ID プレフィックス	デバイスを要求するには	購入リンク	デバイスリージョン†
AWS IoT (米国、EU、日本) のIoT ボタン	Button (ボタン)	P5SJVQ (デバイス ID の最初の 6 桁)	AWS IoT 1-Click モバイルアプリで、デバイス ID を入力して Wi-Fi を設定し、デバイスを要求します。	Seeed Studio Bazaar	米国西部 (オレゴン)
Sercomm IoT ボタン (米国のみ)	Button (ボタン)	7VT4EQ (デバイス ID の最初の 6 桁)	AWS IoT 1-Click モバイルアプリケーションまたは AWS IoT 1-Click コンソールで、デバイスの購入時に取得したクレームコードを入力します。AWS IoT 1-Click モバイルアプリでデバイス ID を入力して、	セルコムモバイルIoT製品オンラインストア	米国西部 (オレゴン)

製品	デバイスタイプ	デバイス ID プレフィックス	デバイスを要求するには	購入リンク	デバイスリージョン†
			デバイスを請求することもできます。		
SORACOM LTE-M ボタン (日本のみ)	Button (ボタン)	7MF6JK (デバイス ID の最初の 6 桁)	AWS IoT 1-Click モバイルアプリで、デバイスを要求するデバイス ID を入力します。	SORACOM	米国西部 (オレゴン)
AWS IoT Enterprise Button (米国、EU、日本)	Button (ボタン)	G030PM (デバイス ID の最初の 6 桁)	AWS IoT 1-Click モバイルアプリで、デバイス ID を入力して、Wi-Fi を設定し、デバイスを要求します。	終了	米国西部 (オレゴン)

製品	デバイスタイプ	デバイス ID プレフィックス	デバイスを要求するには	購入リンク	デバイスリージョン†
AT&T LTE-M ボタン (米国のみ)	Button (ボタン)	B9GHXT (デバイス ID の最初の 6 桁)	AWS IoT 1-Click モバイルアプリケーションまたは AWS IoT 1-Click コンソールで、デバイスの購入時に取得したクレームコードを入力します。AWS IoT 1-Click モバイルアプリでデバイス ID を入力して、デバイスを請求することもできます。	終了	米国西部 (オレゴン)

†デバイスリージョンの詳細については、」AWS IoT 1-Click デバイス。

> **Note**
>
> AWS IoT 1-Click は、デバイスのシリアル番号 (DSN) が G030JF、G030MD および G030PT で始まる AWS IoT ボタンをサポートしていません。これらのボタンを (AWS IoT 1-Click を使用しないで) AWS IoT クラウドに接続する方法については、クラウドでプログラム可能なダッシュボタン。

AWS IoT 1-Click サービスの制限

- 配置テンプレートごとに最大 5 つのデバイステンプレートを使用できます。これは、配置ごとに 5 つのデバイスに対応します。
- AWS IoT 1-Click プロジェクトは、あたり最大 512 です。AWS リージョンアカウントあたり
- AWS IoT 1-Click リソースごとに最大 50 個のタグがあります。タグは、リソースの管理に使用できるキーと値のペア (メタデータ) です。詳細については、「AWS タグ付け戦略」を参照してください。

開発者ガイドのドキュメント履歴

次の表は、今回リリースされた AWS IoT 1-Click のドキュメントをまとめたものです。

- API バージョン: 最新
- ドキュメント最新更新日: 2018 年 10 月 22 日

変更	説明	日付
リリース	ドキュメントの初回リリース。	2018 年 5 月 14 日
編集	編集上の改訂。	2018 年 5 月 31 日
編集	サポートされているデバイステーブルを更新しました。	2018 年 10 月 22 日

AWS の用語集

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。