Guide de l'utilisateur

AWS Messagerie push destinée aux utilisateurs finaux

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Messagerie push destinée aux utilisateurs finaux: Guide de l’utilisateur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Qu'est-ce que AWS Final User Messaging Push ?

> **Note**
>
> Les fonctionnalités de notification push d'Amazon Pinpoint sont désormais appelées « AWS End User Messaging ».

Avec AWS End User Messaging Push, vous pouvez engager les utilisateurs de vos applications en envoyant des notifications push via un canal de notification push. Nous prenons en charge le service de notification push d'Apple (APNs), Firebase Cloud Messaging (FCM), Amazon Device Messaging (ADM) et Baidu Push.

Rubriques

Utilisez-vous la messagerie push pour la première fois à l'utilisateur AWS final ?

Si vous utilisez AWS End User Messaging Push pour la première fois, nous vous recommandons de commencer par lire les sections suivantes :

- Configuration d'un Compte AWS
- Commencer à utiliser AWS Final User Messaging Push
- Création d'une application et activation des canaux push

Caractéristiques de la messagerie push destinée aux utilisateurs AWS finaux

Vous pouvez envoyer des notifications push à vos applications à l'aide de canaux séparés pour les services de notification push suivants :

- Firebase Cloud Messaging (FCM)
- Service de notifications push Apple (APNs)

> **Note**
>
> Vous pouvez l'utiliser APNs pour envoyer des messages à des appareils iOS tels que les iPhones et les iPads, ainsi qu'au navigateur Safari sur les appareils macOS, tels que les ordinateurs portables et de bureau Mac.

- Baidu Cloud Push
- Amazon Device Messaging (ADM)

Accès à AWS la messagerie Push pour les utilisateurs finaux

Expliquez brièvement les différentes manières d'accéder au service, que ce soit par console, CLI ou API.

Vous pouvez gérer les messages push destinés aux utilisateurs AWS finaux à l'aide des interfaces suivantes :

AWS Console push de messagerie à l'utilisateur final

Interface Web dans laquelle vous créez et gérez les ressources push de messagerie utilisateur AWS final. Si vous vous êtes inscrit à un Compte AWS, vous pouvez accéder à la console AWS Final User Messaging Push depuis le AWS Management Console.

AWS Command Line Interface

Interagissez avec les AWS services à l'aide des commandes de votre interface de ligne de commande. AWS Command Line Interface Il est pris en charge sur Windows, macOS et Linux. Pour plus d'informations à ce sujet AWS CLI, consultez le Guide de AWS Command Line Interface l'utilisateur. Vous trouverez les commandes push de messagerie à l'utilisateur AWS final dans la AWS CLI référence des commandes.

AWS SDKs

Si vous êtes un développeur de logiciels qui préfère créer des applications à l'aide d'un langage spécifique APIs au lieu de soumettre une demande via HTTP ou HTTPS, vous trouverez AWS des bibliothèques, des exemples de code, des didacticiels et d'autres ressources. Ces bibliothèques fournissent des fonctions de base qui automatisent les tâches, telles que la

signature cryptographique de vos demandes, les nouvelles tentatives et la gestion des réponses aux erreurs. Ces fonctions vous aident à démarrer plus efficacement. Pour de plus amples informations, veuillez consulter Outils pour créer sur AWS.

Disponibilité par région

AWS La messagerie push destinée aux utilisateurs finaux est disponible Régions AWS dans plusieurs pays d'Amérique du Nord, d'Europe, d'Asie et d'Océanie. Dans chaque région, AWS gère plusieurs zones de disponibilité. Ces zones de disponibilité sont physiquement isolées mais sont reliées par des connexions réseau privées, à latence faible, à débit élevé et à forte redondance. Ces zones de disponibilité sont utilisées pour fournir des niveaux très élevés de disponibilité et de redondance, tout en minimisant le temps de latence.

Pour en savoir plus Régions AWS, consultez Spécifiez ce que Régions AWS votre compte peut utiliser dans le Référence générale d'Amazon Web Services. Pour obtenir la liste de toutes les régions dans lesquelles la messagerie push pour les utilisateurs AWS finaux est actuellement disponible et le point de terminaison de chaque région, consultez la section Points de terminaison et quotas pour l'API Amazon Pinpoint AWS et les points de terminaison de service dans le. Référence générale d'Amazon Web Services Pour plus d'informations sur le nombre de zones de disponibilité disponibles dans chaque région, consultez Infrastructure mondiale AWS.

Configuration d'un Compte AWS

Avant de pouvoir utiliser AWS la messagerie push à l'utilisateur final pour envoyer des notifications push à votre application, vous devez d'abord obtenir Compte AWS une autorisation IAM suffisante. Cela Compte AWS peut également être utilisé pour d'autres services de l' AWS écosystème.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d’un utilisateur doté d’un accès administratif

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d’inscription, vous recevrez un appel téléphonique ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l’utilisateur racine a accès à l’ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l’utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d’un accès à d’autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d’autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d’un ensemble d’autorisations dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l’utilisateur AWS IAM Identity Center .

Commencer à utiliser AWS Final User Messaging Push

Pour configurer AWS End User Messaging Push afin qu'il puisse envoyer des notifications push à vos applications, vous devez d'abord fournir les informations d'identification qui autorisent AWS End User Messaging Push à envoyer des messages à votre application. Les informations d'identification que vous fournissez dépendent du système de notification push que vous utilisez :

- Pour les informations d'identification du service de notification push (APN) Apple, consultez les sections Obtenir une clé de chiffrement et un identifiant de clé auprès d'Apple et Obtenir un certificat de fournisseur auprès d'Apple dans la documentation destinée aux développeurs Apple.
- Pour les informations d'identification Firebase Cloud Messaging (FCM), elles peuvent être obtenues via la console Firebase, voir Firebase Cloud Messaging.
- Pour les informations d'identification Baidu, consultez Baidu.
- Pour les informations d'identification Amazon Device Messaging (ADM), consultez Obtenir des informations d'identification.

Création d'une application et activation des canaux push

Avant de pouvoir utiliser AWS la messagerie push à l'utilisateur final pour envoyer des notifications push, vous devez d'abord créer une application et activer le canal de notifications push.

Contextuel

Application

Une application est un conteneur de stockage pour tous vos paramètres push de messagerie utilisateur AWS final. L'application enregistre également les paramètres de vos chaînes, campagnes et parcours Amazon Pinpoint.

Clé

Clé de signature privée utilisée par AWS End User Messaging Push pour signer cryptographiquement les jetons APNs d'authentification. La clé de signature est disponible dans votre compte de développeur Apple.

Si vous fournissez une clé de signature,AWS End User Messaging Push utilise un jeton pour s'authentifier APNs pour chaque notification push que vous envoyez. Avec votre clé de signature, vous pouvez envoyer des notifications push aux environnements de APNs production et de sandbox.

Contrairement aux certificats, votre clé de signature n'expire pas. Vous ne la fournissez qu'une seule fois et vous n'avez pas besoin de la renouveler. Vous pouvez utiliser la même clé de signature pour plusieurs applications. Pour plus d'informations, consultez Communiquer à APNs l'aide de jetons d'authentification dans l'aide de Xcode.

Certificat

Certificat TLS utilisé par AWS End User Messaging Push pour s'authentifier APNs lorsque vous envoyez des notifications push. Un APNs certificat peut prendre en charge à la fois les environnements de production et les environnements sandbox, ou il ne peut prendre en charge que l'environnement sandbox. Le certificat est disponible dans votre compte de développeur Apple.

Un certificat expire au bout d'un an. Dans ce cas, vous devez créer un nouveau certificat, que vous fournissez ensuite à AWS End User Messaging Push pour renouveler les envois de

notifications push. Pour plus d'informations, voir Communiquer avec un certificat TLS dans APNs l'aide de Xcode.

Prérequis

Avant de pouvoir utiliser un canal push, vous devez disposer d'informations d'identification valides pour le service push. Pour plus d'informations sur l'obtention des informations d'identification, consultezCommencer à utiliser AWS Final User Messaging Push.

Procédure

Suivez ces instructions pour créer une application et activer l'un des canaux push. Pour terminer cette procédure, il vous suffit de saisir le nom de l'application. Vous pouvez activer ou désactiver n'importe quel canal push ultérieurement.

1. Ouvrez la console AWS Final User Messaging Push à l'adresse https://console.aws.amazon.com/push-notifications/.
2. Choisissez Créer une application.
3. Dans Nom de l'application, entrez le nom de votre application.
4. (Facultatif) Suivez cette étape facultative pour activer le service de notification push Apple (APNs).

    a. Pour le service de notification push Apple (APNs), sélectionnez Activer.

    b. Pour le type d'authentification par défaut, choisissez l'une des options suivantes :

        i. Si vous choisissez Key credentials, fournissez les informations suivantes depuis votre compte développeur Apple. AWS Final User Messaging Push a besoin de ces informations pour créer des jetons d'authentification.

            - ID de clé : ID attribué à votre clé de signature.
            - Identifiant de solution groupée : ID attribué à votre application iOS.
            - Identifiant d'équipe : ID attribué à l'équipe chargée de votre compte Apple Developer.
            - Clé d'authentification : fichier .p8 que vous téléchargez depuis votre compte de développeur Apple lorsque vous créez une clé d'authentification.

        ii. Si vous choisissez Certificate credentials (Informations d'identification de certificat), fournissez les informations suivantes :

- SSL certificate (Certificat SSL) : fichier .p12 de votre certificat TLS.
- Mot de passe de certificat : si vous avez attribué un mot de passe à votre certificat, entrez-le ici.
- Type de certificat : sélectionnez le type de certificat à utiliser.

5. (Facultatif) Suivez cette étape facultative pour activer Firebase Cloud Messaging (FCM).

   a. Pour Firebase Cloud Messaging (FCM), sélectionnez Activer.
   b. Pour le type d'authentification par défaut, choisissez l'une des options suivantes :

      i. Pour les informations d'identification Token (recommandé), choisissez Choose files, puis choisissez le fichier JSON de service.
      ii. Pour les informations d'identification clés, entrez votre clé dans la clé API.

6. (Facultatif) Suivez cette étape facultative pour activer le Baidu Cloud Push.

   a. Pour Baidu Cloud Push, sélectionnez Activer.
   b. Pour la clé d'API, entrez votre clé d'API.
   c. Dans le champ Clé secrète, entrez votre clé secrète.

7. (Facultatif) Suivez cette étape facultative pour activer Amazon Device Messaging.

   a. Pour Amazon Device Messaging, sélectionnez Activer.
   b. Dans le champ ID client, entrez votre identifiant client.
   c. Dans le champ Secret client, entrez votre secret client.

8. Choisissez Créer une application.

Désactiver les canaux push

Suivez ces instructions pour désactiver l'un des canaux push.

1. Ouvrez la console AWS Final User Messaging Push à l'adresse https://console.aws.amazon.com/push-notifications/.
2. Choisissez l'application qui contient vos informations d'identification push.
3. (Facultatif) Pour le service de notification push Apple (APNs), désactivez Activer.
4. (Facultatif) Pour Firebase Cloud Messaging (FCM), désactivez Activer.
5. (Facultatif) Pour Baidu Cloud Push, désactivez Enable.
6. (Facultatif) Pour Amazon Device Messaging, désactivez Activer.
7. Sélectionnez Enregistrer les modifications.

Envoi d'un message

L'API push de messagerie utilisateur AWS final peut envoyer des notifications push transactionnelles à des identifiants d'appareils spécifiques. Cette section contient des exemples de code complets que vous pouvez utiliser pour envoyer des notifications push via l'API push de messagerie utilisateur AWS final à l'aide d'un AWS SDK.

Vous pouvez utiliser ces exemples pour envoyer des notifications push via n'importe quel service de notification push pris en charge par AWS End User Messaging Push. Actuellement, AWS End User Messaging Push prend en charge les canaux suivants : Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push et Amazon Device Messaging (ADM).

Pour plus d'exemples de code sur les points de terminaison, les segments et les canaux, voir Exemples de code.

> **Note**
>
> Lorsque vous envoyez des notifications push via le service Firebase Cloud Messaging (FCM), utilisez le nom du service `GCM` dans votre appel à l'API push de messagerie utilisateur AWS final. Le service Google Cloud Messaging (GCM) a été interrompu par Google le 10 avril 2018. Toutefois, l'API push de messagerie utilisateur AWS final utilise le nom du `GCM` service pour les messages qu'elle envoie via le service FCM afin de maintenir la compatibilité avec le code API écrit avant l'arrêt du service GCM.

GCM (AWS CLI)

L'exemple suivant utilise send-messages pour envoyer une notification Push GCM avec le. AWS CLI*token*Remplacez-le par le jeton unique de l'appareil et *611e3e3cdd47474c9c1399a50example* par l'identifiant de votre application.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request file://myfile.json \
--region us-west-2

Contents of myfile.json:
{
```

```
  "Addresses": {
    "token": {
      "ChannelType" : 'GCM'
    }
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "URL",
      "Body": "This is a sample message",
      "Priority": "normal",
      "SilentPush": True,
      "Title": "My sample message",
      "TimeToLive": 30,
      "Url": "https://www.example.com"
      }
  }
}
```

L'exemple suivant utilise send-messages pour envoyer une notification GCM Push, à l'aide de toutes les clés existantes, avec le. AWS CLI*token*Remplacez-le par le jeton unique de l'appareil et *611e3e3cdd47474c9c1399a50example* par l'identifiant de votre application.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\"notification\": {\n \"title\": \"string\",\n \"body\":
 \"string\",\n \"android_channel_id\": \"string\",\n \"body_loc_args\": [\n \"string
\"\n ],\n \"body_loc_key\": \"string\",\n \"click_action\": \"string\",\n \"color\":
 \"string\",\n \"icon\": \"string\",\n \"sound\": \"string\",\n \"tag\": \"string
\",\n \"title_loc_args\": [\n \"string\"\n ],\n \"title_loc_key\": \"string\"\n },
\"data\":{\"message\":\"hello in data\"} }",
      "TimeToLive" : 309744
     }
   },
  "Addresses": {
    "token": {
      "ChannelType":"GCM"
      }
   }
}'
```

```
\ --region us-east-1
```

L'exemple suivant utilise send-messages pour envoyer une notification Push GCM avec la charge utile du FCMv1 message à l'aide du. AWS CLI*token*Remplacez-le par le jeton unique de l'appareil et *611e3e3cdd47474c9c1399a50example* par l'identifiant de votre application.

```
aws pinpoint send-messages \
--application-id 6a2dafd84bec449ea75fb773f4c41fa1 \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\n \"fcmV1Message\": \n {\n \"message\" :{\n  \"notification
\": {\n \"title\": \"string\",\n \"body\": \"string\"\n },\n \"android\": {\n
 \"priority\": \"high\",\n \"notification\": {\n \"title\": \"string\",\n \"body
\": \"string\",\n \"icon\": \"string\",\n \"color\": \"string\",\n \"sound\":
 \"string\",\n \"tag\": \"string\",\n \"click_action\": \"string\",\n \"body_loc_key
\": \"string\",\n \"body_loc_args\": [\n \"string\"\n ],\n \"title_loc_key
\": \"string\",\n \"title_loc_args\": [\n \"string\"\n ],\n \"channel_id\":
 \"string\",\n \"ticker\": \"string\",\n \"sticky\": true,\n \"event_time\":
 \"2024-02-06T22:11:55Z\",\n \"local_only\": true,\n \"notification_priority\":
 \"PRIORITY_UNSPECIFIED\",\n \"default_sound\": false,\n \"default_vibrate_timings
\": true,\n \"default_light_settings\": false,\n \"vibrate_timings\": [\n \"22s
\"\n ],\n \"visibility\": \"VISIBILITY_UNSPECIFIED\",\n \"notification_count\": 5,
\n \"light_settings\": {\n \"color\": {\n \"red\": 1,\n \"green\": 2,\n \"blue\":
 3,\n \"alpha\": 6\n },\n \"light_on_duration\": \"112s\",\n \"light_off_duration
\": \"1123s\"\n },\n \"image\": \"string\"\n },\n \"data\": {\n \"dataKey1\":
 \"priority message\",\n \"data_key_3\": \"priority message\",\n \"dataKey2\":
 \"priority message\",\n \"data_key_5\": \"priority message\"\n },\n \"ttl\":
 \"10023.32s\"\n },\n \"apns\": {\n \"payload\": {\n \"aps\": {\n \"alert\": {\n
 \"subtitle\": \"string\",\n \"title-loc-args\": [\n \"string\"\n ],\n \"title-loc-
key\": \"string\",\n \"launch-image\": \"string\",\n \"subtitle-loc-key\": \"string
\",\n \"subtitle-loc-args\": [\n \"string\"\n ],\n \"loc-args\": [\n \"string
\"\n ],\n \"loc-key\": \"string\",\n \"title\": \"string\",\n \"body\": \"string
\"\n },\n \"thread-id\": \"string\",\n \"category\": \"string\",\n \"content-
available\": 1,\n \"mutable-content\": 1,\n \"target-content-id\": \"string\",\n
 \"interruption-level\": \"string\",\n \"relevance-score\": 25,\n \"filter-criteria
\": \"string\",\n \"stale-date\": 6483,\n \"content-state\": {},\n \"timestamp\":
 673634,\n \"dismissal-date\": 4,\n \"attributes-type\": \"string\",\n \"attributes
\": {},\n \"sound\": \"string\",\n \"badge\": 5\n }\n }\n },\n \"webpush\": {\n
 \"notification\": {\n \"permission\": \"granted\",\n \"maxActions\": 2,\n \"actions
\": [\n \"title\"\n ],\n \"badge\": \"URL\",\n \"body\": \"Hello\",\n \"data\": {\n
 \"hello\": \"hey\"\n },\n \"dir\": \"auto\",\n \"icon\": \"icon\",\n \"image\":
```

```
 \"image\",\n \"lang\": \"string\",\n \"renotify\": false,\n \"requireInteraction\":
 true,\n \"silent\": false,\n \"tag\": \"tag\",\n \"timestamp\": 1707259524964,\n
 \"title\": \"hello\",\n \"vibrate\": [\n 100,\n 200,\n 300\n ]\n },\n \"data\": {\n
 \"data1\": \"priority message\",\n \"data2\": \"priority message\",\n \"data12\":
 \"priority message\",\n \"data3\": \"priority message\"\n }\n },\n \"data\": {\n
 \"data7\": \"priority message\",\n \"data5\": \"priority message\",\n \"data8\":
 \"priority message\",\n \"data9\": \"priority message\"\n }\n }\n \n}\n}",
      "TimeToLive" : 309744
    }
  },
  "Addresses": {
    token: {
      "ChannelType":"GCM"
    }
   }
}'
\ --region us-east-1
```

si vous utilisez un `ImageUrl` champ pour GCM, pinpoint envoie le champ sous forme de notification de données, la clé étant`pinpoint.notification.imageUrl`, ce qui peut empêcher le rendu de l'image prête à l'emploi. Veuillez utiliser RawContent ou ajouter la gestion des clés de données, par exemple en intégrant votre application à AWS Amplify.

Safari (AWS CLI)

Vous pouvez utiliser AWS End User Messaging Push pour envoyer des messages aux ordinateurs macOS qui utilisent le navigateur Web Safari d'Apple. Pour envoyer un message au navigateur Safari, vous devez spécifier le contenu brut du message et inclure un attribut spécifique dans la charge utile du message. Vous pouvez le faire en créant un modèle de notification push avec une charge utile de message brute, ou en spécifiant le contenu brut du message directement dans un message de campagne, dans le guide de l'utilisateur Amazon Pinpoint.

> **Note**
>
> Cet attribut spécial est requis pour les envois vers les ordinateurs portables et de bureau macOS qui utilisent le navigateur Web Safari. Il n'est pas nécessaire pour l'envoi vers des appareils iOS tels que les iPhones et les iPads.

Pour envoyer un message aux navigateurs Web Safari, vous devez spécifier la charge utile brute du message. La charge utile du message brut doit inclure un tableau `url-args` dans l'objet `aps`. Le tableau `url-args` est nécessaire pour envoyer des notifications push au navigateur Web Safari. Toutefois, il est acceptable que le tableau contienne un seul élément vide.

L'exemple suivant utilise l'option d'envoi de messages pour envoyer une notification au navigateur Web Safari avec le. AWS CLI*token*Remplacez-le par le jeton unique de l'appareil et *611e3e3cdd47474c9c1399a50example* par l'identifiant de votre application.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
  "Addresses": {
    "token":
    {
      "ChannelType":"APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
        "RawContent":
          "{\"aps\": {\"alert\": { \"title\": \"Title of my message\", \"body\":
 \"This is a push notification for the Safari web browser.\"},\"content-available\":
 1,\"url-args\": [\"\"]}}"
     }
  }
}'
\ --region us-east-1
```

Pour plus d'informations sur les notifications push Safari, consultez Configuration des notifications push Safari sur le site Web des développeurs Apple.

APNS (AWS CLI)

L'exemple suivant utilise send-messages pour envoyer une notification APNS Push avec le. AWS CLI*token*Remplacez-le par le jeton unique de l'appareil, *611e3e3cdd47474c9c1399a50example* par l'identifiant de votre application et *GAME_INVITATION* par un identifiant unique.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
```

```
--message-request
'{
    "Addresses": {
     "token":
    {
      "ChannelType":"APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
      "RawContent": "{\"aps\" : {\"alert\" : {\"title\" : \"Game Request\",
\"subtitle\" : \"Five Card Draw\",\"body\" : \"Bob wants to play poker\"},\"category
\" : \"GAME_INVITATION\"},\"gameID\" : \"12345678\"}"
      }
    }
}'
\ --region us-east-1
```

JavaScript (Node.js)

Utilisez cet exemple pour envoyer des notifications push à l'aide du AWS SDK pour JavaScript dans Node.js. Cet exemple suppose que vous avez déjà installé et configuré le SDK pour JavaScript dans Node.js.

Cet exemple suppose que vous utilisez un fichier d'informations d'identification partagé pour spécifier la clé d'accès et la clé d'accès secrète d'un utilisateur existant. Pour plus d'informations, consultez la section Configuration des informations d'identification dans le AWS SDK pour JavaScript le guide du développeur Node.js.

```
'use strict';

const AWS = require('aws-sdk');

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the API is available
const region = 'us-east-1';

// The title that appears at the top of the push notification.
var title = 'Test message sent from End User Messaging Push.';

// The content of the push notification.
var message = 'This is a sample message sent from End User Messaging Push by using
 the '
```

```
            + 'AWS SDK for JavaScript in Node.js';

// The application ID that you want to use when you send this
// message. Make sure that the push channel is enabled for the project that
// you choose.
var applicationId = 'ce796be37f32f178af652b26eexample';

// An object that contains the unique token of the device that you want to send
// the message to, and the push service that you want to use to send the message.
var recipient = {
  'token': 'a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0',
  'service': 'GCM'
  };

// The action that should occur when the recipient taps the message. Possible
// values are OPEN_APP (opens the app or brings it to the foreground),
// DEEP_LINK (opens the app to a specific page or interface), or URL (opens a
// specific URL in the device's web browser.)
var action = 'URL';

// This value is only required if you use the URL action. This variable contains
// the URL that opens in the recipient's web browser.
var url = 'https://www.example.com';

// The priority of the push notification. If the value is 'normal', then the
// delivery of the message is optimized for battery usage on the recipient's
// device, and could be delayed. If the value is 'high', then the notification is
// sent immediately, and might wake a sleeping device.
var priority = 'normal';

// The amount of time, in seconds, that the push notification service provider
// (such as FCM or APNS) should attempt to deliver the message before dropping
// it. Not all providers allow you specify a TTL value.
var ttl = 30;

// Boolean that specifies whether the notification is sent as a silent
// notification (a notification that doesn't display on the recipient's device).
var silent = false;

function CreateMessageRequest() {
  var token = recipient['token'];
  var service = recipient['service'];
  if (service == 'GCM') {
    var messageRequest = {
```

```
      'Addresses': {
        [token]: {
          'ChannelType' : 'GCM'
        }
      },
      'MessageConfiguration': {
        'GCMMessage': {
          'Action': action,
          'Body': message,
          'Priority': priority,
          'SilentPush': silent,
          'Title': title,
          'TimeToLive': ttl,
          'Url': url
        }
      }
    };
  } else if (service == 'APNS') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'APNS'
        }
      },
      'MessageConfiguration': {
        'APNSMessage': {
          'Action': action,
          'Body': message,
          'Priority': priority,
          'SilentPush': silent,
          'Title': title,
          'TimeToLive': ttl,
          'Url': url
        }
      }
    };
  } else if (service == 'BAIDU') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'BAIDU'
        }
      },
      'MessageConfiguration': {
```

```
        'BaiduMessage': {
          'Action': action,
          'Body': message,
          'SilentPush': silent,
          'Title': title,
          'TimeToLive': ttl,
          'Url': url
        }
      }
    };
  } else if (service == 'ADM') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'ADM'
        }
      },
      'MessageConfiguration': {
        'ADMMessage': {
          'Action': action,
          'Body': message,
          'SilentPush': silent,
          'Title': title,
          'Url': url
        }
      }
    };
  }

  return messageRequest
}

function ShowOutput(data){
  if (data["MessageResponse"]["Result"][recipient["token"]]["DeliveryStatus"]
      == "SUCCESSFUL") {
    var status = "Message sent! Response information: ";
  } else {
    var status = "The message wasn't sent. Response information: ";
  }
  console.log(status);
  console.dir(data, { depth: null });
}

function SendMessage() {
```

```
  var token = recipient['token'];
  var service = recipient['service'];
  var messageRequest = CreateMessageRequest();

  // Specify that you're using a shared credentials file, and specify the
  // IAM profile to use.
  var credentials = new AWS.SharedIniFileCredentials({ profile: 'default' });
  AWS.config.credentials = credentials;

  // Specify the AWS Region to use.
  AWS.config.update({ region: region });

  //Create a new Pinpoint object.
  var pinpoint = new AWS.Pinpoint();
  var params = {
    "ApplicationId": applicationId,
    "MessageRequest": messageRequest
  };

  // Try to send the message.
  pinpoint.sendMessages(params, function(err, data) {
    if (err) console.log(err);
    else     ShowOutput(data);
  });
}

SendMessage()
```

Python

Utilisez cet exemple pour envoyer des notifications push à l'aide du kit AWS SDK pour Python (Boto3). Cette procédure suppose que vous avez déjà installé et configuré le kit SDK pour Python (Boto3).

Cet exemple suppose que vous utilisez un fichier d'informations d'identification partagé pour spécifier la clé d'accès et la clé d'accès secrète d'un utilisateur existant. Pour plus d'informations, consultez Informations d'identité dans la Référence des API du kit SDK AWS pour Python (Boto).

```
import json
import boto3
from botocore.exceptions import ClientError

# The AWS Region that you want to use to send the message. For a list of
```

```
# AWS Regions where the API is available
region = "us-east-1"

# The title that appears at the top of the push notification.
title = "Test message sent from End User Messaging Push."

# The content of the push notification.
message = ("This is a sample message sent from End User Messaging Push by using the
 "
           "AWS SDK pour Python (Boto3).")

# The application ID to use when you send this message.
# Make sure that the push channel is enabled for the project or application
# that you choose.
application_id = "ce796be37f32f178af652b26eexample"

# A dictionary that contains the unique token of the device that you want to send
 the
# message to, and the push service that you want to use to send the message.
recipient = {
    "token": "a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0",
    "service": "GCM"
    }

# The action that should occur when the recipient taps the message. Possible
# values are OPEN_APP (opens the app or brings it to the foreground),
# DEEP_LINK (opens the app to a specific page or interface), or URL (opens a
# specific URL in the device's web browser.)
action = "URL"

# This value is only required if you use the URL action. This variable contains
# the URL that opens in the recipient's web browser.
url = "https://www.example.com"

# The priority of the push notification. If the value is 'normal', then the
# delivery of the message is optimized for battery usage on the recipient's
# device, and could be delayed. If the value is 'high', then the notification is
# sent immediately, and might wake a sleeping device.
priority = "normal"

# The amount of time, in seconds, that the push notification service provider
# (such as FCM or APNS) should attempt to deliver the message before dropping
# it. Not all providers allow you specify a TTL value.
ttl = 30
```

```
# Boolean that specifies whether the notification is sent as a silent
# notification (a notification that doesn't display on the recipient's device).
silent = False

# Set the MessageType based on the values in the recipient variable.
def create_message_request():

    token = recipient["token"]
    service = recipient["service"]

    if service == "GCM":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'GCM'
                }
            },
            'MessageConfiguration': {
                'GCMMessage': {
                    'Action': action,
                    'Body': message,
                    'Priority' : priority,
                    'SilentPush': silent,
                    'Title': title,
                    'TimeToLive': ttl,
                    'Url': url
                }
            }
        }
    elif service == "APNS":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'APNS'
                }
            },
            'MessageConfiguration': {
                'APNSMessage': {
                    'Action': action,
                    'Body': message,
                    'Priority' : priority,
                    'SilentPush': silent,
                    'Title': title,
```

```
                'TimeToLive': ttl,
                'Url': url
            }
        }
    }
elif service == "BAIDU":
    message_request = {
        'Addresses': {
            token: {
                'ChannelType': 'BAIDU'
            }
        },
        'MessageConfiguration': {
            'BaiduMessage': {
                'Action': action,
                'Body': message,
                'SilentPush': silent,
                'Title': title,
                'TimeToLive': ttl,
            'Url': url
            }
        }
    }
elif service == "ADM":
    message_request = {
        'Addresses': {
            token: {
                'ChannelType': 'ADM'
            }
        },
        'MessageConfiguration': {
            'ADMMessage': {
                'Action': action,
                'Body': message,
                'SilentPush': silent,
                'Title': title,
                'Url': url
            }
        }
    }
else:
    message_request = None

return message_request
```

```
# Show a success or failure message, and provide the response from the API.
def show_output(response):
    if response['MessageResponse']['Result'][recipient["token"]]['DeliveryStatus']
 == "SUCCESSFUL":
        status = "Message sent! Response information:\n"
    else:
        status = "The message wasn't sent. Response information:\n"
    print(status, json.dumps(response,indent=4))

# Send the message through the appropriate channel.
def send_message():

    token = recipient["token"]
    service = recipient["service"]
    message_request = create_message_request()

    client = boto3.client('pinpoint',region_name=region)

    try:
        response = client.send_messages(
            ApplicationId=application_id,
            MessageRequest=message_request
        )
    except ClientError as e:
        print(e.response['Error']['Message'])
    else:
        show_output(response)

send_message()
```

Ressources supplémentaires

- Pour plus d'informations sur les modèles de canaux Push, consultez la section Création de modèles de notifications push dans le guide de l'utilisateur Amazon Pinpoint.

Recevoir des notifications push dans votre application

Les rubriques suivantes décrivent comment modifier votre application Swift, Android, React Native ou Flutter afin qu'elle reçoive des notifications push.

Rubriques

Configuration des notifications Swift Push

Les notifications push pour les applications iOS sont envoyées à l'aide du service Apple Push Notification (APNs). Avant de pouvoir envoyer des notifications push aux appareils iOS, vous devez créer un ID d'application sur le portail Apple Developer, ainsi que les certificats requis. Vous trouverez plus d'informations sur la réalisation de ces étapes dans Configuration des services de notification push dans la documentation AWS Amplify.

Travailler avec des APNs jetons

Une bonne pratique consiste à développer votre application afin que les jetons d'appareil de vos clients soient actualisés lorsque l'application est réinstallée.

Si un destinataire met à niveau son appareil vers une nouvelle version majeure d'iOS (par exemple, d'iOS 12 à iOS 13) et réinstalle ultérieurement votre application, celle-ci génère un nouveau jeton. Si votre application n'actualise pas le jeton, l'ancien jeton est utilisé pour envoyer la notification. Par conséquent, le service Apple Push Notification (APNs) rejette la notification, car le jeton n'est plus valide. Lorsque vous tentez d'envoyer la notification, vous recevez une notification d'échec du message de APNs.

Configuration des notifications push Android

Les notifications push pour les applications Android sont envoyées à l'aide de Firebase Cloud Messaging (FCM), qui remplace Google Cloud Messaging (GCM). Avant de pouvoir envoyer des notifications push aux appareils Android, vous devez obtenir des informations d'identification FCM. Vous pouvez utiliser ces informations pour créer un projet Android et lancer un exemple d'application qui peut recevoir des notifications push. Vous trouverez plus d'informations sur la réalisation de ces étapes dans la section Notifications push de la documentation AWS Amplify.

Configuration des notifications push Flutter

Les notifications push pour les applications Flutter sont envoyées à l'aide de Firebase Cloud Messaging (FCM) pour Android et pour APNs iOS. Pour plus d'informations sur la réalisation de ces étapes, consultez la section Notifications push de la documentation AWS d'Amplify Flutter.

Configuration des notifications push React Native

Les notifications push pour les applications React Native sont envoyées à l'aide de Firebase Cloud Messaging (FCM) pour Android et pour APNs iOS. Vous trouverez plus d'informations sur la réalisation de ces étapes dans la section Notifications push de la documentation AWS Amplify JavaScript.

Création d'une application dans AWS End User Messaging Push

Pour commencer à envoyer des notifications push dans AWS End User Messaging Push, vous devez créer une application. Vous devez ensuite activer les canaux de notification push à utiliser en fournissant les informations d'identification appropriées.

Vous pouvez créer de nouvelles applications et configurer des canaux de notification push à l'aide de la console AWS Find User Messaging Push. Pour de plus amples informations, veuillez consulter Création d'une application et activation des canaux push.

Vous pouvez également créer et configurer une application à l'aide de l'API, d'un AWS SDK ou du AWS Command Line Interface(AWS CLI). Pour créer une application, utilisez la `Apps` ressource. Pour configurer des canaux de notification push, utilisez les ressources suivantes :

- APNs canal pour envoyer des messages aux utilisateurs d'appareils iOS à l'aide du service Apple Push Notification.

- Canal ADM pour envoyer des messages aux utilisateurs d'appareils Amazon Kindle Fire.
- Canal Baidu pour envoyer des messages aux utilisateurs de Baidu.
- Canal GCM pour envoyer des messages aux appareils Android à l'aide de Firebase Cloud Messaging (FCM), qui remplace Google Cloud Messaging (GCM).

Gestion des notifications push

Après avoir obtenu les informations d'identification requises pour envoyer des notifications push, vous pouvez mettre à jour votre application afin qu'elle puisse recevoir des notifications push. Pour plus d'informations, consultez la section Notifications push : mise en route dans la documentation. AWS Amplify

Suppression d'une application

Cette procédure supprime l'application de votre compte et de toutes les ressources qu'elle contient.

Contextuel

Application

Une application est un conteneur de stockage pour tous vos paramètres push de messagerie utilisateur AWS final. L'application enregistre également les paramètres de vos chaînes, campagnes et parcours Amazon Pinpoint.

Procédure

1. Ouvrez la console AWS Final User Messaging Push à l'adresse https://console.aws.amazon.com/push-notifications/.
2. Choisissez une application, puis cliquez sur Supprimer.
3. Dans la fenêtre Supprimer l'application, entrez **`delete`** puis choisissez Supprimer.

> **Important**
>
> Tous les canaux, campagnes, parcours ou segments Amazon Pinpoint sont également supprimés.

Bonnes pratiques

Même lorsque vous avez l'intérêt de vos clients à l'esprit, vous pouvez être confronté à des situations qui ont un impact négatif sur la délivrabilité de vos messages. Les sections suivantes contiennent des recommandations qui vous aideront à vous assurer que vos communications push atteignent le public visé.

Envoi d’un volume élevé de notifications push

Avant d'envoyer un volume élevé de notifications push, assurez-vous que votre compte est configuré pour répondre à vos exigences de débit. Par défaut, tous les comptes sont configurés pour envoyer 25 000 messages par seconde. Si vous devez être en mesure d’envoyer plus de 25 000 messages en une seconde, demandez une augmentation de quota. Pour de plus amples informations, veuillez consulter Quotas pour les messages push destinés aux utilisateurs AWS finaux.

Assurez-vous que votre compte est correctement configuré avec les informations d'identification de chacun des fournisseurs de notifications push que vous prévoyez d'utiliser, tels que FCM ou APNs.

Enfin, trouvez un moyen de gérer les exceptions. Chaque service de notification push fournit des messages d’exception différents. Pour les envois transactionnels, vous recevez un code de statut principal de 200 pour l’appel d’API, avec un code de statut par point de terminaison de 400 défaillances permanentes s’il est déterminé que le jeton de plateforme (par exemple, FCM) ou le certificat (par exemple, APN) correspondant n’est pas valide lors de l’envoi des messages.

Sécurité dans les messages push destinés aux utilisateurs AWS finaux

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS la messagerie push destinée aux utilisateurs finaux, voir AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de la messagerie push pour l'utilisateur AWS final. Les rubriques suivantes expliquent comment configurer la messagerie push destinée aux utilisateurs AWS finaux pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources de messagerie push destinées aux utilisateurs AWS finaux.

Rubriques

- Analyse de la configuration et des vulnérabilités
- Bonnes pratiques de sécurité

Protection des données dans AWS Final User Messaging Push

Le modèle de responsabilité AWS partagée Le modèle s'applique à la protection des données dans AWS Final User Messaging Push. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée d'AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec AWS End User Messaging Push ou autre Services AWS à l'aide de la console AWS CLI, de l'API ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement des données

AWS Messagerie utilisateur final Les données push sont cryptées en transit et au repos. Lorsque vous soumettez des données à AWS End User Messaging Push, celui-ci chiffre les données au fur et à mesure qu'il les reçoit et les stocke. Lorsque vous récupérez des données depuis AWS End User Messaging Push, celui-ci vous les transmet en utilisant les protocoles de sécurité actuels.

Chiffrement au repos

AWS End User Messaging Push chiffre toutes les données qu'il stocke pour vous. Cela inclut les données de configuration, les données des utilisateurs et des terminaux, les données analytiques et toutes les données que vous ajoutez ou importez dans AWS End User Messaging Push. Pour chiffrer vos données, AWS End User Messaging Push utilise des clés internes AWS Key Management Service (AWS KMS) que le service possède et gère en votre nom. Ces clés font l'objet d'une rotation régulière. Pour plus d'informations à ce sujet AWS KMS, consultez le guide du AWS Key Management Service développeur.

Chiffrement en transit

AWS La messagerie push destinée aux utilisateurs finaux utilise le protocole HTTPS et le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure pour communiquer avec vos clients et vos applications. Pour communiquer avec d'autres AWS services, le service AWS Final User Messaging Push utilise le protocole HTTPS et le protocole TLS 1.2. En outre, lorsque vous créez et gérez des ressources push de messagerie utilisateur AWS final à l'aide de la console, d'un AWS SDK ou du AWS Command Line Interface, toutes les communications sont sécurisées à l'aide des protocoles HTTPS et TLS 1.2.

Gestion des clés

Pour chiffrer les données Push de votre messagerie utilisateur AWS AWS final, la messagerie push de l'utilisateur final utilise des AWS KMS clés internes que le service possède et gère en votre nom. Ces clés font l'objet d'une rotation régulière. Vous ne pouvez pas fournir et utiliser vos propres clés AWS KMS ou d'autres clés pour chiffrer les données que vous stockez dans AWS End User Messaging Push.

Confidentialité du trafic inter-réseaux

La confidentialité du trafic interréseau fait référence à la sécurisation des connexions et du trafic entre l'utilisateur AWS final de Messaging Push et vos clients et applications sur site, et entre l'utilisateur AWS final Messaging Push et les autres AWS ressources de la même AWS région. Les fonctionnalités et pratiques suivantes peuvent vous aider à garantir la confidentialité du trafic interréseau pour les messages push destinés aux utilisateurs AWS finaux.

Trafic entre AWS la messagerie Push de l'utilisateur final et les clients et applications sur site

Pour établir une connexion privée entre l'utilisateur AWS final Messaging Push et les clients et applications de votre réseau local, vous pouvez utiliser Direct Connect. Cela vous permet de relier votre réseau à un emplacement AWS Direct Connect à l'aide d'un câble Ethernet standard à fibre optique. Une extrémité du câble est connectée à votre routeur. L'autre extrémité est connectée à un Direct Connect routeur. Pour plus d'informations, consultez Présentation de Direct Connect dans le Guide de l'utilisateur Direct Connect .

Pour sécuriser l'accès aux messages push through publiés pour les utilisateurs AWS finaux APIs, nous vous recommandons de respecter les exigences relatives à la messagerie push pour les utilisateurs AWS finaux pour les appels d'API. AWS La messagerie push destinée aux utilisateurs finaux exige que les clients utilisent le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Les clients doivent également prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un identifiant de clé d'accès et d'une clé d'accès secrète associés à un principal Gestion des identités et des accès AWS (IAM) de votre AWS compte. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Trafic entre l'utilisateur AWS final, Messaging Push et les autres AWS ressources

Pour sécuriser les communications entre AWS la messagerie push de l'utilisateur final et les autres AWS ressources de la même AWS région, la messagerie push de l'utilisateur AWS final utilise HTTPS et TLS 1.2 par défaut.

Gestion des identités et des accès pour AWS Final User Messaging Push

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources push de messagerie utilisateur AWS final. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir Résolution des problèmes liés AWS à l'identité et à l'accès push de l'utilisateur final)
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir Comment fonctionne AWS Final User Messaging Push avec IAM)

- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir Exemples de politiques basées sur l'identité pour les messages push destinés aux utilisateurs AWS finaux)

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez Signature AWS Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle utilisateur à un rôle IAM (console) ou en appelant une opération AWS CLI ou AWS API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez spécifier un principal dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez Politiques de contrôle de service dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.

- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment fonctionne AWS Final User Messaging Push avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS End User Messaging Push, découvrez quelles fonctionnalités IAM peuvent être utilisées avec AWS End User Messaging Push.

Fonctionnalités IAM que vous pouvez utiliser avec AWS End User Messaging Push

Fonctionnalité IAM	AWS Assistance Push pour les utilisateurs finaux
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Oui
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique	Oui
ACLs	Non
ABAC (identifications dans les politiques)	Partielle
Informations d'identification temporaires	Oui
Autorisations de principal	Oui

Fonctionnalité IAM	AWS Assistance Push pour les utilisateurs finaux
Rôles de service	Oui
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont les services AWS Final User Messaging Push et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour la messagerie push de l'utilisateur AWS final

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour les messages push destinés aux utilisateurs AWS finaux

Pour consulter des exemples de politiques de messagerie push basées sur l'identité des utilisateurs AWS finaux, consultez. Exemples de politiques basées sur l'identité pour les messages push destinés aux utilisateurs AWS finaux

Politiques basées sur les ressources dans le cadre de AWS Final User Messaging Push

Prend en charge les politiques basées sur les ressources : oui

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Actions stratégiques pour la messagerie push destinée aux utilisateurs AWS finaux

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions push de messagerie utilisateur AWS final, voir Actions définies par le push de messagerie utilisateur AWS final dans la référence d'autorisation de service.

Les actions de politique dans AWS End User Messaging Push utilisent le préfixe suivant avant l'action :

```
mobiletargeting
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "mobiletargeting:action1",
```

```
      "mobiletargeting:action2"
         ]
```

Pour consulter des exemples de politiques de messagerie push basées sur l'identité des utilisateurs AWS finaux, consultez. Exemples de politiques basées sur l'identité pour les messages push destinés aux utilisateurs AWS finaux

Ressources relatives aux politiques relatives à AWS la messagerie push destinée aux utilisateurs finaux

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de ressources Push de messagerie utilisateur AWS final et leurs caractéristiques ARNs, voir Ressources définies par le service Push de messagerie utilisateur AWS final dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, voir Actions définies par le push de messagerie utilisateur AWS final.

Pour consulter des exemples de politiques de messagerie push basées sur l'identité des utilisateurs AWS finaux, consultez. Exemples de politiques basées sur l'identité pour les messages push destinés aux utilisateurs AWS finaux

Clés de conditions de politique pour les messages push destinés aux utilisateurs AWS finaux

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions s'exécutent en fonction de critères définis. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition Push de messagerie utilisateur AWS final, voir Clés de condition pour AWS la messagerie push de l'utilisateur final dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, voir Actions définies par le push de messagerie utilisateur AWS final.

Pour consulter des exemples de politiques de messagerie push basées sur l'identité des utilisateurs AWS finaux, consultez. Exemples de politiques basées sur l'identité pour les messages push destinés aux utilisateurs AWS finaux

ACLs dans AWS Final User Messaging Push

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec messagerie push pour les utilisateurs AWS finaux

Prend en charge ABAC (identifications dans les politiques) : partiellement

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs nommés balise. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez Définition d'autorisations avec l'autorisation ABAC dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec AWS Final User Messaging Push

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM et Services AWS compatibles avec IAM dans le Guide de l'utilisateur IAM.

Autorisations principales interservices pour les messages push destinés aux utilisateurs AWS finaux

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section Sessions de transmission d'accès.

Rôles de service pour l'utilisateur AWS final Messaging Push

Prend en charge les rôles de service : oui

Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber la fonctionnalité Push de messagerie utilisateur AWS final. Modifiez les rôles de service uniquement lorsque AWS End User Messaging Push fournit des instructions à cet effet.

Rôles liés à un service pour la messagerie push de l'utilisateur AWS final

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour les messages push destinés aux utilisateurs AWS finaux

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources AWS Final User Messaging Push. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques IAM (console) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par AWS End User Messaging Push, y compris le format du ARNs pour chacun des types de ressources, voir Actions, ressources et clés de condition pour le push de messagerie utilisateur AWS final dans la référence d'autorisation de service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console push de messagerie utilisateur AWS final

- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources push de messagerie utilisateur AWS final dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console push de messagerie utilisateur AWS final

Pour accéder à la console AWS Final User Messaging Push, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher des informations détaillées sur les ressources Push de messagerie utilisateur AWS final de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la console push de messagerie utilisateur AWS final, associez également la politique `AWSEndUserMessaging` AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AWSEndUserMessaging",
   "Effect": "Allow",
   "Action": [
    "mobiletargeting:CreateApp",
                "mobiletargeting:GetApp",
                "mobiletargeting:GetApps",
```

```
                                "mobiletargeting:DeleteApp",
                                "mobiletargeting:GetChannels",
                                "mobiletargeting:GetApnsChannel",
                                "mobiletargeting:GetApnsVoipChannel",
                                "mobiletargeting:GetApnsVoipSandboxChannel",
                                "mobiletargeting:GetApnsSandboxChannel",
                    "mobiletargeting:GetAdmChannel",
                    "mobiletargeting:GetBaiduChannel",
                    "mobiletargeting:GetGcmChannel",
                    "mobiletargeting:UpdateApnsChannel",
                    "mobiletargeting:UpdateApnsVoipChannel",
                    "mobiletargeting:UpdateApnsVoipSandboxChannel",
                    "mobiletargeting:UpdateBaiduChannel",
                    "mobiletargeting:UpdateGcmChannel",
                    "mobiletargeting:UpdateAdmChannel"
                ],
    "Resource": [
     "*"
    ]
   }
  ]
}
```

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
```

```
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Résolution des problèmes liés AWS à l'identité et à l'accès push de l'utilisateur final

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous utilisez AWS End User Messaging Push et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS End User Messaging Push
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures Compte AWS à moi à accéder à mes ressources push de messagerie utilisateur AWS final

Je ne suis pas autorisé à effectuer une action dans AWS End User Messaging Push

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `mobiletargeting:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mobiletargeting:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `mobiletargeting:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à AWS End User Messaging Push.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans AWS End User Messaging Push. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures Compte AWS à moi à accéder à mes ressources push de messagerie utilisateur AWS final

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si AWS End User Messaging Push prend en charge ces fonctionnalités, consultezComment fonctionne AWS Final User Messaging Push avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Validation de conformité pour les messages push destinés aux utilisateurs AWS finaux

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez AWS la documentation de sécurité.

Résilience dans les messages push destinés aux utilisateurs AWS finaux

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Outre l'infrastructure AWS mondiale, AWS End User Messaging Push propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

Sécurité de l'infrastructure dans le cadre de la messagerie instantanée destinée aux utilisateurs AWS finaux

En tant que service géré, AWS End User Messaging Push est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à la messagerie push de l'utilisateur AWS final via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service

(AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Analyse de la configuration et des vulnérabilités

En tant que service géré, AWS End User Messaging Push est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité. Cela signifie qu'il AWS gère et exécute des tâches et des procédures de sécurité de base pour renforcer, corriger, mettre à jour et entretenir de toute autre manière l'infrastructure sous-jacente de votre compte et de vos ressources. Ces procédures ont été vérifiées et certifiées par les tiers appropriés.

Bonnes pratiques de sécurité

Utilisez les comptes AWS Identity and Access Management (IAM) pour contrôler l'accès aux opérations d'API, en particulier aux opérations qui créent, modifient ou suppriment des ressources. Pour l'API , ces ressources incluent les projets, les campagnes et les parcours.

- Créez un utilisateur pour chaque personne qui gère les ressources , y compris vous-même. N'utilisez pas les informations d'identification AWS root pour gérer les ressources.
- Accordez à chaque utilisateur un ensemble minimum d'autorisations requises pour exécuter ses tâches.
- Utilisez des groupes IAM pour gérer efficacement des autorisations pour plusieurs utilisateurs.
- Effectuer une rotation régulière des informations d'identification IAM.

Pour plus d'informations sur la sécurité, consultez Sécurité dans les messages push destinés aux utilisateurs AWS finaux. Pour plus d'informations sur IAM, consultez AWS Identity and Access Management. Pour plus d'informations sur les bonnes pratiques IAM, consultez Bonnes pratiques IAM.

Surveillance des messages push destinés aux utilisateurs AWS finaux

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances de AWS Final User Messaging Push et de vos autres solutions AWS. AWS fournit les outils de surveillance suivants pour surveiller les messages envoyés aux utilisateurs AWS finaux, signaler les problèmes et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs de vos EC2 instances Amazon et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir d' EC2 instances Amazon et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.
- Amazon EventBridge peut être utilisé pour automatiser vos AWS services et répondre automatiquement aux événements du système, tels que les problèmes de disponibilité des applications ou les modifications des ressources. Les événements AWS liés aux services sont diffusés EventBridge 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. Pour plus d'informations, consultez le guide de EventBridge l'utilisateur Amazon.
- AWS CloudTrailcapture les appels d'API et les événements associés effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source à partir de laquelle les appels ont été effectués et la date des appels. Pour plus d'informations, consultez le AWS CloudTrail Guide de l'utilisateur .

Surveillance des messages push destinés aux utilisateurs AWS finaux avec Amazon CloudWatch

Vous pouvez surveiller les messages push destinés aux utilisateurs AWS finaux à l'aide de ce système CloudWatch, qui collecte les données brutes et les traite en indicateurs lisibles en temps quasi réel. Ces statistiques sont enregistrées pour une durée de 15 mois ; par conséquent, vous pouvez accéder aux informations historiques et acquérir un meilleur point de vue de la façon dont votre service ou application web s'exécute. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Pour obtenir la liste des mesures et des dimensions, consultez la section Surveillance d'Amazon Pinpoint with CloudWatch dans le guide de l'utilisateur d'Amazon Pinpoint.

Enregistrement des appels de l'API Push de messagerie utilisateur AWS final à l'aide de AWS CloudTrail

AWS End User Messaging Push est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS End User Messaging Push. CloudTrail capture tous les appels d'API pour le push de messagerie utilisateur AWS final sous forme d'événements. Les appels capturés incluent des appels provenant de la console AWS Final User Messaging Push et des appels de code vers les opérations de l'API AWS Final User Messaging Push. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour le push de messagerie utilisateur AWS final. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à AWS End User Messaging Push, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

AWS Messagerie à l'utilisateur final Transférez les informations CloudTrail

CloudTrail est activé sur votre compte Compte AWS lorsque vous créez le compte. Lorsqu'une activité se produit dans AWS End User Messaging Push, cette activité est enregistrée dans un

CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements survenus dans votre environnement Compte AWS, y compris les événements relatifs à la messagerie push destinée aux utilisateurs AWS finaux, créez une trace. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les actions push de messagerie utilisateur AWS final sont enregistrées CloudTrail et documentées dans le document de référence de l'API AWS Final User Messaging Push. Par exemple, les appels au `GetAdmChannel` `UpdateApnsChannel` et les `GetApnsVoipChannel` actions génèrent des entrées dans les fichiers CloudTrail journaux.

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

- Si la demande a été faite avec les informations d'identification de l'utilisateur root ou Gestion des identités et des accès AWS (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

AWS Comprendre les entrées du fichier journal push de la messagerie utilisateur final

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

Accédez à AWS la messagerie push de l'utilisateur final à l'aide d'un point de terminaison d'interface (AWS PrivateLink)

Vous pouvez l'utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et l'utilisateur AWS final Messaging Push. Vous pouvez accéder à AWS la messagerie push de l'utilisateur final comme si elle se trouvait dans votre VPC, sans utiliser de passerelle Internet, de périphérique NAT, de connexion VPN ou Direct Connect de connexion. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour accéder à la messagerie push destinée aux utilisateurs AWS finaux.

Vous établissez cette connexion privée en créant un point de terminaison d'interface optimisé par AWS PrivateLink. Nous créons une interface réseau de point de terminaison dans chaque sous-réseau que vous activez pour le point de terminaison d'interface. Il s'agit d'interfaces réseau gérées par les demandeurs qui servent de point d'entrée pour le trafic destiné à la messagerie push de l'utilisateur AWS final.

Pour plus d'informations, consultez la section Accès Services AWS par AWS PrivateLink le biais du AWS PrivateLink guide.

Considérations relatives à AWS la messagerie push destinée aux utilisateurs finaux

Avant de configurer un point de terminaison d'interface pour AWS Final User Messaging Push, consultez les considérations du AWS PrivateLink guide.

AWS End User Messaging Push permet d'appeler toutes ses actions d'API via le point de terminaison de l'interface.

Les politiques de point de terminaison VPC ne sont pas prises en charge pour les messages push destinés aux utilisateurs AWS finaux. Par défaut, l'accès complet à AWS End User Messaging Push est autorisé via le point de terminaison de l'interface. Vous pouvez également associer un groupe de sécurité aux interfaces réseau du point de terminaison afin de contrôler le trafic vers le message push de l'utilisateur AWS final via le point de terminaison de l'interface.

Création d'un point de terminaison d'interface pour les messages push destinés aux utilisateurs AWS finaux

Vous pouvez créer un point de terminaison d'interface pour AWS End User Messaging Push à l'aide de la console Amazon VPC ou du AWS Command Line Interface ()AWS CLI. Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide AWS PrivateLink .

Créez un point de terminaison d'interface pour AWS End User Messaging Push en utilisant le nom de service suivant :

```
com.amazonaws.region.pinpoint
```

Si vous activez le DNS privé pour le point de terminaison de l'interface, vous pouvez envoyer des demandes d'API à AWS End User Messaging Push en utilisant son nom DNS régional par défaut. Par exemple, `com.amazonaws.us-east-1.pinpoint`.

Création d'une politique de point de terminaison pour votre point de terminaison d'interface

Une politique de point de terminaison est une ressource IAM que vous pouvez attacher à votre point de terminaison d'interface. La politique de point de terminaison par défaut autorise un accès complet à AWS la messagerie push de l'utilisateur final via le point de terminaison de l'interface. Pour contrôler l'accès autorisé à la messagerie push de l'utilisateur AWS final depuis votre VPC, associez une politique de point de terminaison personnalisée au point de terminaison de l'interface.

Une politique de point de terminaison spécifie les informations suivantes :

- Les principaux qui peuvent effectuer des actions (Comptes AWS, utilisateurs IAM et rôles IAM).
- Les actions qui peuvent être effectuées.
- La ressource sur laquelle les actions peuvent être effectuées.

Pour plus d'informations, consultez Contrôle de l'accès aux services à l'aide de politiques de point de terminaison dans le Guide AWS PrivateLink .

Exemple : politique de point de terminaison VPC pour les actions AWS push de messagerie utilisateur final

Voici un exemple de politique de point de terminaison personnalisée. Lorsque vous associez cette politique au point de terminaison de votre interface, elle accorde l'accès aux actions push de messagerie utilisateur AWS final répertoriées pour tous les principaux sur toutes les ressources.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
             "mobiletargeting:CreateApp",
             "mobiletargeting:DeleteApp"
         ],
         "Resource":"*"
      }
   ]
}
```

Quotas pour les messages push destinés aux utilisateurs AWS finaux

Vous Compte AWS disposez de quotas par défaut, anciennement appelés limites, pour chaque AWS service. Sauf indication contraire, chaque quota est spécifique à la région. Vous pouvez demander des augmentations pour certains quotas, et d'autres quotas ne peuvent pas être augmentés.

Pour consulter les quotas pour les messages push destinés aux utilisateurs AWS finaux, ouvrez la console Service Quotas. Dans le volet de navigation, choisissez les services AWS, puis Amazon Pinpoint.

Votre compte AWS dispose des quotas suivants relatifs à la messagerie push destinée aux utilisateurs AWS finaux.

Ressource	Quota par défaut	Éligible à une augmentation
Nombre maximal de notifications push qui peuvent être envoyées par seconde dans une campagne	25 000 notifications par seconde	Oui, utilisez la console Service Quotas
Taille de la charge utile des messages Amazon Device Messaging (ADM)	6 Ko par message	Non
Taille de la charge utile des messages du service Apple Push Notification (APNs)	4 Ko par message	Non
APNs taille de la charge utile des messages du sandbox	4 Ko par message	Non
Taille de la charge utile des messages Baidu Cloud Push	4 Ko par message	Non

Ressource	Quota par défaut	Éligible à une augmentation
Taille de la charge utile des messages Firebase Cloud Messaging (FCM)	4 Ko par message	Non

Historique du document pour le guide de l'utilisateur AWS final relatif à la messagerie push

Le tableau suivant décrit les versions de documentation relatives à la messagerie push destinée aux utilisateurs AWS finaux.

Modification	Description	Date
Première version	Publication initiale du guide d'utilisation de la messagerie push destinée aux utilisateurs AWS finaux	24 juillet 2024

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.