



开发人员指南

AWS IoT 1-Click



AWS IoT 1-Click: 开发人员指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

AWS IoT 1-Click 是什么？	1
AWS IoT 1-Click 组件	1
AWS IoT 1-Click 的工作原理	3
AWS IoT 1-Click 设备	3
声明设备	4
项目、模板和置放	4
AWS IoT 1-Click 控制台入门	7
声明设备	7
创建项目	7
示例：会议室满意度项目	8
AWS IoT 1-Click 移动应用程序	11
AWS IoT 1-Click 编程模型	12
AWS IoT 1-Click 回调事件	13
AWS IoT 1-Click 事件	13
AWS IoT 1-Click Health 事件	14
设备方法	15
使用 CloudWatch 指标进行监控	16
使用 AWS CloudTrail 记录 AWS IoT 1-Click API 调用日志	17
CloudTrail 中的 AWS IoT 1-Click 信息	17
示例：AWS IoT 1-Click 日志文件条目	18
AWS CloudFormation 集成	21
AWS IoT 1-Click 的身份验证和访问控制	22
AWS IoT 1-Click 资源和操作	22
为 AWS IoT 1-Click 使用基于身份的策略 (IAM 策略)	22
适用于 AWS IoT 1-Click 的 AWS 托管 (预定义) 策略	23
为 AWS IoT 1-Click 资源添加标签	27
有关标签的基本知识	27
标签限制	28
AWS IoT Enterprise Button 用户指南	29
AWS IoT 1-Click 与 AWS CLI 结合使用	31
AWS IoT 1-Click 附录	49
AWS IoT 1-Click 支持设备	49
AWS IoT 1-Click 服务限制	50
文档历史记录	52

AWS 词汇表	53
.....	liv

AWS IoT 1-Click 是什么？

AWS IoT 1-Click 使企业客户能够轻松将简单 IoT 设备并入其工作流，而不必制造设备、编写固件或为它们配置安全连接。我们的制造合作伙伴可创建一出箱就可以安全连接到 AWS IoT 的设备。这些设备可以触发 [AWS Lambda](#) 函数，这些函数是用像 Java、Python 和 C# 这样的语言编写的。Lambda 函数可以自行实施业务逻辑或在 AWS 云或本地触发操作。

AWS IoT 1-Click 旨在通过尽可能多地抽象化与设备硬件和固件相关的细节来为客户精简物联网。这使您能够查看作为软件组件托管在 AWS 云中的 AWS IoT 1-Click 设备。与任何其他软件组件一样，这些设备符合明确定义的接口。AWS IoT 1-Click 的接口定义为每种设备类型。您可使用这些接口构建应用程序和作为您的应用程序的基础。

利用 AWS IoT 1-Click，您可以按功能、位置或其他标准对设备进行分组。此逻辑设备组称为“project”，请参阅 [AWS IoT 1-Click](#)。对于所需操作，您可以使用项目将设备组与 Lambda 函数关联。

项目所含模板指定所用设备的类型、调用的 Lambda 函数和要为这些设备定义的可选属性 (如位置或功能上下文数据)。

创建项目并定义模板后，您可以在项目内添加位置 - 每个位置都采用模板并按设备的序列号和对特定位置的具体位置或功能有意义的属性值 (键/值对) 指定实际设备。

AWS IoT 1-Click 组件

Claim

是指使用 AWS IoT 1-Click 控制台、AWS IoT 一键下单移动应用程序或 AWS IoT 1-Click API 将 AWS IoT 设备与 AWS 账户关联的过程。

声明码

一个值，用于一次声明多个 AT&T LTE-M 按钮 (即批量)。您还可以使用设备 ID 声明设备。请参阅设备 ID 条目。

设备

物理设备，如 AWS IoT Enterprise Button 或 AT&T LTE-M 按钮。

设备属性

与特定设备关联的默认或自定义数据 (键/值对形式)。默认属性派生自放置。请参阅放置条目。

设备 ID

所有设备都具有设备 ID (如设备序列号 (DSN))。设备 ID 可用于在 AWS IoT 1-Click 中注册 AWS IoT 设备。声明码与设备 ID 不同。请参阅声明码条目。

Placement

一组表示设备的一个或多个模板 (例如, 包含两个模板化按钮的房间)。要填充放置, 请使用 AWS IoT 1-Click 控制台或 AWS IoT 1-Click 移动应用程序选择模板化设备。

放置名称

放置的名称, 通常包括地理位置或对象 ID (例如 Room 217、North Dumpster 或 Container 314)。

项目

由零个或多个放置组成的命名组 (包含模板化设备)。

项目名称

放置组的描述性名称 (如“会议室满意度”或“章程容器选取”)。

模板

用于提供一组设备的默认行为和默认属性。物理设备使用特定模板继承该模板的属性: 其 Lambda 函数和默认设备属性。模板定义放置中一类设备的行为和默认属性。一个项目可以有多个模板。

取消声明

将 AWS IoT 1-Click 设备与 AWS 账户取消关联的过程。例如, 想要借 AWS IoT 1-Click 设备的用户应首先取消设备与 AWS 账户的关联, 以便新用户可以将设备与自己的 AWS 账户关联。

AWS IoT 1-Click 的工作原理

AWS IoT 1-Click 工作流程：

1. 从一组受支持的设备中进行选择。
2. 将 AWS Lambda 函数与设备关联以触发操作。您可以使用自己的 Lambda 函数之一，也可以使用服务提供的预定义函数之一。
3. 物理部署您的设备，并使用 AWS IoT 1-Click 控制台、AWS IoT 1-Click 移动应用程序或 AWS IoT 1-Click API 启用设备。
4. 通过使用现成的 AWS IoT 1-Click 报告或构建自己的报告，获取有关设备状态和使用情况的信息。

AWS IoT 1-Click 设备

AWS IoT 1-Click 支持的设备：

- 现成可用。客户无需设计或制造它们。
- 通过使用[声明功能](#)。
- 在制造商处预置了证书并且被配置为安全地连接到 AWS IoT。您无需花时间安装 AWS IoT 1-Click 设备的证书。
- 每个 AWS IoT 1-Click 设备 type 以 AWS IoT 1-Click 定义的标准格式发出事件。例如，button 类型具有相同的事件格式，而无论是哪个制造商制造的。
- 具有设备类型和产品类型。设备类型指示设备所发出事件的格式及设备支持的设备方法。有关更多信息，请参阅 [AWS IoT 1-Click 编程模型](#)。产品类型提供了制造商和品牌详细信息。例如，如果设备类型是 button，则产品类型可能是 AT&T LTE-M 按钮。

Important

AWS IoT 1-Click 支持的设备在工厂配置为连接到特定的 [AWS 区域](#)。这些区域称为设备区域。将设备与设备区域确保设备在不进行额外用户输入的情况下安全地连接到 AWS IoT。因此，无法更改设备区域。

AWS IoT 1-Click 设备发出的事件始终通过预先配置的设备区域进行路由，因此您可在同一 AWS 区域中访问与设备相关的 Amazon CloudWatch 日志和 AWS CloudTrail 指标。设备区域也是对启用的设备计费的区域。

放置、模板和项目数据存储在与您账户关联的 AWS 区域中。此区域可能与设备区域不同。

有关支持 AWS IoT 1-Click 的设备的信息，包括如何购买和[申请](#)他们，请参阅[AWS IoT 1-Click 附录](#)。

声明设备

AWS IoT 1-Click 设备出厂时，它们并未与 AWS 客户账户关联。客户必须完成声明过程才能使用账户中的设备。有两种设备声明方式：

- 使用折扣码：如果您收到一个折扣码（格式 C-XXXXX），则可以将它输入到 AWS IoT 1-Click 控制台或 AWS IoT 1-Click 移动应用程序，以声明与单个订单相关的设备。并非所有设备（包括 AWS IoT Enterprise Button）都可以使用声明码进行声明。
- 使用设备 ID：您可以使用设备 ID（设备序列号，也称为 DSN）通过 AWS IoT 1-Click 控制台或 AWS IoT 1-Click 移动应用程序声明设备。所有 AWS IoT 1-Click 设备都可以使用设备 ID 进行声明。

有关如何声明设备的更多信息，请参阅[AWS IoT 1-Click 附录](#)和[声明设备](#)。

项目、模板和置放

设备可以按功能、位置或任何其他条件组织。此逻辑设备组称为项目。您可以使用项目将设备组与 Lambda 函数关联。

项目包含一些模板，这些模板指定了所用设备的类型、设备调用的 Lambda 函数和用于保存上下文数据（如位置或功能）的属性名称。

创建项目并定义模板后，您可以在项目中添加置放。置放遵循模板，并通过序列号和属性值指定设备，这些值与该置放的特定位置或功能相关。

以下示例说明项目和置放的使用：

示例 1：

在 SalesPersonNotification 项目中，10 位客户收到一个可以按下联系销售人员的按钮。置放共有 10 个，每个客户一个。每个置放都有 CustomerName（例如，Mr. Jones）、SalesPersonPhoneNumber（例如，1-555-555-1234）和按钮序列号（例如，G030PM12345678）的值。设备模板 NotificationButton 包含在置放中。为每个置放定义 CustomerName 和 SalesPersonPhoneNumber 属性。当客户单击该按钮时，AWS IoT 1-Click 将调用 AWS IoT 1-ClickSendSMSLambda 与 CustomerName 和 SalesPersonPhoneNumber 与该按钮关联的值。根据这些值发送 SMS。

- 放置模板：
 - 由于每位客户都将获得一个通知销售人员的按钮，因此将创建一个名为 NotificationButton 的设备模板。
 - 设备模板 (包含在放置中) 指定 NotificationButton，当单击时，将调用 SendSMSLambdaLambda 函数。
 - 将为每个放置定义名为 CustomerName 和 SalesPersonPhoneNumber 的属性。
- 放置：将创建 10 个放置，每个客户一个放置。每个放置都有 CustomerName (例如，Mr. Jones)、SalesPersonPhoneNumber (例如，1-555-555-1234) 和按钮序列号 (例如，G030PM12345678) 的特定值。
- 运算：当客户单击其按钮时，AWS IoT 1-Click 将调用 AWS IoT 1-ClickSendSMSLambda与CustomerName和SalesPersonPhoneNumber值-并将基于这两个值发送 SMS。

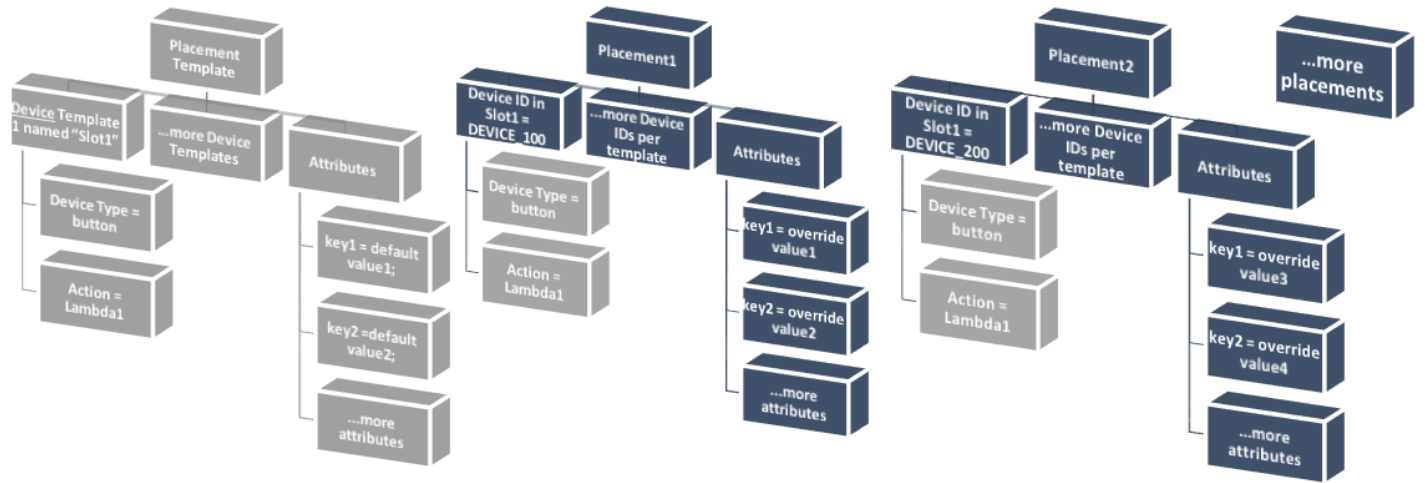
示例 2：

在 MeetingRoomFeedback 项目中，通过跟踪 50 个会议室中各个会议室按下的“赞成”和“反对”按钮来跟踪用户满意度。系统有两个设备模板 ThumbsUp 和 ThumbsDown。当单击“赞成”按钮时，调用 PostiveFeedbackLambda 函数。当单击“反对”按钮时，将调用 NegativeFeedbackLambda。AMeetingRoomNumber属性被定义为包含每个放置的会议室号。创建 50 个设备放置，每个会议室一个。每个放置都包含设置为房间号 (例如，1001) 的 MeetingRoomNumber 键和由按钮的唯一序列号标识的两个按钮 (例如，G030PM12345678 和 G030PM23456789)。当单击会议室中的按钮时，AWS IoT 1-Click 将调用PositiveFeedbackLambda函数或NegativeFeedbackLambda函数MeetingRoomNumber值。然后可以对反馈进行处理和制表。

- 项目名称：MeetingRoomFeedback
- 放置模板：
 - 由于每个房间都将获得两个按钮，因此将创建两个设备模板，名称分别 ThumbsUp 和 ThumbsDown。
 - 设备模板指定 ThumbsUp 按钮在被单击后将调用 PostiveFeedbackLambda，ThumbsDown 按钮在被单击后将调用 NegativeFeedbackLambda。
 - 名为 MeetingRoomNumber 的属性被定义为包含每个放置的房间号。
- 放置：将创建 50 个设备放置，每个房间一个放置。每个放置都包含设置为特定房间号对 (例如，1001) 的 MeetingRoomNumber 键和由按钮的唯一序列号标识的两个按钮 (例如，G030PM12345678 和 G030PM23456789)。

- 运算：当单击会议室中的按钮时，AWS IoT 1-Click 将调用PositiveFeedbackLambda函数或NegativeFeedbackLambda函数MeetingRoomNumber值 — 并且可以对反馈进行处理和制表。

下图显示了这些概念：



有关更多信息，请参阅 [AWS IoT 1-Click 控制台入门](#)。

AWS IoT 1-Click 控制台入门

以下主题介绍如何执行常用 AWS IoT 1-Click 操作。

主题

- [声明设备](#)
- [创建项目](#)

声明设备

以下过程为您演示如何声明一台或多台支持 AWS IoT 1-Click 的设备。

1. 登录您的 AWS 账户。如果您没有 AWS 账户，请打开<https://aws.amazon.com/>中，选择创建 AWS 账户，然后按照联机说明操作。
2. 从 AWS 管理控制台中，搜索“1-Click”，然后选择 AWS IoT 1-Click。
3. 如果您使用的是一个或多个 AWS IoT Enterprise Button，请安装适用于 iOS 或 Android 的 AWS IoT 1-Click 移动应用，然后将按钮连接到您本地的 Wi-Fi 网络。AWS IoT 1-Click 移动应用程序可从注册页面上的第 AWS IoT 1-Click 控制台。LTE-M 按钮不需要此步骤，因为它使用蜂窝网络。
4. 选择注册，然后选择声明设备。
5. 输入一个或多个[设备 ID](#)（例如设备序列号）或[声明码](#)，用逗号分隔，然后选择申请。如果申请按钮不可用，请仔细检查您输入的所有值。
6. 按设备上的按钮，然后选择Done。此时应显示所有已知设备的列表。

创建项目

以下过程为您演示如何为支持 AWS IoT 1-Click 的设备创建 AWS IoT 1-Click 项目。

1. 登录到您的 AWS 账户，然后打开 AWS IoT 1-Click 控制台。
2. 选择注册，然后选择创建项目。
3. 键入项目的名称和可选描述，然后选择下一步。
4. 要为放置定义一个或多个模板，请在对设备模板进行编程中，选择启动。
5. 要为任何按钮设备定义模板，请选择所有按钮类型。

- 对于 Device template name (设备模板名称)，请为您的模板输入描述性名称。在 Action (操作) 下，选择 Send SMS (发送短信) 或 Send email (发送电子邮件)。可以将使用 Lambda 函数的自定义操作选项并选择您自己的 Lambda 函数之一。根据您的选择，输入电话号码、电子邮件地址或 Lambda 函数名称。有关创建 Lambda 函数的更多信息，请参阅[AWS Lambda 开发人员指南](#)。
- UNDER添加另一个设备模板 (如果每个位置需要多台设备)选择Add。
- 输入属性键/值对。如有必要，可以输入其他键/值对。
- 选择 Create project (创建项目)。

下一节,[示例：会议室满意度项目](#)提供了如何使用 AWS IoT 1-Click 控制台创建项目的实际示例。

示例：会议室满意度项目

以下示例可帮助您理解 AWS IoT 1-Click 概念。

- 创建一个跟踪 50 个会议室 (和关联的 AV 设备) 的满意度的项目并将其命名为 MeetingRoomSat。
- 每个会议室都将收到两台设备 (按钮)，一台物理标记为“满意”，另一台物理标记为“不满意”。由于每个房间有两个按钮，因此将创建两个模板，一个名为 Satisfied，另一个名为 Unsatisfied。
- 这些区域有：Satisfied模板配置为调用名为的 Lambda 函数SatLambda。
- 这些区域有：Unsatisfied模板配置为调用名为的 Lambda 函数UnsatLambda。
- 对于这两个模板，将创建一个名为 MeetingRoomNum (键) 的属性 (键/值对)，其值为 TBD (TBD 值将在这两个按钮被物理放置在一个房间内时更改为房间号)。
- 将创建 50 个放置，每个房间一个放置。每个放置都有两个模板 (即 Satisfied 和 Unsatisfied) 与之关联。
- 两个按钮都被物理标记并放置在房间内。然后，通过使用 AWS IoT 1-Click 移动应用程序或 AWS IoT 1-Click 控制台以及按钮的序列号，标记为“满意”和“不满意”的按钮将与 50 个放置之一关联。此过程将一直持续到部署完所有剩余放置。
- 单击会议室中的房间按钮时，AWS IoT 1-Click 将调用SatLambda或者UnsatLambda函数MeetingRoomNum值 — 反馈可在云中处理和存储。
- 之后，可向该项目添加另一个模板，使得 50 个现有放置现在包含新按钮的槽，从而指示每间浴室需要更多毛巾或其他洗浴用品。

以下提供的示例演示使用 AWS IoT 1-Click 控制台创建项目来监控办公大楼 (作为一组办公大楼的一部分) 中的会议室满意度。

为了监控会议室 (包括其音频/视频设备) 的满意度, 在每间会议室放置两个 AWS IoT Enterprise Button, 一个标记为“满意”, 另一个标记为“不满意”。这是一个试点项目, 其结果可用于提高园区内其他建筑的会议室客户满意度。

在会议结束时, 鼓励参与者按“满意”或“不满意”按钮来记录其对会议室及其设备的整体满意度。之后, 此数据用于确定存在 A/V 设备无功能或其他问题的会议室。

AWS IoT 1-Click 控制台可用于设置此项目:

1. 从 AWS IoT 1-Click 控制台中, 选择创建项目。
2. 对于项目名称, 键入 **MeetingRoomSatisfaction**。对于项目说明, 键入 **Project used to track customer meeting room satisfaction, including A/V equipment**。选择下一步。
3. UNDER 对设备模板进行编程选择启动, 然后选择所有按钮类型。
4. 对于 Device template name (设备模板名称), 键入 **Satisfied**。这是用于标记为“满意”的所有按钮的模板。对于 Action (操作), 选择 Send email (发送电子邮件)。

Note

如果会议室满意度试点成功, 则可以在 Action (操作) 下选择 Custom action using a Lambda function (使用 Lambda 函数自定义操作)。此自定义 Lambda 函数可以发送电子邮件或将“满意”按钮数据存储在 Amazon DynamoDB 表格中, 以供日后分析。有关创建 Lambda 函数的信息, 请参阅[AWS Lambda 开发人员指南](#)。

5. UNDER 添加另一个设备模板 (如果每个位置需要多台设备) 选择 Add, 然后选择所有按钮类型。对于 Device template name (设备模板名称), 键入 **Unsatisfied**。这是用于标记为“不满意”的所有按钮的模板。对于 Action (操作), 选择 Send email (发送电子邮件)。
6. 对于 Required email default value (必需电子邮件默认值), 键入电子邮件地址。在 Required subject default value (必需的主题默认值) 中, 键入 **Meeting Room Feedback**。对于 Required body default value (必需的正文默认值), 键入 **Either positive or negative meeting room feedback has been provided.**
7. 对于 Attribute key (属性键), 键入 **Building**。对于 Default value (默认值), 键入 **Headquarters**。会议室满意度试验将在公司总部的建筑中进行。如果试验成功, 则会将其部署到公司的其他建筑。因此, 必须了解提供的是哪栋建筑中会议室设备的信息。
8. 在第二个键/值对行中, 对于 Attribute key (属性键), 键入 **Room**。对于 Default value (默认值), 键入 **TBD**。这些区域有: **TBD** 值将更改为会议室编号 (使用 AWS IoT 1-Click 移动应用或 AWS IoT 1-Click 控制台)。

9. 选择 Create project (创建项目)。

使用 AWS IoT 1-Click 移动应用程序，如果在会议室放置“满意”按钮，则满意模板将与其关联，**TBD**值将替换为会议室编号。在会议室中放置“不满意”按钮时，情况也是如此。

AWS IoT 1-Click 移动应用程序

AWS IoT 1-Click 移动应用程序允许您：

- 使用与 AWS IoT 1-Click 控制台相似的用户界面在现场方便地配置和监控 AWS IoT 1-Click 设备。
- 为连接了 Wi-Fi 的 AWS IoT 1-Click 设备（如 AWS IoT Enterprise Button）配置 Wi-Fi 凭证。

AWS IoT 1-Click 移动应用程序适用于 iPhone 和 Android 移动设备。要下载应用程序，请转至[应用商店](#)或者[Google Play](#)，并搜索 AWS IoT 1-Click。

AWS IoT 1-Click 编程模型

为了使用 AWS IoT 1-Click 设备构建应用程序，编程人员将使用[AWS IoT 1-Click 设备 API](#)和[AWS IoT 1-Click 项目 API](#)。设备 API 与 AWS IoT 1-Click 设备组件交互并处理来自设备的事件。这些事件包括启用和禁用设备以及定义事件格式及其触发的操作 (Lambda 函数)。设备开发工具包与驻留在制造商注册设备的区域中的 AWS 组件紧密结合。这就是为什么[AWS 设备区域](#)可能不同于客户使用设备的区域。项目 API 与 AWS IoT 1-Click 项目服务交互，用于总体上管理 AWS IoT 1-Click 设备，这使得可以：

- 将设备分组到项目中。
- 创建用于为项目中的所有设备设置操作的模板。
- 定义存储与项目相关的上下文数据的属性。

通过使用设备开发工具包，您可以使用 AWS IoT 1-Click 编程模型为单个设备编程。在本例中，您将使用 AWS IoT 1-Click 设备类型。该开发工具包定义了标准事件格式和为该类型的所有设备形成编程接口的方法的列表。要调用与指定设备类型相关的方法，程序员可以使用 [InvokeDeviceMethod API](#) 并将设备方法指定为参数。

例如，设备类型为“按钮”的所有 AWS IoT 1-Click 设备都发出与单击关联的事件，并且具有设置在单击设备时调用的回调函数的方法。有关按钮界面的信息，请参阅[按设备类型划分的接口](#)。以下是设置此回调函数的代码：

```
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);
```


您可使用项目开发工具包为大批设备编程。利用 API，您可首先定义每个放置的外观，包括每个放置的设备模板和属性。完成该操作后，您可使用特定设备 ID 创建放置。每个放置遵循相同的模板。以下是执行此操作的示例代码：

```
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 回调事件

AWS IoT 1-Click 使您能够通过注册回调来订阅设备事件。回调示例为 AWS IoT 1-Click 客户拥有和实施的 AWS Lambda 函数。每当有可供此回调使用的事件时，就会调用此回调。有关事件及其负载的信息，请参阅 [AWS IoT 1-Click 事件](#) 和 [AWS IoT 1-Click Health 事件](#) 部分。

AWS IoT 1-Click 事件

button 类型的设备在每次被单击时将发布单击事件。您可以通过以下方式订阅此事件：

- 在设备上调用设备 SetOnClickCallback 方法。
- 正确配置关联的项目，如前面的创建项目代码示例所示。

在下面的示例中，请注意 `placementInfo` 部分仅在设备具有关联的放置时出现。有关更多信息，请参阅 [项目、模板和置放](#)。

```
{
  "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 事件

设备根据 AWS IoT 1-Click 服务计算的运行状况参数来发布运行状况事件，但您设置其相应的阈值。以下示例表示剩余 10% 生命周期的设备 `G030PMXXXXXXXXXX` 的运行状况事件的 JSON 负载（记下 `"remainingLifeLowerThan":10` 键/值对）。

```
{
  "deviceEvent": {
    "deviceHealthMonitor": {
```

```

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

```

设备方法

AWS IoT 1-Click 设备方法是特定设备类型的设备支持的 API，如下表所示。由任何设备支持的设备方法的完整列表都可以通过调用 [GetDeviceMethods](#) 检索到。

设备类型	方法名称	描述
device	getDeviceHealthParameters	获取设备的运行状况参数，如 remainingLife。
device	setDeviceHealthMonitorCallback	设置设备运行状况参数低于阈值时调用的回调。
device	getDeviceHealthMonitorCallback	获取经过配置、在运行状况参数低于阈值时调用的回调。
button	setOnClickCallback	设置单击按钮时调用的回调。
button	getOnClickCallback	获取经过配置、在单击按钮时调用的回调。

使用 Amazon CloudWatch 监控 AWS IoT 1-Click

AWS IoT 1-Click 自动代表您监控设备并通过[Amazon CloudWatch](#)。这些指标在制造商注册设备的设备区域中进行报告。有关设备区域的更多信息，请参阅[AWS IoT 1-Click 的工作方式](#)。您可以在 Amazon CloudWatch 控制面板中的点击命名空间。

Amazon CloudWatch Events 可以使用自动执行您的 AWS 服务并自动响应系统事件，例如应用程序可用性问题或资源更改。AWS 服务中的事件将实时传输到 CloudWatch 事件。您可以编写简单的规则来指示您关注的事件，并指示要在事件匹配规则时执行的自动化操作。以下操作可触发：

- 调用 AWS Lambda 函数。
- 正在调用 Amazon EC2 运行命令。
- 将事件中继到 Amazon Kinesis Data Streams。
- 激活 AWS Step Functions 状态机。
- 向 Amazon SNS 主题或 AWS SMS 队列发送通知。

AWS IoT 1-Click 跟踪并报告以下指标：

- TotalEvents 将跟踪设备发布的事件的数量。此指标可以按设备事件、项目、设备类型或产品类型查看和绘成图表。
- RemainingLife 表示设备剩余寿命的大致百分比。AWS IoT 1-Click 基于制造商对设备的评级来报告此数字。例如，如果一个按钮被设计为能承受约 2000 次单击，并且已记录 500 次单击，则 RemainingLife 值将报告为 75%。RemainingLife 指标可以按项目、设备类型或产品类型查看和绘成图表。客户可以使用 RemainingLife 指标设置在设备下降到低于特定阈值时触发的警报。然后，客户可以查询 RemainingLife，通过使用 GetDeviceHealthParameters 方法来识别具有低 RemainingLife 值。
- CallbackInvocationErrors 将跟踪设备发出事件时调用回调（Lambda 函数）失败的次数。CallbackInvocationErrors 指标可按调用的回调（设置为回调的 Lambda 函数 ARN）或按项目查看和绘成图表。客户可以为 CallbackInvocationErrors 指标，以在 AWS IoT 1-Click 无法将事件从其设备路由至其已配置 Lambda 函数时收到通知。

有关更多信息，请参阅 [Amazon CloudWatch Events 用户指南](#)。

使用 AWS CloudTrail 记录 AWS IoT 1-Click API 调用日志

AWS IoT 1-Click 与 AWS CloudTrail 集成，后者是一项提供 AWS IoT 1-Click 中用户、角色或 AWS 服务所采取操作的记录的服务。CloudTrail 将 AWS IoT 1-Click 的 API 调用记录为事件。捕获的调用包括来自 AWS IoT 1-Click 控制台的调用和对 AWS IoT 1-Click API 操作的代码调用。如果您创建跟踪，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 AWS IoT 1-Click 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台的 Event history (事件历史记录) 中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 AWS IoT 1-Click 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，包括如何对其进行配置和启用，请参阅 [AWS CloudTrail 用户指南](#)。

CloudTrail 中的 AWS IoT 1-Click 信息

在您创建 CloudTrail 账户时，即针对该账户启用了 AWS。当 AWS IoT 1-Click 中发生受支持的事件活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在事件历史记录。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅 [使用 CloudTrail 事件历史记录查看事件](#)。

要持续记录 AWS 账户中的事件（包括 AWS IoT 1-Click 的事件），请创建跟踪。通过 trail (跟踪)，CloudTrail 可将日志文件传送到 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪在 AWS 分区中记录来自所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- [创建跟踪概述](#)
- [CloudTrail 支持的服务和集成](#)
- [为 CloudTrail 配置 Amazon SNS 通知](#)
- [接收多个区域中的 CloudTrail 日志文件和从多个账户中接收 CloudTrail 日志文件](#)

AWS IoT 1-Click [设备 API](#) 支持在 CloudTrail 日志文件中将以下操作记录为事件：

- [ListDevices](#)
- [DescribeDevice](#)
- [GetDeviceMethods](#)

- [UpdateDeviceState](#)
- [InvokeDeviceMethod](#)

AWS IoT 1-Click [项目 API](#) 支持在 CloudTrail 日志文件中将以下操作记录为事件：

- [CreateProject](#)
- [UpdateProject](#)
- [DescribeProject](#)
- [ListProjects](#)
- [DeleteProject](#)
- [CreatePlacement](#)
- [UpdatePlacement](#)
- [DescribePlacement](#)
- [ListPlacements](#)
- [DeletePlacement](#)
- [AssociateDeviceWithPlacement](#)
- [DisassociateDeviceFromPlacement](#)
- [GetDevicesInPlacement](#)

每个事件或日志条目都包含有关生成请求的人员的信息。身份信息帮助您确定以下内容：

- 请求是使用根用户凭证还是 AWS Identity and Access Management (IAM) 用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

示例：AWS IoT 1-Click 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 DescribeDevice 操作。

```
{
  "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"
}
```

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 CreateProject 操作。

```
{
  "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 集成

AWS IoT 1-Click CloudFormation 与 AWS CloudFormation 集成，这是一种通用语言，用于描述和预置云环境中的所有基础设施资源（例如 Amazon EC2、Auto Scaling、Amazon SNS 等）。AWS CloudFormation 允许您使用简单的文本文件以自动化的安全方式，跨所有区域和账户，为您的应用程序需要的所有资源建模并进行预置。该文件将作为您的云环境的单一信任源。有关更多信息，请参阅 [AWS CloudFormation 用户指南](#) 以及 AWS IoT 1-Click 主题（例如 [AWS::IoT1Click::Project](#)）中的 AWS CloudFormation 用户指南。

AWS IoT 1-Click 的身份验证和访问控制

访问 AWS IoT 1-Click API 需要凭证。这些凭证必须有权访问 AWS 资源，如 AWS IoT 1-Click 项目或设备。下列各节详述如何使用 AWS Identity and Access Management (IAM) 和 AWS IoT 1-Click 来帮助保护对您的资源的访问。

每个 AWS 资源都归某个 AWS 账户所有，创建和访问资源的权限由权限策略进行管理。账户管理员可以向 IAM 身份 (即：用户、组和角色) 挂载权限策略，某些服务 (如 AWS Lambda) 也支持向资源挂载权限策略。授予权限时，管理员确定谁将获得权限、获得哪些资源的权限以及它们允许对这些资源执行的具体操作。

AWS IoT 1-Click 资源和操作

在 AWS IoT 1-Click 中，主要资源为项目和设备。在策略中，您可以使用 Amazon 资源名称 (ARN) 标识策略应用到的资源。这些资源具有关联的唯一 Amazon 资源名称 (ARN)，如下表所示。

资源类型	ARN 格式
设备	arn:aws:iot1click:region:account-id:devices/device-id
项目	arn:aws:iot1click:region:account-id:projects/project-name

AWS IoT 1-Click 实施 API 来配合 AWS IoT 1-Click 资源。这些资源在 IAM 中称为“操作”。有关可用操作的列表，请参阅本主题结尾处的表。

为 AWS IoT 1-Click 使用基于身份的策略 (IAM 策略)

本主题提供了基于身份的策略的示例，这些示例展示了账户管理员如何将权限策略附加到 IAM 身份 (即用户、组和角色)，从而授予对 AWS IoT 1-Click 资源执行操作的权限。

下面介绍权限策略示例。

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

```
{
  "Effect": "Allow",
  "Action": [
    "iot1click:CreateProject"
  ],
  "Resource": "*"
}
```

此策略具有一个语句-它授予对 AWS IoT 1-Click 操作的权限 (iot1click:CreateProject) 的资源使用应用程序的 Amazon 资源名称 (ARN)。此示例中的 ARN 指定通配符 (*), 表示为任何资源授予相应权限。

有关显示所有 AWS IoT 1-Click API 操作及其适用于的资源, 请参阅[AWS IoT 1-Click API 权限: 操作、权限和资源参考](#)。

适用于 AWS IoT 1-Click 的 AWS 托管 (预定义) 策略

Amazon Web Services 通过提供由 AWS 创建和管理的独立 IAM 策略来满足许多常用案例的要求。这些 AWS 托管策略可针对常用案例授予必要的权限, 使您免去调查所需权限的工作。有关更多信息, 请参阅 [《IAM 用户指南》](#) 中的 AWS 托管策略。

下面的 AWS 托管策略可附加到您账户中的用户, 这些托管策略特定于 AWS IoT 1-Click 并按使用案例场景进行分组:

- **AWSIoT1ClickFullAccess**: 使用 AWS 管理控制台授予对 AWS IoT 1-Click 资源的完全访问权限。授予的权限包括用于管理设备和项目的所有 AWS IoT 1-Click 操作。
- **AWSIoT1ClickReadOnlyAccess**: 使用 AWS 管理控制台授予对 AWS IoT 1-Click 资源的只读访问权限。此访问权限使用户能够列出 AWS IoT 1-Click 设备和项目以及查看项目配置。

Note

您可以通过登录 IAM 控制台 (<https://console.aws.amazon.com/iam/>) 并在该位置搜索特定策略名称。

此外, 您还可以创建您自己的自定义 IAM 策略, 以授予 AWS IoT 1-Click 操作和资源的相关权限。您可以将这些自定义策略附加到需要这些权限的 IAM 用户或组。

AWS IoT 1-Click API 权限：操作、权限和资源参考

在设置 AWS 云中的访问控制和编写可附加到 IAM 身份的权限策略 (基于身份的策略) 时，可使用下表作为参考。该表列出了每个 AWS IoT 1-Click API 操作、您可授予执行权限的对应操作以及您可授予权限的 AWS 资源。您可以在策略的 Action 字段中指定这些操作，并在策略的 Resource 字段中指定资源值。

您可以在 AWS IoT 1-Click 策略中使用 AWS 范围的条件键来表达条件。有关 AWS 范围内的键 [可用键](#) 中的 IAM 用户指南。

Note

要指定操作，请在 API 操作名称之前使用 `iot1click:` 前缀 (例如，`iot1click:ListProjects`)。

IoT 1-Click 操作	所需权限 (API 操作)	资源
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

IoT 1-Click 操作	所需权限 (API 操作)	资源
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
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

IoT 1-Click 操作	所需权限 (API 操作)	资源
AssociateDeviceWithPlacement	iot1click:AssociateDeviceWithPlacement	arn:aws:iot1click:region:account-id:projects/project-name
DissociateDeviceFromPlacement	iot1click:DissociateDeviceFromPlacement	arn:aws:iot1click:region:account-id:projects/project-name
GetDevicesInPlacement	iot1click:GetDevicesInPlacement	arn:aws:iot1click:region:account-id:projects/project-name

为 AWS IoT 1-Click 资源添加标签

为了帮助您管理 AWS IoT 1-Click 资源，您可以选择使用标签将自己的元数据分配给任何基于 ARN 的资源。本章介绍标签并说明如何创建标签。

有关标签的基本知识

标签可让您按各种标准（例如用途、所有者或环境）对 AWS IoT 1-Click 资源进行分类。这在您拥有许多同类型资源时很有用-您可以根据分配给资源的标签快速搜索和识别特定资源。每个标签都包含您定义的一个键和一个可选值。例如，您可以为特定经理或账户拥有的多个按钮定义一组标签。您可以根据添加的标签搜索和筛选资源。我们建议您针对每类资源设计一组标签，以满足您的需要。使用一组连续的标签键，管理资源时会更加轻松。有关更多信息，请参阅 [AWS 标记策略](#)。

您还可以使用标签对成本进行分类和跟踪。当您把标签应用于资源时，AWS 将以逗号分隔值 (CSV) 文件格式生成一份成本分配报告，其中包括按标签汇总的使用率和成本。您可以设置代表业务类别（例如成本中心、应用程序名称或所有者）的标签，以便整理多种服务的成本。有关使用成本分配标签的更多信息，请参阅[使用成本分配标签](#)中的[AWS Billing and Cost Management 用户指南](#)。

为便于使用，您可以使用 AWS 管理控制台中的标签编辑器，此编辑器提供了一种用于创建和管理标签的集中而统一的方法。有关更多信息，请参阅 [使用标签编辑器在 AWS 管理控制台入门](#)。

您还可以使用 AWS CLI 和 AWS IoT 1-Click 设备和项目 API 处理标签。您可以在创建 AWS IoT 1-Click 项目和设备时使用标签字段：

- [CreateProject](#) (项目 API)
- [完成声明](#) (设备 API)

您可以使用以下命令添加、修改或删除现有资源的标签：

AWS IoT 1-Click 项目 API (使用项目 ARN)	AWS IoT 1-Click 设备 API (使用设备 ARN)
TagResource	标签
ListTagsForResource	请参阅 TagResource (POST) 、 TagListTagsForResource (GET) 和 UntagResource (DELETE) 。
UntagResource	

您可以修改标签的密钥和值，还可以随时删除资源的标签。您可以将标签的值设为空的字符串，但是不能将其设为空值。如果您添加的标签的值与该实例上现有标签的值相同，新的值就会覆盖旧值。如果删除资源，则所有与资源相关的标签都将被删除。

标签限制

下面是适用于标签的基本限制：

- 每个资源的标签数上限 – 50
- 最大密钥长度 — 127 个 Unicode 字符 (采用 UTF-8 格式)
- 最大值长度 — 255 个 Unicode 字符 (采用 UTF-8 格式)
- 标签键和值区分大小写。
- 请勿在标签名称或值中使用 `aws:` 前缀，因为它专为 AWS 使用预留。您无法编辑或删除带此前缀的标签名称或值。具有此前缀的标签不计入每个资源的标签数限制。
- 如果您的标记方案针对多个服务和资源使用，请记得其他服务可能对允许使用的字符有限制。通常允许使用的字符包括：可以用 UTF-8 格式表示的字母、空格和数字以及特殊字符 `+ - = . _ : / @`

AWS IoT Enterprise Button 用户指南

AWS IoT Enterprise Button 是一个简单且易于配置的基于 Wi-Fi 的按钮。此按钮旨在使企业和开发人员能够通过使用 AWS IoT 1-Click 轻松地与现有业务工作流程和系统集成。

AWS IoT Enterprise Button 支持三种类型的点击：

- 单列排序
- Double
- 长按

要正常工作，您必须使用 AWS IoT 一键式移动应用 (iOS 或 Android) 配置按钮的 Wi-Fi 连接。在应用程序中，您可以通过登录 AWS 账户或者点击应用程序右上角的 Wi-Fi 图标跳过登录，来配置按钮的 Wi-Fi 连接。

在通过移动应用或控制台配置和声明连接后，在执行单击、双击或长按操作时，按钮应闪烁绿光。

如果您怀疑在配置该按钮后出现问题，此表可帮助您进行故障排除。

颜色	状态	建议
闪烁白光	正在连接 Wi-Fi、正在获取 IP 地址或正在连接 AWS IoT。	不适用
纯绿光	已成功连接 Wi-Fi 并向 AWS IoT 发布了消息。	不适用
闪烁蓝光	按钮处于配置模式中。	等待配置过程完成。
纯橙光	未配置 Wi-Fi。	使用 AWS IoT 1-Click 式移动应用配置 Wi-Fi。
红光：短、短、短	连接到配置的无线网络时出现错误。	检查任何网络设置是否已更改，或者该按钮是否离 Wi-Fi 路由器太远。
红光：短、短、长	从无线网络获取 IP 地址时出现错误。	检查是否存在无线网络问题。

颜色	状态	建议
红光：短、长、短	执行主机名查找时出现错误。	检查是否存在无线网络问题。
红光：短、长、长	无法连接 AWS IoT。	检查是否存在无线网络问题。如果不存在网络问题且问题仍然存在，请联系 AWS 支持中心 并提供设备序列号 (DSN)。您会在按钮的背面找到它。
红光：长、短、短	无法建立与服务器的安全连接。	使用 AWS IoT 1-Click 式 iOS 或 Android 移动应用检查您是否拥有最新的固件。
红光：长、短、长	收到了 HTTP 403 禁止访问错误。	请联系 AWS 支持中心 并提供 DSN。您会在按钮的背面找到它。
红光：按下按钮 15 秒后	按钮重置。	您可以通过按下按钮 15 秒来重置 AWS IoT Enterprise Button Wi-Fi 配置。

AWS IoT 1-Click 与 AWS CLI 结合使用

要演示 AWS Command Line Interface (AWS CLI) 的使用，请考虑一个垃圾处理公司想要使用 AWS IoT 1-Click 简化其垃圾箱取件服务的情况。

在此情况下，每个垃圾箱都与 AWS IoT Enterprise Button 配对。当垃圾箱已满时，客户只需按下关联的按钮即可请求更换该垃圾箱。

Note

所有 AWS IoT Enterprise Button 设备 ID 都以 “G030PM” 开头。

垃圾处理公司使用以下步骤准备 AWS IoT Enterprise Button 供客户使用。

准备 AWS IoT Enterprise Button 供客户使用

1. 为 AWS IoT Enterprise Button 设置 Wi-Fi 的唯一方式是使用 AWS IoT 1-Click 移动应用。要安装应用程序，请参阅[AWS IoT 1-Click 移动应用程序](#)。安装应用程序后，不要按 Login to AWS Account (登录 AWS 账户) (通常情况下)。在本练习中，我们希望演示如何使用 AWS CLI。如果您按 Login to AWS Account (登录 AWS 账户)，则会为您调用 `initiate-device-claim` 和 `finalize-device-claim` 命令，我们希望使用 CLI“手动”执行此操作，如以下步骤所示。
2. 出于 AWS CLI 演示目的，而不是按登录 AWS 账户中，选择右上角的小圆形 Wi-Fi 图标。接下来，请选择 Configure Wi-Fi (配置 Wi-Fi)。扫描或输入设备 ID，然后按照移动应用程序的其余说明进行操作。
3. 如果您没有安装 AWS CLI，请按照[安装 AWS CLI](#)。要列出可用的 AWS IoT 1-Click AWS CLI 命令，请运行以下两个命令。

```
aws iot1click-projects help
```

```
aws iot1click-devices help
```

4. 要将现在连接 Wi-Fi 的 AWS IoT Enterprise Button 与垃圾处理公司的 AWS 账户关联，请使用设备中的设备 ID 运行以下命令。

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

```
}
```

按设备上的按钮。白光间歇性闪烁后，您应该会看到一秒钟左右的纯绿色指示灯。如果没有看到，请重复以前的 Wi-Fi 连接步骤。

5. 在上一步中看到纯绿色指示灯后，请运行以下命令（使用设备的 ID 值）。

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

这些区域有："State": "CLAIMED"响应指示设备已成功注册 AWS IoT 1-Click 服务。

Note

如果设备制造商提供了以“C-”开头的声明码，您只能使用 `aws iot1click-devices claim-devices-by-claim-code` 命令利用单个声明码声明一个或多个设备，如以下示例所示。

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

在此示例中，“Total”：9表示与声明码关联的九个设备C-123EXAMPLE已由 AWS IoT 1-Click 服务成功声明。

6. 接下来，您可以通过创建名为的 JSON 文本文件来准备为垃圾处理公司创建相应的 AWS IoT 1-Click 项目 `create-project.json`。此文件包含以下内容：

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

```
    }  
  }  
}
```

placementTemplate 和 deviceTemplates 键值对是将应用于作为 SeattleDumpsters 项目一部分的所有按钮的属性。要创建此项目，请运行以下命令（假定 create-project.json 位于 [当前工作目录](#)）。

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

要查看新创建的项目，请运行以下命令。

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

有关更多详细信息，请运行 describe-project 命令，如下所示。

```
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. 在为西雅图区域创建项目后，接下来为特定的垃圾箱（对于客户 217）创建放置，如下所示。转义引号对于 Windows 是必需的。

```

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

```

要查看新创建的放置，请运行以下命令。

```

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

```

有关更多详细信息，请运行 describe-placement 命令，如下所示。

```

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. 尽管设备现在与垃圾处理公司的 AWS IoT 1-Click 账户关联，但它与放置没有关联。可以通过运行以下命令确认。

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

要将设备与放置关联，请运行以下命令。

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

要确认上一命令，请再次运行 `get-devices-in-placement`。

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

有关更多详细信息，请按如下方式运行 `describe-device` 命令（注意从 `iot1click-projects` 切换到 `iot1click-devices`）。

```
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": {}
  }
}

```

由于当前只有一个设备，因此以下命令会产生类似的结果。

```

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. 要验证设备是否正常工作，请运行以下命令。适当调整时间戳，采用 [ISO 8061 格式](#)。

```

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}"
    }
  ]
}

```

在这里，我们可以看到一个单击事件（`\"clickType\": \"SINGLE\"`）发生在 2019-07-18T23:47:55.015Z。现在双击设备（连续两次快速按下按钮），然后再次运行命令。现在请注意双击事件（`\"clickType\": \"DOUBLE\"`），如下所示。

```

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. 每种设备类型都有一组可调用的设备方法。要列出您的设备类型的可用方法，请运行 `get-device-methods` 命令，如下所示。

```

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

```

要调用其中一种可用方法，请使用 `invoke-device-method` 命令，如下所示。

```

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

```

此处，`invoke-device-method.json` 包含下列内容。

```

{
  "DeviceId": "G030PM0123456789",
  "DeviceMethod": {
    "DeviceType": "device",

```

```
    "MethodName": "getDeviceHealthParameters"
  }
}
```

Note

这些 get 方法 (如 `getDeviceHealthParameters`) 不需要参数。因此, 不能使用 JSON 文件中的 `"DeviceMethodParameters": ""` 行 (这样做会导致: `An error occurred (InvalidRequestException) when calling the InvokeDeviceMethod operation: A request parameter was invalid.`)

11. 通过运行 `aws iot1click-devices list-devices --device-type button`, 您可以看到 `Enabled` 的默认值为 `false`。以下命令将此键设置为 `true`。

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

要将其设置回 `false`, 请运行上一个命令, 再次使用 `--no-enabled` 参数。

12. 如果客户信息发生变化, 您可以按如下所示更新设备的放置信息 (注意从 `iot1click-devices` 切换到 `iot1click-projects`)。运行以下命令查看 `customer217` 的当前信息 (请参阅 `attributes`)。

```
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
  }
}
```

接下来, 运行以下命令以更新客户的电话和位置属性。

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

此处，update-placement.json 包含下列内容。

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

要查看此更新，请再次运行 describe-placement，如下所示。

```
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
  }
}
```

13. 要更新项目信息，请使用 update-project 命令。一个项目通常包含多个客户放置。以下是现有的 SeattleDumpster 项目信息。

```
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": {}
}
```

若要将“西雅图区域中的所有垃圾箱”更改为“所有垃圾箱（院子）浪费、回收和垃圾”，请运行以下命令。

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

您可以看到所有 SeattleDumpsters 放置项的 "description" 密钥值已更新。

```
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. 您可以使用标签将元信息应用于项目资源 (iot1click-projects) 和放置资源 (iot1click-devices) , 如下所示。

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

此处 , `projects-tag-resource.json` 包含下列内容。

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

要列出项目资源的标签 , 请运行以下命令。

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

要在上下文中查看项目标签 , 请运行以下命令。

```
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": {
      "Manager": "Tom Jones",
      "Account": "45215"
    }
  }
}
```

要查找设备 Amazon 资源名称 (ARN)，请运行以下命令。

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

```
}
```

要向上一个设备添加标签，请运行以下命令。

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

此处，devices-tag-resources.json 包含以下内容（注意 ResourceArn 和 Tags 所需的外壳）。

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

要列出设备资源的标签，请运行以下命令。

```
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"
  }
}
```

要在上下文中查看设备标签，请运行 list-devices。

```
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. 此时，您可以将操作与按下设备的按钮相关联，例如触发 AWS Lambda 函数或发送 Amazon SNS 消息。您可以使用 AWS IoT 1-Click 控制台轻松执行此操作（[AWS IoT 1-Click 编程模型](#)也是一个选项）。将相应的操作与设备关联后，您可以使用步骤 1 和 2 中所述的相同步骤将设备带到客户所在的位置，并将其连接到其 Wi-Fi 网络。

AWS IoT 1-Click 设备拆除

以下步骤描述了如何反转（撤消）前面的步骤。

1. 要取消标记项目资源，请运行以下命令。

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

这将删除项目的 Manager 标签，如下所示。

```
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. 要取消标记设备资源，请运行以下命令。

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

这将删除设备的标签，如下所示（注意空列表 "Tags": {}）。

```
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": {}
    }
  ]
}

```

- 要取消设备与放置的关联，请运行以下命令。

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

如下所示，放置 customer217 不再具有与其关联的设备。

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

- 要从项目中删除放置，请运行以下命令。

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

如下所示，项目 SeattleDumpsters 没有放置，因为放置 customer217 是 SeattleDumpsters 其中的唯一放置。

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

- 要删除项目，请运行以下命令。

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

正如您在以下内容中所看到的，所有项目都已删除，因为 SeattleDumpsters 是唯一与您的 AWS IoT 1-Click 账户关联的项目。

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

例如，如果您想让朋友使用其 AWS 账户试用您的设备，则必须先从 AWS IoT 1-Click 账户中取消声明您的设备，如下所示。

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

此设备现在可以与任何 AWS IoT 1-Click 账户配合使用。

AWS IoT 1-Click 附录

本节提供了如下所示的额外 AWS IoT 1-Click 信息。

AWS IoT 1-Click 支持设备

产品	设备类型	设备 ID 前缀	声明设备	购买链接	设备区域 [†]
AWS IoT (美国、欧盟和日本) 的 IoT Button	按钮	P5SJVQ (设备 ID 的前 6 位)	在 AWS IoT 1-Click 移动应用程序中，输入设备 ID 以配置 Wi-Fi 并申请设备。	Seeed Studio Bazaar	US West (Oregon)
Sercomm IoT 按钮 (仅限美国)	按钮	7VT4EQ (设备 ID 的前 6 位)	在 AWS IoT 1-Click 移动应用程序或 AWS IoT 1-Click 控制台中，输入您在购买设备时获得的优惠码。您还可以通过 在 AWS IoT 1-Click 移动应用程序中输入设备 ID 来声明设备。	赛通手机产品在线商店	US West (Oregon)
SORACOM LTE-M Button (日本)	按钮	7MF6JK (设备 ID 的前 6 位)	在 AWS IoT 1-Click 移动应用程序中，输入设备 ID 以声明设备。	SORACOM	US West (Oregon)

产品	设备类型	设备 ID 前缀	声明设备	购买链接	设备区域 [†]
AWS IoT Enterprise Button (美国、欧盟和日本)	按钮	G030PM (设备 ID 的前 6 位)	在 AWS IoT 1-Click 移动应用程序中，输入设备 ID 以配置 Wi-Fi 并声明设备。	已停产	US West (Oregon)
AT&T LTE-M 按钮 (仅限美国)	按钮	B9GHXT (设备 ID 的前 6 位)	在 AWS IoT 1-Click 移动应用程序或 AWS IoT 1-Click 控制台中，输入您在购买设备时获得的优惠码。您还可以通过 AWS IoT 1-Click 移动应用程序中输入设备 ID 来声明设备。	已停产	US West (Oregon)

[†]有关设备区域的更多信息，请参阅[AWS IoT 1-Click 设备](#)。

Note

AWS IoT 1-Click 不支持设备序列号 (DSN) 以 G030JF、G030MD 和 G030PT 开头的 AWS IoT Button。要了解如何将它们连接到 AWS IoT 云 (而不使用 AWS IoT 1-Click)，请参阅[云可编程短跑按钮](#)。

AWS IoT 1-Click 服务限制

- 每个放置模板最多可以有 5 个设备模板。这相当于每个放置 5 个设备。

- 每个 AWS IoT 1-Click 项目最多有 512 个 IoT 1-Click 项目 [AWS 区域](#) 每个账户个。
- 每个 AWS IoT 1-Click 资源最多有 50 个标签。标签是用于管理资源的键/值对 (元数据)。有关更多信息，请参阅 [AWS 标记策略](#)。

《开发人员指南》的文档历史记录

下表介绍了此版本的 AWS IoT 1-Click 的文档。

- API 版本：最新
- 最近文档更新时间：2018 年 10 月 22 日

变更	描述	日期
版本	本文档的初始版本。	2018 年 5 月 14 日
编辑	编辑改进。	2018 年 5 月 31 日
编辑	更新了支持的设备表。	2018 年 10 月 22 日

AWS 词汇表

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

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。