



Manuel du développeur

# AWS IoT 1-Click



# AWS IoT 1-Click: Manuel du développeur

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

Les marques commerciales et la présentation commerciale d'Amazon ne peuvent pas être utilisées en relation avec un produit ou un service extérieur à Amazon, d'une manière susceptible d'entraîner une confusion chez les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon sont la propriété de leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Qu'est-ce qu'AWS IoT 1-Click ? .....	1
Composants AWS IoT 1-Click .....	1
Fonctionnement d'AWS IoT 1-Click .....	4
Périphériques AWS IoT 1-Click .....	4
Demande de périphériques .....	5
Projets, modèles et placements .....	6
Commencer à utiliser la console AWS IoT 1-Click .....	9
Demande de périphériques .....	9
Création d'un projet .....	9
Exemple : Projet Satisfaction de la salle de réunion .....	10
Application mobile AWS IoT 1-Click .....	14
Modèle de programmation AWS IoT 1-Click .....	15
Événements de rappel AWS IoT 1-Click .....	16
Événements AWS IoT 1-Click .....	17
Événements de Health AWS IoT 1-Click .....	18
Méthodes de périphérique .....	18
Surveillance à l'aide des métriques CloudWatch .....	20
Journalisation des appels d'API AWS IoT 1-Click .....	22
Informations AWS IoT 1-Click .....	22
Exemple : AWS IoT 1-Click .....	24
Intégration d'AWS CloudFormation .....	27
Authentification et contrôle d'accès pour AWS IoT 1-Click .....	28
Ressources et opérations AWS IoT 1-Click .....	28
Utilisation des stratégies basées sur l'identité (stratégies IAM) pour AWS IoT 1-Click .....	29
Stratégies gérées par AWS (stratégies prédéfinies) pour AWS IoT 1-Click .....	29
Balises de vos ressources AWS IoT 1-Click .....	33
Principes de base des balises .....	33
Restrictions liées aux balises .....	34
Guide de l'utilisateur du AWS IoT Enterprise Button .....	35
Utilisation d'AWS IoT 1-Click avec l'interface de ligne de commande AWS .....	38
Annexe AWS IoT 1-Click .....	57
Périphériques AWS IoT 1-Click .....	57
Limites de service AWS IoT 1-Click .....	60
Historique du document .....	61

---

Glossaire AWS .....	62
.....	lxiii

# Qu'est-ce qu'AWS IoT 1-Click ?

AWS IoT 1-Click facilite aux clients l'incorporation de périphériques IoT simples dans leurs flux de travail sans avoir à fabriquer de périphériques, concevoir des microprogrammes ou les configurer pour une connectivité sécurisée. Nos partenaires industriels créent des périphériques qui sont prêts à se connecter à AWS IoT en toute sécurité. Ces périphériques peuvent déclencher [AWS Lambda](#) qui sont écrites dans des langages tels que Java, Python et C#. Les fonctions Lambda peuvent implémenter d'elles-mêmes une logique métier ou déclencher des actions dans le cloud AWS ou sur site.

AWS IoT 1-Click vise à simplifier l'Internet des objets pour les clients en extrayant autant de détails que possible concernant le matériel et le firmware de l'appareil. Cela permet d'afficher les périphériques AWS IoT 1-Click comme des composants logiciels hébergés dans le cloud AWS. Comme pour n'importe quel autre composant logiciel, ces périphériques sont adaptés aux interfaces bien définies. AWS IoT 1-Click possède des interfaces définies par type de périphérique. Vous pouvez utiliser ces interfaces pour construire et baser vos applications.

AWS IoT 1-Click, vous pouvez regrouper des appareils par fonction, emplacement ou autres critères. Ce groupement logique d'appareils est appelé «project » dans AWS IoT 1-Click. Vous pouvez utiliser les projets pour associer des groupes de périphériques à des fonctions Lambda pour les actions désirées.

Les projets contiennent des modèles qui spécifient les types de périphériques utilisés, les fonctions Lambda qu'ils appellent et les attributs facultatifs qui sont définis pour ces périphériques, comme les données contextuelles pour l'emplacement ou la fonction.

Une fois le projet créé et les modèles définis, vous pouvez ajouter des placements au sein du projet. Chaque placement suit le modèle et spécifie les périphériques par leurs numéros de série et leurs valeurs d'attribut (paires clé/valeur) qui sont pertinents pour l'emplacement ou la fonction de ce placement en particulier.

## Composants AWS IoT 1-Click

### Demander

Fait référence au processus d'association d'un appareil AWS IoT 1-Click à un compte AWS à l'aide de la console AWS IoT 1-Click, de l'application mobile AWS IoT 1-Click ou de l'API AWS IoT 1-Click.

## Code de demande

Valeur utilisée pour demander un certain nombre de boutons AT&T LTE-M à la fois (c'est-à-dire en bloc). Vous pouvez également utiliser des ID d'appareil pour demander des appareils. Voir l'entrée ID d'appareil.

## Device

Appareil physique, tel que AWS IoT Enterprise Button ou AT&T LTE-M Button.

## Attributs du périphérique

Données par défaut ou personnalisées associées à un appareil particulier sous la forme de paires de clé/valeur. Les attributs par défaut sont dérivés du placement. Voir l'entrée du placement.

## ID de l'appareil

Tous les périphériques disposent d'un ID, tel que le numéro de série de périphérique (DSN). Un ID de périphérique peut être utilisé pour enregistrer un appareil AWS IoT 1-Click avec AWS IoT 1-Click. Un code de demande n'est pas la même chose qu'un identifiant d'appareil. Voir l'entrée du code de demande.

## Placement

Groupe d'un ou plusieurs modèles représentant des appareils (par exemple, une salle contenant deux boutons modélisés). Pour remplir un emplacement, utilisez la console AWS IoT 1-Click ou l'application mobile AWS IoT 1-Click pour choisir des appareils modélisés.

## Nom du placement

Nom du placement, qui comprend souvent un emplacement géographique ou un ID d'objet (par exemple, Room 217, North Dumpster ou Container 314).

## Projet

Un groupe nommé de zéro placement ou plus (contenant des appareils modélisés).

## Nom du projet

Nom descriptif d'un groupe de placements (par exemple, « Satisfaction envers la salle de réunion » ou « Collecte de charte de conteneur »).

## Modèle

Sert à fournir un comportement et des attributs par défaut à un groupe de périphériques. Un périphérique physique utilise un modèle particulier pour hériter des propriétés de ce modèle (sa

fonction Lambda et ses attributs périphérique par défaut). Un modèle définit le comportement et les attributs par défaut d'une classe d'appareils dans un placement. Un projet peut avoir plusieurs modèles.

### Annuler la demande

Processus de dissociation d'un appareil AWS IoT 1-Click d'un compte AWS. Par exemple, une personne qui souhaite prêter un appareil AWS IoT 1-Click doit d'abord dissocier cet appareil du compte AWS afin que le nouvel utilisateur puisse associer l'appareil à son propre compte AWS.

# Fonctionnement d'AWS IoT 1-Click

Voici le flux de travail AWS IoT 1-Click :

1. Choisissez un ensemble de périphériques pris en charge.
2. Associez les fonctions AWS Lambda à des appareils pour déclencher des actions. Vous pouvez utiliser l'une de vos propres fonctions Lambda ou l'une des fonctions prédéfinies fournies par le service.
3. Déployez physiquement vos appareils et utilisez la console AWS IoT 1-Click, l'application mobile AWS IoT 1-Click ou l'API AWS IoT 1-Click pour les activer.
4. Obtenez des informations sur l'état et l'utilisation de l'appareil en utilisant les rapports AWS IoT 1-Click, prêts à l'emploi.

## Périphériques AWS IoT 1-Click

### Appareils AWS IoT 1-Click

- Sont prêts à l'emploi. Les clients n'ont pas besoin de les concevoir ou les fabriquer.
- Les appareils peuvent être ajoutés aux comptes AWS IoT 1-Click [Fonctions de demande](#).
- Sont provisionnés en amont avec des certificats au moment de leur fabrication, et configurés pour se connecter à AWS IoT en toute sécurité. Vous n'avez ainsi plus à perdre de temps à installer des certificats sur les périphériques AWS IoT 1-Click.
- Chaque périphérique AWS IoT 1-ClicktypeÉmet des événements dans un format standard défini par AWS IoT 1-Click. Par exemple, tous les périphériques AWS IoT 1-Clickbut tonont le même format d'événement, quel que soit le fabricant.
- Avoir un type d'appareil et un type de produit. Le type d'appareil indique le format des événements émis par l'appareil et les méthodes d'appareil qu'il prend en charge. Pour plus d'informations, consultez [Modèle de programmation AWS IoT 1-Click](#). Le type de produit fournit les détails du fabricant et de la marque. Par exemple, si le type de périphérique estbut ton, le type de produit peut être AT&T LTE-M.

### Important

Les appareils AWS IoT 1-Click [Région AWS](#). Ces régions sont appelées régions d'appareils. Cette association d'un périphérique et un périphérique Région de l'appareil est nécessaire pour s'assurer que les appareils se connectent à AWS IoT en toute sécurité et sans entrée utilisateur supplémentaire. C'est la raison pour laquelle, la région de l'appareil ne peut pas être modifiée.

Les événements émis par les appareils AWS IoT 1-Click sont toujours routés via la région de périphérique préconfigurée, afin que vous puissiez accéder aux métriques de Amazon CloudWatch Logs AWS CloudTrail relatives au périphérique, dans la même région AWS. La région de l'appareil est également visible là où les appareils activés sont facturés.

Les données de placement, de modèle et de projet sont stockées dans la région AWS associée à votre compte. Cette région peut être différente de la région de l'appareil.

Pour de plus amples informations sur les appareils AWS IoT 1-Click [Demander](#) eux, voir [Annexe AWS IoT 1-Click](#).

## Demande de périphériques

Lorsque les appareils AWS IoT 1-Click quittent l'usine, ils ne sont associés à aucun compte client AWS. Les clients doivent passer par un processus de demande pour utiliser les appareils de leur compte. Il existe deux façons de demander des appareils.

- Utilisation d'un code de réclamation : Si vous recevez un code de réclamation (au format C-xxxxxx) depuis votre point d'achat, saisissez-le dans la console AWS IoT 1-Click et AWS IoT 1-Click. Tous les appareils, y compris AWS IoT Enterprise Button, ne peuvent pas être demandés à l'aide d'un code de demande.
- Utilisation d'un ID de périphérique : Vous pouvez utiliser l'ID du périphérique (numéro de série de l'appareil, aussi appelé DSN) pour demander des périphériques via AWS IoT 1-Click. Tous les appareils AWS IoT 1-Click

Pour de plus amples informations sur la procédure de demande des appareils, veuillez consulter [Annexe AWS IoT 1-Click](#) et [Demande de périphériques](#).

## Projets, modèles et placements

Les périphériques peuvent être organisés par fonction, emplacement ou n'importe quel autre critère. Ce groupement logique d'appareils est appelé « projet ». Vous pouvez utiliser les projets pour associer des groupes de périphériques à des fonctions Lambda.

Les projets contiennent des modèles qui spécifient les types de périphériques utilisés, les fonctions Lambda qu'ils appellent et les noms d'attribut pour donner du contexte comme l'emplacement ou la fonction.

Une fois le projet créé et les modèles définis, vous pouvez ajouter des emplacements dans le projet. Les emplacements suivent le modèle et spécifient les appareils par leurs numéros de série et leurs valeurs d'attribut qui ont un sens pour l'emplacement ou la fonction spécifique de ce placement.

Les exemples suivants illustrent l'utilisation des projets et des placements :

Exemple 1 :

Dans le projet `SalesPersonNotification`, 10 clients reçoivent un bouton sur lequel ils peuvent appuyer pour contacter un vendeur. Il y a 10 placements, un pour chaque client. Chaque emplacement a des valeurs pour `CustomerName` (par exemple, M. Jones), `SalesPersonPhoneNumber` (par exemple, 1-555-555-1234) et le numéro de série du bouton (par exemple, G030PM12345678). Le modèle d'appareil, `NotificationButton`, est contenu dans le placement. Les attributs `CustomerName` et `SalesPersonPhoneNumber` sont définis pour chaque placement. Lorsqu'un client clique sur le bouton, AWS IoT 1-Click `SendSMSLambda` avec le `CustomerName` et `SalesPersonPhoneNumber` associées à ce bouton. Le SMS est envoyé en fonction de ces valeurs.

- Modèle de placement :
  - Puisque chaque client dispose d'un bouton pour notifier un vendeur, un modèle de périphérique nommé `NotificationButton` est créé.
  - Le modèle de périphérique (contenu dans le placement) spécifie que le paramètre `NotificationButton`, lorsqu'il est cliqué, appellera la méthode `SendSMSLambda`.
  - Les attributs appelés `CustomerName` et `SalesPersonPhoneNumber` sont définis pour chaque placement.

- Placements : 10 placements sont créés, un par client. Chaque emplacement a des valeurs spécifiques pour `CustomerName` (par exemple, « M. Jones »), `SalesPersonPhoneNumber` (par exemple, 1-555-555-1234) et le numéro de série du bouton (par exemple, G030PM12345678).
- Opération : Lorsqu'un client clique sur son bouton, AWS IoT 1-ClickSendSMSLambdaavec leCustomerNameandSalesPersonPhoneNumberassociées à ce bouton particulier - et un SMS est envoyé en fonction de ces valeurs.

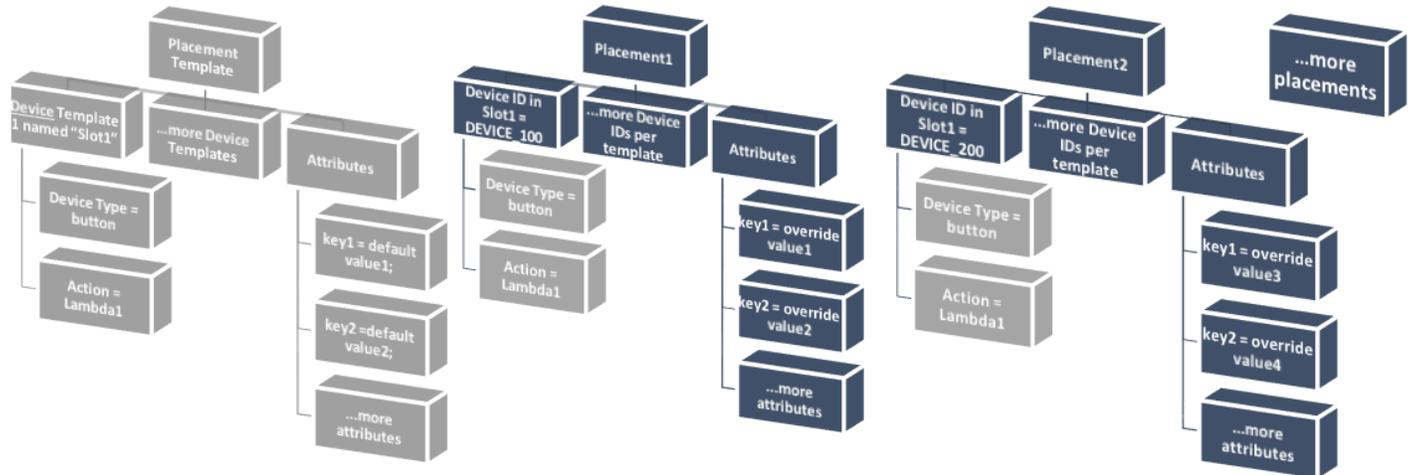
## Exemple 2 :

Dans le cadre du projet `MeetingRoomFeedback`, la satisfaction des utilisateurs est surveillée en appuyant sur les boutons Pouce vers le haut et Pouce vers le bas dans chacune des 50 salles de conférence. Il existe deux modèles d'appareils, `ThumbsUp` et `ThumbsDown`. Lorsque vous cliquez sur le bouton Pouce vers le haut, la fonction `PositiveFeedbackLambda` est appelée. Lorsque vous cliquez sur le bouton Pouce vers le bas, le `NegativeFeedbackLambda` est appelé. `MeetingRoomNumber` l'attribut est défini pour conserver le numéro de la salle pour chaque placement. 50 emplacements de périphériques sont créés, un par salle de conférence. Chaque placement comprend la clé `MeetingRoomNumber` définie sur une paire de numéros de salle particulière (par exemple, 1001) et deux boutons, identifiés par leurs numéros de série uniques (par exemple G030PM23456789 et G030PM12345678). Lorsqu'un bouton est cliqué dans une salle de réunion, AWS IoT 1-Click appelle la fonction `PositiveFeedbackLambda` ou `NegativeFeedbackLambda` Avec la fonction `MeetingRoomNumberValeur`. Les commentaires peuvent ensuite être traités et compilés.

- Nom du projet : `MeetingRoomFeedback`
- Modèle de placement :
  - Étant donné que chaque salle comprend deux boutons, deux modèles de périphérique sont créés, respectivement nommés `ThumbsUp` et `ThumbsDown`.
  - Les modèles de périphérique spécifient que les boutons `ThumbsUp` appelleront `PositiveFeedbackLambda` à l'activation, et que les boutons `ThumbsDown` appelleront `NegativeFeedbackLambda` à l'activation.
  - Un attribut appelé `MeetingRoomNumber` est défini pour conserver le numéro de la salle pour chaque placement.
- Placements : 50 placements d'appareils sont créés, un placement par salle. Chaque placement comprend la clé `MeetingRoomNumber` définie sur une paire de numéros de salle particulière (par exemple, 1001) et deux boutons, identifiés par leurs numéros de série uniques (par exemple G030PM23456789 et G030PM12345678).

- Opération : Lorsqu'un bouton est cliqué dans une salle de réunion, AWS IoT 1-Click appelle la fonction `PositiveFeedbackLambda` ou `NegativeFeedbackLambda` avec la fonction `MeetingRoomNumber`. Les commentaires peuvent être traités et compilés.

Le diagramme suivant montre ces concepts :



Pour plus d'informations, consultez [Commencer à utiliser la console AWS IoT 1-Click](#).

# Commencer à utiliser la console AWS IoT 1-Click

Les rubriques suivantes expliquent comment effectuer des tâches AWS IoT 1-Click courantes.

## Rubriques

- [Demande de périphériques](#)
- [Création d'un projet](#)

## Demande de périphériques

La procédure suivante vous montre comment demander un ou plusieurs périphériques AWS IoT 1-Click pris en charge.

1. Ouvrez une session de votre compte AWS. Si vous ne possédez pas de compte AWS, ouvrez <https://aws.amazon.com/>, choisissez **Créer un compte AWS**, et suivez les instructions en ligne.
2. Depuis la console AWS Management Console, recherchez « 1-Click », puis choisissez **AWS IoT 1-Click**.
3. Si vous utilisez un ou plusieurs boutons AWS IoT Enterprise, installez l'application mobile AWS IoT 1-Click pour iOS ou Android et connectez les boutons à votre réseau Wi-Fi local. L'application mobile AWS IoT 1-Click est disponible à partir de la page **Intégration** de la console AWS IoT 1-Click. Cette étape n'est pas requise pour le bouton LTE-M car il utilise le réseau cellulaire.
4. Choisissez **Intégration**, puis choisissez **Appareils de demande**.
5. Saisissez une ou plusieurs options **ID de l'appareil** (tel qu'un numéro de série de l'appareil) ou **codes de demande**, séparés par des virgules, puis choisissez **Demander**. Si l'icône **Demander** n'est pas disponible, vérifiez deux fois toutes les valeurs que vous avez saisies.
6. Appuyez sur le (s) bouton (s) de votre (vos) appareil (s), puis choisissez **Terminé**. Une liste de tous les appareils connus doit s'afficher.

## Création d'un projet

La procédure suivante vous montre comment créer un projet AWS IoT 1-Click pour les périphériques AWS IoT 1-Click pris en charge.

1. Connectez-vous à votre compte AWS et ouvrez la console AWS IoT 1-Click.
2. Choisissez **Intégration**, puis choisissez **Crée un projet**.
3. Saisissez un nom et une description facultative pour le projet, puis choisissez **Suivant**.
4. Pour définir un ou plusieurs modèles pour votre placement, sous **Programmer un modèle de périphérique**, choisissez **Démarrer**.
5. Pour définir un modèle pour n'importe quel bouton d'appareil, choisissez **Tous les types de bouton de**.
6. Pour **Nom du modèle d'appareil**, entrez un nom descriptif du modèle. Sous **Action**, choisissez **Envoyer un SMS** ou **Envoyer un e-mail**. Vous pouvez utiliser la stratégie **Action personnalisée** à l'aide d'une fonction Lambda et choisissez une de vos propres fonctions Lambda. Saisissez le numéro de téléphone, l'adresse e-mail ou le nom de la fonction Lambda, selon votre choix. Pour de plus amples informations sur la création de fonctions Lambda, veuillez consulter le [Manuel du développeur AWS Lambda](#).
7. **UNDER** Ajoutez un autre modèle de périphérique (si vous avez besoin de plusieurs périphériques par placement). Choisissez **Addition**.
8. Entrez un attribut paire clé/valeur. Vous pouvez entrer des paires clé-valeur supplémentaires, si nécessaire.
9. Choisissez **Créer un projet**.

La section suivante, [Exemple : Projet Satisfaction de la salle de réunion](#), fournit un exemple réel de la façon d'utiliser la console AWS IoT 1-Click pour créer un projet.

## Exemple : Projet Satisfaction de la salle de réunion

L'exemple suivant peut vous aider à comprendre les concepts AWS IoT 1-Click.

- Un projet pour évaluer la satisfaction de 50 salles de réunion (et des équipements AV associés) est créé et nommé `MeetingRoomSat`.
- Chaque salle de réunion recevra deux périphériques (boutons), un physique marqué « Satisfait » et l'autre marqué « Non satisfait ». Dans la mesure où il existe deux boutons par espace, deux modèles sont créés, l'un appelé `Satisfied` et l'autre `Unsatisfied`.
- Le `.Satisfied` est configuré pour appeler une fonction Lambda nommée `SatLambda`.
- Le `.Unsatisfied` est configuré pour appeler une fonction Lambda nommée `UnsatLambda`.

- Pour ces deux modèles, un attribut (paire clé/valeur) nommé `MeetingRoomNum` (clé) est créé, dont la valeur est TBD (la valeur TBD sera remplacée par le numéro de salle lorsque les deux boutons seront physiquement placés dans une salle).
- 50 placements sont créés, un par salle. Chaque placement comporte les deux modèles qui y sont associés (par exemple, `Satisfied` et `Unsatisfied`).
- Deux boutons physiques sont étiquetés et placés dans une salle. Ensuite, à l'aide de l'application mobile AWS IoT 1-Click ou de la console AWS IoT 1-Click et des numéros de série des boutons, les boutons marqués « Satisfait » et « Non satisfait » sont associés à l'un des 50 placements. Ce processus continue jusqu'à ce que tous les placements restants aient été déployés.
- Lorsqu'un bouton d'une salle est pressé, AWS IoT 1-Click appelle la stratégie `SatLambda` ou `UnsatLambda` avec la fonction `MeetingRoomNum`. Les commentaires peuvent alors être traités et stockés dans le Cloud.
- Plus tard, un autre modèle peut être ajouté au projet afin que les 50 placements existants ne contiennent qu'un emplacement pour un nouveau bouton indiquant qu'un plus grand nombre de serviettes ou autres linges de bain sont nécessaires dans chaque salle de bain.

Voici un exemple d'utilisation de la console AWS IoT 1-Click pour créer un projet qui surveille la satisfaction d'une salle de réunion dans un immeuble de bureaux (dans un groupe de bâtiments de bureaux).

Pour surveiller la satisfaction des salles de réunion, y compris leur équipement audio/vidéo, deux boutons AWS IoT Enterprise, l'un intitulé « Satisfait » et l'autre « Mécontent », sont placés dans chaque salle de réunion. Il s'agit d'un projet pilote dont les résultats peuvent être utilisés pour améliorer la satisfaction des clients envers les salles de réunion dans d'autres bâtiments du campus.

À la fin d'une réunion, les participants sont invités à appuyer sur le bouton « Satisfait » ou « Mécontent » pour enregistrer leur satisfaction générale concernant la salle de réunion et l'équipement associé. Ces données sont ensuite utilisées pour identifier les salles de réunion présentant un équipement audiovisuel non fonctionnel ou d'autres problèmes.

La console AWS IoT 1-Click peut être utilisée pour configurer ce projet comme suit :

1. Depuis la console AWS IoT 1-Click, choisissez **Créer un projet**.
2. Pour le nom du projet, tapez **MeetingRoomSatisfaction**. Pour la description du projet, tapez **Project used to track customer meeting room satisfaction, including A/V equipment**. Choisissez **Suivant**.

3. UNDERProgrammer un modèle de périphériquechoisirDémarrer, puis choisissezTous les types de bouton de.
4. Pour Nom du modèle d'appareil, tapez **Satisfied**. Il s'agit du modèle utilisé pour tous les boutons intitulés « Satisfait ». Pour Action, choisissez Envoyer un e-mail.

 Note

Si le pilote de satisfaction de la salle de réunion est réussi, sous Action, vous pouvez choisir Action personnalisée à l'aide d'une fonction Lambda. Cette fonction Lambda personnalisée peut envoyer un e-mail ou stocker les données du bouton « Satisfait » dans une table Amazon DynamoDB pour analyse ultérieure. Pour de plus amples informations sur la création de fonctions Lambda, veuillez consulter le [Manuel du développeur AWS Lambda](#).

5. UNDERAjoutez un autre modèle de périphérique (si vous avez besoin de plusieurs périphériques par placement).choisirAddition, puis choisissezTous les types de bouton de. Pour Nom du modèle d'appareil, tapez **Unsatisfied**. Il s'agit du modèle utilisé pour tous les boutons intitulés « Mécontent ». Pour Action, choisissez Envoyer un e-mail.
6. Pour Valeur par défaut de l'e-mail obligatoire, saisissez une adresse e-mail. Pour Valeur par défaut de l'objet obligatoire, tapez **Meeting Room Feedback**. Pour Valeur par défaut du corps obligatoire, tapez **Either positive or negative meeting room feedback has been provided**.
7. Pour Clé d'attribut, tapez **Building**. Pour Valeur par défaut, tapez **Headquarters**. Le pilote de satisfaction de la salle de réunion se déroule dans le bâtiment du siège social de l'entreprise. Si le pilote fonctionne, il sera déployé dans les autres bâtiments de l'entreprise. Par conséquent, il est important de savoir à quel bâtiment correspondent les informations fournies par les périphériques de salles de réunion.
8. Dans la deuxième ligne de paire clé/valeur, pour Clé d'attribut, saisissez **Room**. Pour Valeur par défaut, tapez **TBD**. La **TBD** sera remplacée par un numéro de la salle de réunion lorsque les boutons y sont placés (à l'aide de l'application mobile AWS IoT 1-Click) ou de la console AWS IoT 1-Click).
9. Choisissez Créer un projet.

À l'aide de l'application mobile AWS IoT 1-Click, lorsqu'un bouton « Satisfait » est placé dans une salle de réunion, le boutonSatisfaitdevient associé à celui-ci et le modèle**TBD**est remplacée par le

numéro de la salle de réunion. Il en va de même lorsque le bouton « Mécontent » est placé dans une salle de réunion.

# Application mobile AWS IoT 1-Click

L'application mobile AWS IoT 1-Click vous permet :

- Configurer et surveiller facilement des périphériques AWS IoT 1-Click dans le champ utilisant une interface utilisateur similaire à celle de la console AWS IoT 1-Click.
- Configurer des informations d'identification Wi-Fi pour des périphériques AWS IoT 1-Click connectés au Wi-Fi (tel que AWS IoT Enterprise Button).

L'application mobile AWS IoT 1-Click est disponible pour iPhone et téléphones Android. Pour télécharger l'application, accédez à la [App Store](#) ou [Google](#) et recherchez AWS IoT 1-Click.

# Modèle de programmation AWS IoT 1-Click

Pour créer des applications à l'aide d'appareils AWS IoT 1-Click, les programmeurs utilisent le [API AWS IoT 1-Click](#) et l'[API AWS IoT 1-Click](#). L'API Devices interagit avec le composant AWS IoT 1-Click et gère les événements provenant de ces derniers. Ces événements incluent l'activation et la désactivation des périphériques, et la définition des formats d'événement et des actions (fonctions Lambda) qu'ils déclenchent. L'API Devices est intimement liée aux composants AWS qui résident dans la région dans laquelle le fabricant a inscrit les périphériques. C'est pourquoi [Régions AWS](#) peut être différente des régions dans lesquelles le client utilise les périphériques. L'API Projects interagit avec le service AWS IoT 1-Click Projects et est utilisée pour gérer globalement les périphériques AWS IoT 1-Click Projects, ce qui permet de :

- Regrouper les appareils en projets.
- Créer des modèles utilisés pour définir des actions pour tous les appareils du projet.
- Définir les attributs stockant les données contextuelles pertinentes pour le projet.

Vous pouvez utiliser le modèle de programmation AWS IoT 1-Click pour programmer des périphériques individuellement avec l'API Devices. Dans ce cas, vous utilisez le type de périphérique AWS IoT 1-Click. L'API définit des formats d'événement standard ainsi qu'une liste de méthodes formant l'interface de programmation de tous les périphériques du même type. Afin d'appeler les méthodes relatives à un type d'appareil donné, un programmeur peut se servir de l'[API InvokeDeviceMethod](#) et spécifier la méthode d'appareil comme paramètre.

Par exemple, tous les périphériques AWS IoT 1-Click de type « bouton » émettront des événements relatifs aux clics et disposeront de méthodes pour définir des fonctions de rappel invoquées lorsque l'on clique sur le périphérique. Pour de plus amples informations sur l'interface du bouton, veuillez consulter [Interfaces par type d'appareil](#). Voici le code pour définir cette fonction de rappel :

```
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);
```

Vous utilisez l'API Projects pour programmer une flotte de périphériques. À l'aide des API, vous définissez d'abord ce à quoi ressemble chaque placement, y compris les modèles et attributs des périphériques de chaque placement. Ceci fait, vous créez des placements avec des ID d'appareil spécifiques. Chaque placement suit le même modèle. Voici l'exemple de code pour le faire :

```
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)
```

## Événements de rappel AWS IoT 1-Click

AWS IoT 1-Click vous permet de vous abonner à des événements de périphérique en enregistrant des rappels. Un exemple de rappel est une fonction AWS Lambda détenue et implémentée par vous, le client AWS IoT 1-Click. Ce rappel est appelé chaque fois qu'un événement est disponible pour sa

consommation. Pour de plus amples informations sur les événements et leurs charges utiles, veuillez consulter les sections [Événements AWS IoT 1-Click](#) et [Événements de Health AWS IoT 1-Click](#).

## Événements AWS IoT 1-Click

Les appareils de type `button` publient un événement de clic chaque fois que l'on clique dessus. Vous pouvez vous abonner à cet événement en :

- Appelant la méthode `SetOnClickCallback` d'appareil sur un appareil.
- Configuriant correctement le projet associé, comme indiqué dans l'exemple précédent de création d'un code de projet.

Dans le code suivant, notez que la section `placementInfo` n'est présente que lorsqu'un placement est associé à l'appareil. Pour plus d'informations, consultez [Projets, modèles et placements](#).

```
{
  "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 "
```

```
    }  
  }  
}
```

## Événements de Health AWS IoT 1-Click

Les périphériques publient un événement d'état de santé en fonction des paramètres de santé calculés par le service AWS IoT 1-Click, mais vous définissez leurs seuils correspondants.

L'exemple suivant représente la charge utile JSON d'un événement d'état de santé pour l'appareil G030PMXXXXXXXXXX avec une durée de vie restante de 10 % (notez la paire clé-valeur "remainingLifeLowerThan":10).

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

## Méthodes de périphérique

Les méthodes de périphérique AWS IoT 1-Click sont des API qui sont prises en charge par un certain type de périphérique, comme illustré dans le tableau suivant. La liste complète des méthodes d'appareil prises en charge par n'importe quel appareil peut être récupérée en appelant [GetDeviceMethods](#).

Type de périphérique	Nom de méthode	Description
device	getDeviceHealthParameters	Permet d'obtenir les paramètres de santé du périphérique, par exemple <code>remainingLife</code> .
device	setDeviceHealthMonitorCallback	Définit un rappel à appeler lorsque les paramètres de santé du périphérique sont sous un certain seuil.
device	getDeviceHealthMonitorCallback	Obtient le rappel configuré qui est appelé lorsque les paramètres de santé sont sous un certain seuil.
button	setOnClickCallback	Définit un rappel à appeler lorsqu'un utilisateur clique sur le bouton.
button	getOnClickCallback	Permet d'obtenir le rappel configuré qui est appelé lorsqu'un utilisateur clique sur le bouton.

# Supervision d'AWS IoT 1-Click avec Amazon CloudWatch

AWS IoT 1-Click surveille automatiquement les périphériques pour vous et présente les métriques via [Amazon CloudWatch](#). Ces métriques sont présentées dans la région de périphérique où les périphériques ont été enregistrés par le fabricant. Pour de plus amples informations sur les régions d'appareils, veuillez consulter [Fonctionnement d'AWS IoT 1-Click](#). Vous pouvez trouver les métriques dans le tableau de bord Amazon CloudWatch, sous la rubrique `IoT1ClickEspace de noms`.

Amazon CloudWatch Events vous permet d'automatiser vos services AWS et de répondre automatiquement à des événements système tels que des problèmes de disponibilité d'application ou des modifications de ressource. Les événements des services AWS sont transmis à CloudWatch Events en temps quasi réel. Vous pouvez écrire des règles simples pour préciser les événements qui vous intéressent et les actions automatisées à effectuer quand un événement correspond à une règle. Les actions suivantes qui peuvent être déclenchées :

- Appel d'une fonction AWS Lambda.
- Invoquer la commande Exécuter Amazon EC2.
- Relais de l'événement à Amazon Kinesis Data Streams.
- Activation d'une machine d'état AWS Step Functions.
- Notification d'une rubrique Amazon SNS ou d'une file d'attente AWS SMS.

AWS IoT 1-Click suit et signale les mesures suivantes :

- `TotalEvents` suit le nombre d'événements publiés par les appareils. Cette métrique peut être consultée et représentée en graphique par événement de périphérique, projet, type de périphérique ou type de produit.
- `RemainingLife` représente le pourcentage approximatif de la durée de vie restante d'un périphérique. AWS IoT 1-Click présente ce nombre en se basant sur la notation du périphérique par le fabricant. Par exemple, si un bouton est conçu pour durer environ 2000 clics, et que 500 clics ont été enregistrés, la valeur `RemainingLife` correspond à 75 %. La métrique `RemainingLife` peut être consultée et représentée en graphique par événement d'appareil, projet, type d'appareil ou type de produit. Les clients peuvent utiliser les métriques `RemainingLife` pour configurer des alarmes qui se déclenchent lorsque les appareils passent sous un certain seuil. Les clients peuvent alors demander la durée de vie restante `RemainingLife` des appareils avec la méthode d'appareil `GetDeviceHealthParameters` pour identifier ceux qui ont une valeur `RemainingLife` basse.

- La métrique `CallbackInvocationErrors` suit les échecs d'invocation des rappels (fonctions Lambda) lorsqu'un appareil émet un événement. La métrique `CallbackInvocationErrors` peut être consultée et représentée en graphique par rappel invoqué (ARN de fonction Lambda définis comme rappels) ou par projet. Les clients peuvent configurer des alarmes pour le `CallbackInvocationErrors` pour être notifié lorsque AWS IoT 1-Click n'a pas été en mesure de router des événements depuis leurs périphériques vers leurs fonctions Lambda configurées.

Pour de plus amples informations, consultez le [Guide de l'utilisateur Amazon CloudWatch Events](#).

# Journalisation des appels d'API AWS IoT 1-Click

AWS IoT 1-Click (AWS IoT 1-Click) est intégré à AWS CloudTrail, service qui enregistre les actions effectuées par un utilisateur, un rôle ou un service AWS dans AWS IoT 1-Click. CloudTrail capture les appels d'API pour AWS IoT 1-Click. Les appels capturés incluent les appels depuis la console AWS IoT 1-Click et les appels de code vers les opérations d'API AWS IoT 1-Click. Si vous créez un journal de suivi, vous pouvez diffuser en continu les événements CloudTrail dans un compartiment Amazon S3, y compris les événements pour AWS IoT 1-Click. Si vous ne configurez pas de journal de suivi, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). Avec les informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à AWS IoT 1-Click, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur et la date de la demande, ainsi que d'autres informations.

Pour en savoir plus sur CloudTrail, notamment sur sa configuration et son activation, consultez le [Guide de l'utilisateur AWS CloudTrail](#).

## Informations AWS IoT 1-Click

CloudTrail est activé sur votre compte AWS lorsque vous créez le compte. Quand une activité d'événement prise en charge a lieu dans AWS IoT 1-Click, cette activité est enregistrée dans un événement CloudTrail avec d'autres événements de services AWS dans Historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour plus d'informations, consultez [Affichage des événements avec l'historique des événements CloudTrail](#).

Pour un enregistrement continu des événements dans votre compte AWS, y compris les événements pour AWS IoT 1-Click, créez un journal de suivi. Un journal de suivi permet à CloudTrail de livrer des fichiers journaux dans un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS. Le journal de suivi consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres services AWS pour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour plus d'informations, consultez les ressources suivantes :

- [Présentation de la création d'un journal de suivi](#)
- [Intégrations et services pris en charge par CloudTrail](#)

- [Configuration des Notifications de Amazon SNS pour CloudTrail](#)
- [Réception de fichiers journaux CloudTrail de plusieurs régions](#) et [Réception de fichiers journaux CloudTrail de plusieurs comptes](#)

AWS IoT 1-Click [API Appareils](#) prend en charge la journalisation des actions suivantes en tant qu'événements dans les fichiers journaux CloudTrail :

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

AWS IoT 1-Click [API Projets](#) prend en charge la journalisation des actions suivantes en tant qu'événements dans les fichiers journaux CloudTrail :

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

Chaque événement ou entrée du journal contient des informations sur la personne qui a généré la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification racine ou utilisateur AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez la section [Élément userIdentity CloudTrail](#).

## Exemple : AWS IoT 1-Click

Un journal de suivi est une configuration qui permet la livraison d'événements sous forme de fichiers journaux vers un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux CloudTrail peuvent contenir une ou plusieurs entrées de journal. Un événement représente une demande individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne constituent pas une série ordonnée retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'action DescribeDevice.

```
{
  "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"
}

```

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'action CreateProject.

```

{
  "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"
          }
        }
      }
    }
  }
}

```

```
}  
  }  
    }  
      }  
        }
```

# Intégration d'AWS CloudFormation

AWS IoT 1-Click est intégré à AWS CloudFormation, un langage commun pour décrire et provisionner toutes vos ressources d'infrastructure dans un environnement cloud (par exemple, Amazon EC2, Auto Scaling, Amazon SNS, etc.) AWS CloudFormation vous permet d'utiliser un simple fichier texte pour modéliser et provisionner, de manière automatisée et sécurisée, toutes les ressources nécessaires à vos applications dans toutes les régions et tous les comptes. Ce fichier sert de source unique de vérité pour votre environnement cloud. Pour plus d'informations, consultez le [Guide de l'utilisateur AWS CloudFormation](#) ainsi que les rubriques AWS IoT 1-Click (telles que [AWS::IoT1Click::Project](#)) dans la Guide de l'utilisateur AWS CloudFormation.

# Authentification et contrôle d'accès pour AWS IoT 1-Click

L'accès aux API 1-Click Ces informations d'identification doivent disposer d'autorisations pour accéder aux ressources AWS, telles qu'un projet ou un périphérique AWS IoT 1-Click Les sections suivantes décrivent comment utiliser AWS Identity and Access Management (IAM) et AWS IoT 1-Click

Chaque ressource AWS appartient à un compte AWS et les autorisations permettant de créer des ressources et d'y accéder sont régies par les stratégies d'autorisation. Un administrateur de compte peut attacher des stratégies d'autorisation aux identités IAM (c'est-à-dire aux utilisateurs, groupes et rôles), et certains services (tels qu'AWS Lambda) prennent également en charge l'attachement de stratégies d'autorisation aux ressources. Lors de l'octroi d'autorisations, l'administrateur décide qui les reçoit, à quelles ressources ces autorisations s'appliquent et les actions spécifiques qu'il souhaite autoriser sur ces ressources.

## Ressources et opérations AWS IoT 1-Click

Dans AWS IoT 1-Click Dans votre stratégie, vous utilisez un Amazon Resource Name (ARN) pour identifier la ressource à laquelle la stratégie s'applique. Ces ressources ont des noms ARN (Amazon Resource Name) uniques qui leur sont associés, comme cela est illustré dans le tableau suivant.

Type de ressource	Format ARN
Device	arn:aws:iot1click:region:account-id:devices/device-id
Projet	arn:aws:iot1click:region:account-id:projects/project-name

AWS IoT 1-Click Ces dernières sont appelées Actions dans IAM. Pour obtenir une liste des opérations disponibles, consultez le tableau en fin de rubrique.

# Utilisation des stratégies basées sur l'identité (stratégies IAM) pour AWS IoT 1-Click

Cette rubrique fournit des exemples de stratégies basées sur l'identité qui montrent comment un administrateur de compte peut attacher des stratégies d'autorisations à des identités IAM (autrement dit, des utilisateurs, groupes et rôles) et les autoriser ainsi à effectuer des opérations sur les ressources AWS IoT 1-Click

Un exemple de stratégie d'autorisation est exposé ci-dessous.

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

La stratégie comporte une seule déclaration qui accorde des autorisations pour une action AWS IoT 1-Click (`iot1click:CreateProject`) sur une ressource en utilisant l'Amazon Resource Name (ARN) de l'application. Dans ce cas, l'ARN spécifie un caractère générique (\*) pour indiquer que l'autorisation est accordée pour n'importe quelle ressource.

Pour obtenir le tableau de toutes les opérations API AWS IoT 1-Click et les ressources auxquelles elles s'appliquent, consultez [Autorisations d'API 1-Click Référence des actions, des autorisations et des ressources](#).

## Stratégies gérées par AWS (stratégies prédéfinies) pour AWS IoT 1-Click

Amazon Web Services est approprié pour de nombreux cas d'utilisation courants et fournit des stratégies IAM autonomes qui sont créées et administrées par AWS. Ces stratégies gérées AWS octroient les autorisations requises dans les cas d'utilisation courants pour que vous évitiez d'avoir à réfléchir aux autorisations qui sont requises. Pour de plus amples informations, veuillez consulter [Stratégies gérées par AWS](#) dans le Guide de l'utilisateur IAM.

Les stratégies gérées AWS suivantes, que vous pouvez attacher aux utilisateurs de votre compte, sont propres à AWS IoT 1-Click

- `AWSIoT1ClickFullAccess` : accorde un accès complet aux ressources AWS IoT 1-Click Les autorisations accordées incluent toutes les actions AWS IoT 1-Click pour gérer les périphériques et les projets.
- `AWSIoT1ClickReadOnlyAccess` : accorde un accès en lecture seule aux ressources AWS IoT 1-Click Cet accès permet à un utilisateur de dresser une liste des périphériques et des projets AWS IoT 1-Click et de vérifier la configuration d'un projet.

#### Note

Vous pouvez consulter ces stratégies d'autorisation en vous connectant à la console IAM (<https://console.aws.amazon.com/iam/>) et y rechercher le ou les noms de stratégie spécifiques.

Vous pouvez également créer vos propres stratégies IAM personnalisées afin d'accorder des autorisations pour les actions et les ressources AWS IoT 1-Click Vous pouvez attacher ces stratégies personnalisées aux utilisateurs ou groupes IAM qui nécessitent ces autorisations.

## Autorisations d'API 1-Click Référence des actions, des autorisations et des ressources

Lorsque vous configurez le contrôle d'accès dans le cloud AWS et que vous écrivez une stratégie d'autorisation que vous pouvez attacher à une identité IAM (stratégie basée sur une identité), vous pouvez utiliser le tableau suivant comme référence. Le tableau répertorie chaque opération d'API AWS IoT 1-Click, les actions correspondantes pour lesquelles vous pouvez accorder des autorisations permettant d'exécuter l'action et la ressource AWS pour laquelle vous pouvez accorder des autorisations. Vous spécifiez les actions dans le champ `Action` de la stratégie ainsi que la valeur des ressources dans le champ `Resource` de la stratégie.

Vous pouvez utiliser des clés de condition à l'échelle d'AWS IoT 1-Click Pour obtenir la liste complète des clés à l'échelle d'AWS, veuillez consulter [Clés disponibles](#) dans le Guide de l'utilisateur IAM.

**Note**

Pour spécifier une action, utilisez le préfixe `iot1click:` suivi du nom de l'opération d'API (par exemple, `iot1click:ListProjects`).

Opérations en 1 clic IoT	Autorisations requises (actions d'API)	Ressources
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	*

Opérations en 1 clic IoT	Autorisations requises (actions d'API)	Ressources
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
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

# Balisage de vos ressources AWS IoT 1-Click

Pour vous aider à gérer vos ressources AWS IoT 1-Click, vous pouvez affecter vos propres métadonnées à n'importe quelle ressource ARN à l'aide de balises. Ce chapitre décrit les balises et vous montre comment les créer.

## Principes de base des balises

Les balises vous permettent de classer vos ressources AWS IoT 1-Click de différentes manières, par exemple, par objectif, par propriétaire ou par environnement. Cette approche est utile lorsque vous avez de nombreuses ressources de même type : elle vous permet d'identifier rapidement une ressource spécifique en fonction des balises que vous lui avez attribuées. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez. Par exemple, vous pouvez définir un ensemble de balises pour plusieurs boutons appartenant à un responsable ou à un compte particulier. Vous pouvez rechercher et filtrer les ressources en fonction des balises que vous ajoutez. Nous vous recommandons de concevoir un ensemble de clés de balise répondant à vos besoins pour chaque type de ressource. L'utilisation d'un ensemble de clés de balise cohérent facilite la gestion de vos ressources. Pour plus d'informations, consultez [Stratégies de balisage AWS](#).

Vous pouvez également utiliser les balises pour classer les coûts par catégorie et en effectuer le suivi. Lorsque vous appliquez des balises à des ressources, AWS génère un rapport de répartition des coûts sous forme de fichier CSV faisant ressortir l'utilisation et les coûts regroupés par les balises. Vous pouvez appliquer des balises associées à des catégories métier (telles que les centres de coûts, les noms d'applications ou les propriétaires) pour organiser les coûts relatifs à divers services. Pour en savoir plus sur l'utilisation des balises pour la répartition des coûts, consultez [Utilisation des balises de répartition des coûts](#) dans le [Manuel de l'utilisateur AWS Billing and Cost Management](#).

À des fins de facilité d'utilisation, utilisez Tag Editor dans AWS Management Console, qui offre une façon centralisée et unifiée de créer et gérer vos balises. Pour plus d'informations, consultez [Utilisation de Tag Editor](#) dans le [Mise en route avec AWS Management Console](#).

Vous pouvez aussi gérer les balises à l'aide de l'interface de ligne de commande AWS et des API AWS IoT 1-Click. Vous pouvez associer des balises à des projets et des appareils AWS IoT 1-Click lorsque vous les créez à l'aide de l'outil `balises`. dans les commandes suivantes :

- [CreateProject](#) (Projets API)
- [Finaliser la réclamation](#) (Appareils API)

Vous pouvez ajouter, modifier ou supprimer des balises pour les ressources existantes en utilisant les commandes suivantes :

<p>AWS IoT 1-Click <a href="#">API Projets</a> (en utilisant les ARN de projet)</p>	<p>AWS IoT 1-Click <a href="#">API Appareils</a> (en utilisant les ARN de périphériques)</p>
<p><a href="#">TagResource</a></p>	<p><a href="#">Tag</a></p>
<p><a href="#">ListTagsForResource</a></p>	<p>Voir <a href="#">TagResource</a> (POST), <a href="#">TagListTagsForResource</a> (GET) et <a href="#">UntagResource</a> (DELETE).</p>
<p><a href="#">UntagResource</a></p>	

Vous pouvez modifier les clés et valeurs de balise, et vous pouvez retirer des balises d'une ressource à tout moment. Vous pouvez définir la valeur d'une balise sur une chaîne vide, mais vous ne pouvez pas définir la valeur d'une balise sur null. Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur. Si vous supprimez une ressource, toutes les balises associées à celle-ci sont également supprimées.

## Restrictions liées aux balises

Les restrictions de base suivantes s'appliquent aux balises :

- Nombre maximal de balises par ressource – 50
- Longueur de clé maximale : 127 caractères Unicode en UTF-8
- Longueur de valeur maximale : 255 caractères Unicode en UTF-8
- Les clés et valeurs de balise sont sensibles à la casse.
- N'utilisez pas le préfixe `aws:` dans les noms ou valeurs de vos balises, car celui-ci est réservé à AWS. Vous ne pouvez pas modifier ou supprimer des noms ou valeurs de balise ayant ce préfixe. Les balises avec ce préfixe ne sont pas comptabilisées comme vos balises pour la limite de ressources.
- Si votre schéma de balisage est utilisé pour plusieurs services et ressources, n'oubliez pas que d'autres services peuvent avoir des restrictions concernant les caractères autorisés. Les caractères généralement autorisés sont les lettres, les espaces et les chiffres représentables en UTF-8, ainsi que les caractères spéciaux suivants : `+ - = . _ : / @`

# Guide de l'utilisateur du AWS IoT Enterprise Button

Le AWS IoT Enterprise Button est un bouton Wi-Fi facile à configurer. Il est conçu pour que les entreprises et les développeurs intègrent facilement avec les flux de travail et les systèmes métier existants à l'aide d'AWS IoT 1-Click.

Le AWS IoT Enterprise Button prend en charge trois types de clics :

- Unique
- Double
- Presse longue

Pour que cela fonctionne correctement, vous devez configurer la connexion Wi-Fi du bouton utilisant l'application mobile AWS IoT 1-Click (iOS ou Android). Dans l'application, vous pouvez configurer la connexion Wi-Fi du bouton en vous connectant à votre compte AWS ou en appuyant sur l'icône Wi-Fi en haut à droite de l'application pour ignorer la connexion.

Une fois la connexion configurée et revendiquée via l'application mobile ou la console, le bouton doit clignoter en vert solide lorsqu'un clic simple, double ou long se produit.

Si vous soupçonnez qu'il y a un problème avec le bouton après l'avoir configuré, ce tableau peut vous aider à résoudre les problèmes.

Couleur	État	Recommandation
Clignotement blanc	Connexion au Wi-Fi en cours, acquisition de l'adresse IP ou connexion à AWS IoT.	N/A
Vert fixe	Connexion au Wi-Fi réussie, message envoyé à AWS IoT.	N/A
Clignotement bleu	Le bouton est en mode de configuration.	Attendez que le processus de configuration se termine.
Orange fixe	La connexion Wi-Fi n'est pas configurée.	Utilisez l'application mobile AWS IoT 1-Click pour configurer le Wi-Fi.

Couleur	État	Recommandation
Rouge : court, court, court	Une erreur est survenue pendant la connexion au réseau sans fil configuré.	Vérifiez si des paramètres réseau ont été modifiés ou si le bouton est trop loin du routeur Wi-Fi.
Rouge : court, court, long	Une erreur est survenue pendant l'obtention d'une adresse IP auprès du réseau sans fil.	Vérifiez les problèmes de connectivité réseau sans fil.
Rouge : court, long, court	Une erreur est survenue pendant la recherche d'un nom d'hôte.	Vérifiez les problèmes de connectivité réseau sans fil.
Rouge : court, long, long	Impossible de se connecter à AWS IoT.	Vérifiez les problèmes de connectivité réseau sans fil. Si aucun problème réseau n'existe et que le problème persiste, contactez <a href="#">Centre de support AWS</a> Indiquez le numéro de série de votre périphérique (DSN). Vous le trouverez au dos du bouton.
Rouge : long, court, court	Impossible d'établir une connexion sécurisée avec le serveur.	Utilisez les applications mobiles AWS IoT 1-Click pour vérifier si vous disposez du dernier firmware.
Rouge : long, court, long	Une erreur d'accès interdit HTTP 403 a été reçue.	Contactez le <a href="#">Centre de support AWS</a> et fournissez le DSN. Vous le trouverez au dos du bouton.

Couleur	État	Recommandation
Rouge : après 15 secondes, appuyez sur le bouton	Réinitialisation du bouton.	Vous pouvez réinitialiser la configuration Wi-Fi du bouton d'entreprise AWS IoT en appuyant sur le bouton pendant 15 secondes.

# Utilisation d'AWS IoT 1-Click avec l'interface de ligne de commande AWS

Pour démontrer l'utilisation de l'interface de ligne de commande AWS (AWS CLI), réfléchissez au scénario d'une entreprise d'élimination qui veut rationaliser son service de ramassage de bennes à ordures à l'aide d'AWS IoT 1-Click.

Dans ce scénario, chaque benne à ordures est jumelée à un AWS IoT Enterprise Button. Lorsqu'une benne à ordures est pleine, le client n'a qu'à appuyer sur le bouton associé pour demander son remplacement.

## Note

Tous les ID d'appareil AWS IoT Button commencent par « G030PM ».

Les étapes suivantes sont utilisées par la société d'élimination des déchets pour préparer le AWS IoT Enterprise Button pour l'utilisation par le client.

Pour préparer le AWS IoT Enterprise Button pour l'utilisation du client

1. La seule façon de configurer le Wi-Fi pour un AWS IoT Enterprise Button est d'utiliser l'application mobile AWS IoT 1-Click. Pour installer l'application, veuillez consulter [Application mobile AWS IoT 1-Click](#). Après avoir installé l'application, n'appuyez pas sur Connexion à un compte AWS (comme c'est normalement le cas). Dans cet exercice, nous voulons démontrer comment utiliser l'interface de ligne de commande AWS. Si vous appuyez sur Connexion à un compte AWS, les commandes `initiate-device-claim` et `finalize-device-claim` sont appelées pour vous, et nous voulons le faire « manuellement » à l'aide de l'interface de ligne de commande comme indiqué dans les étapes suivantes.
2. Pour la démonstration de l'interface de ligne de commande AWS, au lieu d'appuyer sur Se connecter au compte AWS, choisissez la petite icône Wi-Fi ronde dans le coin supérieur droit. Ensuite, choisissez Configurer le Wi-Fi. Scannez ou saisissez l'ID de l'appareil et suivez le reste des instructions de l'application mobile.
3. Si l'interface de ligne de commande AWS n'est pas installée, suivez les instructions décrites dans la rubrique [Installation de l'interface de ligne de commande AWS](#). Pour répertorier les commandes AWS CLI 1-Click disponibles, exécutez les deux commandes suivantes.

```
aws iot1click-projects help
```

```
aws iot1click-devices help
```

4. Pour associer le AWS IoT Enterprise Button maintenant connecté au Wi-Fi au compte AWS de la société d'élimination des déchets, exécutez la commande suivante à l'aide de l'ID de votre appareil.

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

Appuyez sur le bouton de l'appareil. Après un clignotement intermittent de lumière blanche, vous devriez voir un voyant vert fixe pendant environ une seconde. Si tel n'est pas le cas, répétez la procédure de connexion Wi-Fi précédente.

5. Une fois que vous avez vu le voyant vert fixe à l'étape précédente, exécutez la commande suivante (en utilisant la valeur d'ID de votre appareil).

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

La `"State": "CLAIMED"` indique que l'appareil est enregistré avec succès auprès du service AWS IoT 1-Click.

#### Note

Si le fabricant de l'appareil a fourni un code de demande commençant par « C- », vous pouvez utiliser simplement la commande `aws iot1click-devices claim-devices-by-claim-code` pour demander un ou plusieurs appareils à l'aide d'un seul code de demande, comme illustré dans l'exemple suivant.

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

```
}
```

Dans cet exemple, "Total": 9 indique que les neuf appareils associés au code de demande C-123EXAMPLE ont été demandés avec succès par le service AWS IoT 1-Click.

6. Ensuite, vous vous préparez à créer un projet AWS IoT 1-Click approprié pour la société de cession en créant un fichier texte JSON nommé `create-project.json`. Le fichier zip contient les éléments suivants :

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

Les paires clé-valeur `placementTemplate` et `deviceTemplates` sont les attributs qui seront appliqués à tous les boutons qui font partie du projet `SeattleDumpsters`. Pour créer ce projet, exécutez la commande suivante (qui suppose que `create-project.json` se trouve dans le [Répertoire de travail actuel](#) de l'invite de commande AWS).

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

Pour afficher le projet nouvellement créé, exécutez la commande suivante.

```
aws iot1click-projects list-projects
{
  "projects": [
    {
      "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/SeattleDumpsters",
      "projectName": "SeattleDumpsters",
```

```

        "createdDate": 1563483100,
        "updatedAt": 1563483100,
        "tags": {}
    }
]
}

```

Pour plus de détails, exécutez la commande `describe-project`, comme suit.

```

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": {}
  }
}

```

7. Avec le projet créé pour la région de Seattle, vous créez ensuite un emplacement pour une benne à ordures particulière (pour le client 217), comme suit. Les guillemets échappés sont nécessaires pour Windows.

```

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

```

Pour afficher le placement nouvellement créé, exécutez la commande suivante.

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

Pour plus de détails, exécutez la commande `describe-placement`, comme suit.

```
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. Bien que l'appareil soit désormais associé au compte AWS IoT 1-Click de la société de cession, il n'est pas associé au placement. Pour ce faire, exécutez la commande suivante.

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

Pour associer l'appareil au placement, exécutez la commande suivante.

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

Pour confirmer la commande précédente, exécutez `get-devices-in-placement` à nouveau.

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

Pour plus de détails, exécutez la commande `describe-device`, comme suit (remarquez le basculement de `iot1click-projects` vers `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": {}
  }
}
```

Étant donné qu'il n'y a actuellement qu'un seul appareil, la commande suivante produit des résultats similaires.

```
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. Pour vérifier que l'appareil fonctionne correctement, exécutez la commande suivante. Ajustez les horodatages de manière appropriée, qui sont au [format ISO 8061](#).

```
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\\\":
\\\"fe8798a6c97c62ef8756b80eeefdcb2280f3352f82faa8080c74cc4f4a4d1811\\\",
\\\"remainingLife\\\": 99.85000000000001, \\\"testMode\\\": false}"
    }
  ]
}
```

Ici, nous voyons qu'un événement en un clic (`\"clickType\": \"SINGLE\"`) s'est produit le 2019-07-18T 23:47:55.015Z. Double-cliquez maintenant sur l'appareil (deux appuis successifs

rapides sur le bouton), puis exécutez à nouveau la commande. Notez maintenant l'événement de double-clic (`\\"clickType\\": \\"DOUBLE\\"`), similaire à ce qui suit.

```
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\\":
      "\\"fe8798a6c97c62ef8756b80eeefdcb2280f3352f82faa8080c74cc4f4a4d1811\\"",
      "\\"remainingLife\\": 99.850000000000001, "\\"testMode\\": false}"
    },
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\\"clickType\\": \\"DOUBLE\\"",
      "\\"reportedTime\\": \\"2019-07-19T00:14:41.353Z\\"", "\\"certificateId\\":
      "\\"fe8798a6c97c62ef8756b80eeefdcb2280f3352f82faa8080c74cc4f4a4d1811\\"",
      "\\"remainingLife\\": 99.8, "\\"testMode\\": false}"
    }
  ]
}
```

10. Chaque type d'appareil a un ensemble de méthodes d'appareil appelables. Pour répertorier les méthodes disponibles pour votre type d'appareil, exécutez la commande `get-device-methods`, comme suit.

```
aws iot1click-devices get-device-methods --device-id G030PM0123456789
{
  "DeviceMethods": [
    {
      "MethodName": "getDeviceHealthParameters"
    },
    {
```

```
        "MethodName": "setDeviceHealthMonitorCallback"
    },
    {
        "MethodName": "getDeviceHealthMonitorCallback"
    },
    {
        "MethodName": "setOnClickCallback"
    },
    {
        "MethodName": "getOnClickCallback"
    }
]
}
```

Pour appeler l'une des méthodes disponibles, utilisez la commande `invoke-device-method`, comme indiqué ci-dessous.

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

Ici, `invoke-device-method.json` contient ce qui suit.

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

#### Note

Les méthodes `get` (telles que `getDeviceHealthParameters`) ne s'attendent pas à des paramètres. Par conséquent, la ligne `"DeviceMethodParameters": ""` dans le fichier JSON ne peut pas être utilisée (cela entraînera ceci : `An error occurred (InvalidRequestException) when calling the InvokeDeviceMethod operation: A request parameter was invalid.`)

11. En exécutant `aws iot1click-devices list-devices --device-type button`, vous pouvez voir que la valeur par défaut pour `Enabled` est `false`. La commande suivante définit cette clé sur `true`.

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

Pour le redéfinir sur `false`, exécutez la commande précédente, à nouveau en utilisant l'argument `--no-enabled`.

12. Si les informations client changent, vous pouvez mettre à jour les informations de placement d'un appareil comme indiqué ci-après (notez le basculement de `iot1click-devices` vers `iot1click-projects`). Exécutez la commande suivante pour afficher les informations actuelles de `customer217` (voir `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,
    "updatedAt": 1563488454
  }
}
```

Ensuite, exécutez la commande suivante pour mettre à jour les attributs de téléphone et d'emplacement du client.

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

Ici, `update-placement.json` contient ce qui suit.

```
{
  "projectName": "SeattleDumpsters",
  "placementName": "customer217",
  "attributes": {
```

```
    "phone": "206-266-1000",
    "location": "410 Terry Ave N Seattle, WA 98109"
  }
}
```

Pour consulter cette mise à jour, exécutez `describe-placement` à nouveau, comme indiqué.

```
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. Pour mettre à jour les informations du projet, utilisez la commande `update-project`. Un projet contient généralement plusieurs placements de clients. Voici les informations existantes sur le projet `SeattleDumpster`.

```
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": {}
  }
}

```

Pour remplacer « Toutes les bennes à ordures dans la région de Seattle » par « Toutes les bennes à ordures (jardin de jardin)déchet, recyclage et déchets) dans la région de Seattle », exécutez la commande suivante.

```

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

```

Vous pouvez voir que la valeur de la clé "description" a été mise à jour pour tous les emplacements SeattleDumpsters.

```

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. Vous pouvez utiliser des balises pour appliquer des méta-informations aux ressources de projet (iot1click-projects) et aux ressources de placement (iot1click-devices), comme suit.

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

Ici, `projects-tag-resource.json` contient ce qui suit.

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

Pour répertorier les balises de la ressource du projet, exécutez la procédure suivante.

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

Pour afficher les balises de projet dans le contexte, exécutez la procédure suivante.

```
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"
    }
  }
}

```

Pour découvrir les appareils Amazon Resource Names (ARN), exécutez la procédure suivante.

```

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": {}
    }
  ]
}

```

Pour ajouter des balises à l'appareil précédent, exécutez la procédure suivante.

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

Ici, `devices-tag-resources.json` contient ce qui suit (notez l'imbrication requise de `ResourceArn` et `Tags`).

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

Pour répertorier les balises de la ressource de l'appareil, exécutez la procédure suivante.

```
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"
  }
}
```

Pour afficher les balises d'appareil dans le contexte, exécutez `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. À ce stade, vous pouvez associer une action à une pression sur le bouton de l'appareil, par exemple déclencher une fonction AWS Lambda ou envoyer un message Amazon SNS. Vous pouvez le faire facilement à l'aide de la console AWS IoT 1-Click (la [Modèle de programmation AWS IoT 1-Click](#) est également une option). Une fois les actions appropriées associées à l'appareil, vous pouvez l'amener à l'emplacement du client et le connecter à son réseau Wi-Fi en suivant la procédure décrite aux étapes 1 et 2.

## Démontage du périphérique AWS IoT 1-Click

Les étapes suivantes décrivent comment inverser (annuler) les étapes précédentes.

1. Pour retirer les balises d'une ressource de projet, exécutez la commande suivante.

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

Cela supprime la balise `Manager` du projet, comme indiqué ci-dessous.

```
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. Pour retirer les balises d'une ressource d'appareil, exécutez la commande suivante.

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

Cela supprime les balises de l'appareil, comme indiqué ci-dessous (notez la liste vide "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": {}
    }
  ]
}

```

```
}
```

3. Pour dissocier un appareil d'un emplacement, exécutez la commande suivante.

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

Comme vous pouvez le voir dans ce qui suit, le placement `customer217` n'est plus associé à aucun appareil.

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

4. Pour supprimer un placement d'un projet, exécutez la commande suivante.

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

Comme vous pouvez le voir dans ce qui suit, le projet `SeattleDumpsters` n'a pas de placements car le placement `customer217` était le seul à l'intérieur de `SeattleDumpsters`.

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

5. Pour supprimer un projet, exécutez la commande suivante.

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

Comme vous pouvez le voir dans ce qui suit, tous les projets sont supprimés car `SeattleDumpsters` était le seul projet associé à votre compte AWS IoT 1-Click.

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

```
}
```

Si, par exemple, vous souhaitez laisser un ami essayer votre appareil à l'aide de son compte AWS, vous devez d'abord annuler la demande de votre appareil depuis votre compte AWS IoT 1-Click, comme suit.

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

L'appareil peut désormais être utilisé avec n'importe quel compte AWS IoT 1-Click.

## Annexe AWS IoT 1-Click

Cette section fournit les informations supplémentaires AWS IoT 1-Click.

### Périphériques AWS IoT 1-Click

Produit	Type d'appareil	Préfixe de l'ID d'appareil	Pour demander l'appareil	Lien d'achat	Région de l'appareil <sup>†</sup>
Bouton IoT pour AWS IoT (États-Unis, UE et Japon)	Button	P5SJVQ (les 6 premiers caractères de l'ID de l'appareil)	Dans l'application mobile AWS IoT 1-Click, entrez l'ID de l'appareil pour configurer le Wi-Fi et revendiquer l'appareil.	<a href="#">Seeed Studio Bazaar</a>	US West (Oregon)
Bouton Sercomm IoT (États-Unis uniquement)	Button	7VT4EQ (les 6 premiers caractères de l'ID du périphérique)	Dans l'application mobile AWS IoT 1-Click ou dans la console AWS IoT 1-Click, entrez le code de réclamation que vous avez obtenu lors de l'achat du ou des appareils.	<a href="#">Boutique en ligne Sercomm Mobilelot</a>	US West (Oregon)

Produit	Type d'appareil	Préfixe de l'ID d'appareil	Pour demander l'appareil	Lien d'achat	Région de l'appareil <sup>†</sup>
			Vous pouvez également revendiquer l'appareil en saisissant l'ID de l'appareil dans l'application mobile AWS IoT 1-Click.		
Bouton SORACOM LTE-M (Japon uniquement)	Button	7MF6JK (les 6 premiers caractères de l'ID d'appareil)	Dans l'application mobile AWS IoT 1-Click, entrez l'ID de l'appareil pour demander l'appareil.	<a href="https://soracom.com">SORACOM</a>	US West (Oregon)
AWS IoT Enterprise Button AWS IoT (États-Unis, UE et Japon)	Button	G030PM (les 6 premiers caractères de l'ID d'appareil)	Dans l'application mobile AWS IoT 1-Click, entrez l'ID de l'appareil pour configurer le Wi-Fi et pour demander l'appareil.	Interrompu	US West (Oregon)

Produit	Type d'appareil	Préfixe de l'ID d'appareil	Pour demander l'appareil	Lien d'achat	Région de l'appareil <sup>†</sup>
Bouton AT&T LTE-M (États-Unis seulement)	Button	B9GHXT (les 6 premiers caractères de l'ID du périphérique)	Dans l'application mobile AWS IoT 1-Click ou dans la console AWS IoT 1-Click, entrez le code de réclamation que vous avez obtenu lorsque vous avez acheté le ou les appareils. Vous pouvez également revendiquer l'appareil en saisissant l'ID de l'appareil dans l'application mobile AWS IoT 1-Click.	Interrompu	US West (Oregon)

<sup>†</sup>Pour de plus amples informations sur les régions d'appareils, veuillez consulter [Périphériques AWS IoT 1-Click](#).

**Note**

AWS IoT 1-Click ne prend pas en charge les boutons AWS IoT dont le numéro de série (DSN) commence par G030JF, G030MD et G030PT. Pour savoir comment connecter ces boutons au Cloud AWS IoT (sans utiliser AWS IoT 1-Click), veuillez consulter [Bouton de tableau de bord programmable](#).

## Limites de service AWS IoT 1-Click

- Vous pouvez disposer d'un maximum de 5 modèles d'appareil par modèle de placement. Cela correspond à 5 appareils par emplacement.
- 512 projets AWS IoT 1-Click [Région AWS](#) par compte.
- Il y a un maximum de 50 balises par ressource AWS IoT 1-Click. Les balises sont des paires clé/valeur (métadonnées) qui peuvent être utilisées pour gérer une ressource. Pour plus d'informations, consultez [Stratégies de balisage AWS](#).

# Historique du document pour le Manuel du développeur

Le tableau suivant répertorie la documentation liée à cette version d'AWS IoT 1-Click.

- Version de l'API : dernière en date
- Dernière mise à jour de la documentation 22 octobre 2018

Modification	Description	Date
Version	Première version de la présente documentation.	14 mai 2018
Éditorial	Améliorations rédactionnelles.	31 mai 2018
Éditorial	Tableau des appareils pris en charge mis à jour.	22 octobre 2018

# Glossaire AWS

Pour connaître la terminologie AWS la plus récente, veuillez consulter le [Glossaire AWS](#) dans les Références générales AWS.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.